

# Trip Report

## POPL/VMCAI 2010

Madrid, Spain

오학주 [pronto@ropas.snu.ac.kr](mailto:pronto@ropas.snu.ac.kr)



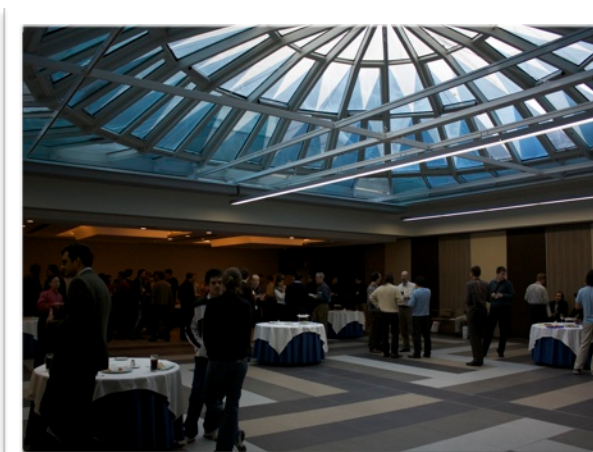
## 1. 개요

2010년 1월 15일부터 1월 25일의 일정으로 스페인 마드리드에서 열린 POPL/VMCAI 2010 학회에 다녀왔다. POPL은 명실공히 프로그래밍 언어 분야의 최고학회이다. 과연 이런 학회에는 어떤 사람들이 모일지, 어떤 논문들이 발표될지 출발전부터 많은 기대가 되었다. 또한 처음 발을 디더보는 스페인이라는 나라가 어떤 곳일지도 궁금했다. 이 글을 통해서 이번 여행을 통해 보고 배우고 느낀것들을 조금이나마 공유해 보고자 한다.

## 2. 학회

마드리드에 도착 후 이틀뒤부터 3일씩 VMCAI와 POPL이 진행되었다. 학회시작 전날 아침까지만 해도 호텔 레스토랑에 사람들이 별로 없었는데, 학회 첫날 아침에는 상당히 북적북적 해진것을 알 수 있었다. VMCAI는 POPL보다는 아담한 분위기로 시작되었다. 학회장은 생각보다 규모가 작아서 약 60-70명 정도 앉을 수 있는 곳이었었는데 나중에는 사람들이 많아져서 서서 듣는 사람들도 많았다. 또한 POPL이 프로그래밍 언어 학회라고 해도 다양한 분야의 논문들이 발표되는데 비해서 VMCAI에서는 보다 범위가 좁았다. 개인적으로는 POPL보다 VMCAI 논문들이 좀 더 친숙하게 느껴졌다. 이에 비해 POPL은 발표시간이 좀 더 짧기도 했고(VMCAI는 30분, POPL은 20분) 분야도 좀 더 다양해서 짧은 시간에 각 연구의 내용을 파악하기가 어려웠다.

### 2.1 VMCAI



VMCAI는 Reinhard Wilhelm 교수님의 wcet(worst case execution time)에 대한 invited talk으로 시작했다. VMCAI에서는 model checking, static analysis, program verification 분야의 논문발표가 주였던 것 같다. static analysis나 program verification 에 관한 발표는 문제와 아이디어의 전체적인 아이디어를 파악하는데 큰 어려움은 없었지만 model checking, logic, concurrency관련 논문들은 문제부터 이해하는데 어려움을 겪었다. 문제를 이해못하다 보니 해결책을 논하는 부분이 이해될리 없었다. 내 분야가 아니더라도 비슷한 커뮤니티의 사람들이 연구하는 기본적인 문제들에 대한 이해는 필요하다는 생각이 학회기간내내 들었다.

여유로운 발표 스케줄과 넉넉한 쉬는시간이 인상적이었다. 주로 3개의 논문발표(POPL: 1시간 또는 1시간 30분이 걸린다)에 30분 쉬는 시간이었다. 쉬는 시간에는 사람들은 커피, 과자와 함께 다른사람들과 자유로운 분위기에서 대화한다. 이전 세션에서 발표된 논문에 대한 이야기를 하기도 하고, 알고 지냈던 사람등과 이야기 한다. 나도 Antoine

Mine를 비롯해서 평소 논문을 읽으면서 궁금했던 것들에 대해 저자에게 직접 물어보는 기회를 가질 수도 있었다.

사람들을 직접 보고 한마디 말이라도 직접 건네볼 수 있는것이 학회의 매력임을 다시 느낄 수 있었던 기회였다. 사실 VMCAI를 하는 동안에는 누가 누구인지 잘 알 수 없었다. 사람들의 홈페이지에 얼굴사진들이 있기 때문에 웬만큼 유명한 사람들은 바로 알아볼 수 있을거라고 생각했지만 생각만큼 그렇지 못한 것 같다. 서양사람 얼굴에 익숙치 않아서 그런지 이름은 익히 들어본 사람도 실제 모습과 이미지가 잘 연결되지 않았다. 하나의 예로, POPL reception에서 20명씩 그룹으로 묶어서 프라도 미술관 가이드 투어를 했는데, 나중에 알고보니 내가 속해있던 조에 Benjarmin C.Pierce가 있었다. 그 때는 못알아 봤는데 나중에 POPL에서 알게 되었다. reception에서 Matthew Might (그 전날 VMCAI 발표를 통해서 얼굴과 이름을 매치시킬수 있었던)과 이야기할 기회가 있었는데, 2008년 PLDI에서 포스터 발표를 할 때 내 발표에 관심을 많이 보이며 이것저것 물어보던 사람이었음이 생각났다. 이번 SAS에 논문을 낼 것이라고 했고 주제에 대해서 간단히 얘기하니 아마도 자기가 리뷰를 하게 될것 같다고 했다.

## 2.2 POPL



POPL은 굉장히 큰 컨퍼런스 룸에서 진행되었다. 200명 정도가 앉을 수 있고 뒤에서는 스크린이 잘 보이지 않아서 중간중간 양 벽쪽으로 TV가 설치되어 있다. 그럼에도 불구하고 뒤에 앉으면 슬라이드를 보기 어려웠다. 질문은 중간에 설치된 마이크로 나가서 하는 구조였다.

POPL의 논문발표들은 예상외로 소박했다. 2008년 PLDI에 참가했을때 발표들이 하나 같이 화려하고 현란해서 POPL도 그럴것이라 생각했다. 하지만 POPL 발표자들은 발표 스킬이 화려하다기 보다는 의외로 소박하고 충실히 내용전달을 하는 분위기 였다. PLDI의 화려함도 멋졌지만 소박하고 충실한 POPL의 스타일도 좋았다. 재밌었던 점은 각 세션마다 그 분야의 대가들이 나서서 적극적으로 질문하고 공격한다는 것이었다. 예를 들어, 타입에 대한 발표면 어김없이 Phillip Wadler가 질문과 코멘트를 했다. VMCAI나 지난번 PLDI때와는 또 다른 분위기 였다.

역시나 발표 자체보다는 사람들끼리의 토론에 시간이 많이 할애 되었다. 각 발표는 20분 쯤으로 논문수는 VMCAI보다 훨씬 많았지만 기간은 똑같이 3일로 유지되었다. 세션들 사이에 있는 쉬는 시간에 호텔로비는 커피를 마시며 대화하는 사람들로 굉장히 시끌벅적해 졌으며, 꽤 많은 사람들이 쉬는 시간이 끝나도 얘기를 마치지 못하고 다음 세션에 조

금 늦게 들어오기도 했다. 점심시간은 호텔 뷔페식이었는데 이 때에도 역시 천천히 식사하면서 이야기를 하는 분위기이다.

다양한 분야의 논문들이 비슷한 비율로 발표되었던 것 같다. static analysis, concurrency, verification, synthesis, type, compiler 등의 분야에서 비슷한 비율로 논문이 있었다. 특히 type에 대한 논문이 생각보다 많았다. 앞서서도 얘기했지만 POPL기간 내내 다른 분야에도 꾸준히 관심을 가지고 어떤 이슈들이 있는지 알아가야 하겠다는 생각을 하지 않을 수 없었다. 20분의 발표가 의미있으려면 그 분야에 대한 기본적인 지식은 필수일것 같다.

## 2.3 논문발표들

이번 학회에서 재밌게 들었던 논문들을 간략히 요약해 보자면 다음과 같다

### -An Abstract Domain to Discover Interval Linear Equalities, VMCAI

예전에 중국에서 몇명이 Cousot교수님의 group으로 AI를 전수받으러 갔다고 들은적이 있다. 이 논문은 그 중 Liqian Chen이란 박사과정 학생이 Antoine Mine와 함께한 연구였다. google해보니 2008년부터 APLAS, SAS, VMCAI에 abstract domain관련 논문을 계속 발표해 오고 있었다. 이번에는 Interval Linear Equalities (intLivEqs)라는 새로운 abstract domain을 디자인 하였다. 기존의 affine equality domain( $a_1 x_1 + \dots + a_n x_n = b$ )의  $a_k$ 와  $b$ 들을 interval로 확장한 것이다. 이러한 확장으로 인한 도메인 연산들을 정의하였다. 이 때 각 연산이 polynomial-time으로 만들기 위해서 constraint system을 row echelon form으로 표현한다. 이렇게 확장된 도메인은 affine equalities나 convex polyhedra가 표현하지 못하는 invariant를 표현할 수 있게 된다. 하지만 talk에서 이러한 도메인이 왜 중요하고 필요한지에 대한 전달이 충분치 않았고 기존의 interval polyhedra 등의 도메인들과의 비교등이 부족했다. 또한, domain 표현 및 operation정의의 detail보다는 바탕이 되는 직관적인 원리 위주로 설명했더라면 더 이해가 쉬웠을 것 같다.

### -Invariant and Type Inference for Matrices, VMCAI

multi-dimensional array(matrice)를 순차적으로 도는 다중루프(nested loop)의 quantified invariant를 구하는 방법에 대한 논문이다. 또한 계산된 invariant를 이용해서 matrice의 모양을 유추한다. invariant를 구하는 방법은 먼저 nested loop를 동일한 의미를 나타내는 single loop의 집합으로 쪼갬 후, 쪼개어진 각각의 single loop의 invariant를 기존의 방법을 이용하여 구한 후 conjunction으로 결합한다. 그리고 미리 주어진 matrix type predicate을 만족하는지 theorem prover를 이용하여 유추를 시도한다. 찾은 invariant로 matrix모양을 유추하는 시도가 흥미로웠다.

### -Shape Analysis in the Absense of Pointers and Structure, VMCAI

higher-order program에 대한 shape analysis가 무엇일까에 대한 논문이었다. C-like language의 pointer analysis는 higher-order program의 CFA로 비유할 수 있다. 그렇다면 C program의 shape analysis는 higher-order program의 무슨 분석인가? Matthew Might은 environment analysis가 이것에 대응되는 것이라고 하면서 왜 그런지, 그것으로 무엇을 할 수 있는지 설명하였다. 함수는 environment에 따라서 "shape"이 변한다. 즉, 의미가 달라진다. 그러므로 higher-order program에 대한 shape analysis, 즉 environment analysis는 함수의 의미를 결정하는 것이라고 할 수 있다. 기존의 CFA는 environment를 무시한 채 closure (code+env)의 반쪽 (code part)만 분석하는 것이므로 차이가 있다. 실제로 기존의 CFA가 분석하지 못하는 성질의 문제들이 컴파일러 최적화를 위해서 필요하다 (예를 들어 environment분석없이 function inlining을 정확히 할 수 없다). 이 talk에서는 기존의 environment problem을 일반화한 문제를 소개하고 해결방법을 간략히 보였다. 내용도 흥미로웠지만 발표가 아주 훌륭했다. 문제 도입부터 직관적인 해결책까지 흐름이 아주 매끄럽고 흡입력이 있었다.

### -An Analysis of Permutations in Arrays, VMCAI

어떤 자료구조의 global invariant 분석을 제시한 논문으로, array contents가 바뀌지 않음을 보이는 문제로 특화시켜서 설명하였다. 예를 들어 array A가 array B의 permutation인가를 분석한다. 이 논문에서 이러한 문제를 non positional property라고 부른다. 전체적인 아이디어는 간단하다. array의 global state를 multisite of array contents로 보고 각 프로그램 포인트 별로 multisite equation을 세우고 푸는 것이다. 그런데 이 때 array cell의 내용이 변경되므로 어떤 array cell들이 어떤 인덱스 변수들로 인해서 aliasing되어 있는지의 여부를 알아야 한다. 구체적으로는 index 변수들간의 관계, index와 array contents의 관계가 필요하다. 분석을 진행하면서 이 정보들도 부가적으로 유지한다. 이슈는 방정식을 풀면서 두 multisite equation을 join하는 것이다. 이 논문의 main topic으로 두 가지 방법을 제시하였다. 기존에 보기 힘든 array의 permutation property를 분석하는 것 자체가 흥미로웠고, 간단한 분석으로 작동하는 것도 재미있었다. 다만, array update 시에 발생가능한 여러 상황들을 고려하기 위해서 인덱스 변수관계등의 추가적인 분석결과들을 필요로 하는데, 본 분석과 추가적인 분석을 조화시키는 부분이 깔끔하진 않은 것 같다는 생각이 들었다. 하지만 이러한 추가적인 분석으로 본래의 문제가 간단해지는 경향이 있어서 꼭 나쁜것 같지만은 않다.

### -From Boolean to Quantitative Notions of Correctness, POPL

Thmoas A. Henzinger의 invited talk이다. 시스템의 옳고그름(correctness)을 표현할 때 보통 참/거짓으로 기술하기 마련인데, 정성적(quantitative)인 옳고그름을 엄밀하게 정의하는 연구에 대한 내용이었다. 프로그램을 비롯한 시스템의 동작은 보통 올바르게나 틀렸거나로 나뉜다. 하지만 옳고그름을 단순히 참/거짓으로 나누기 보다는 어떤 성질에 얼마나 가까운지에 대한 것으로 표현할 때 더 유연한 경우가 많이 있다. 기존의 참거짓으로 표현되던 것을 정성적인 의미로 일반화하고 확장할 때 나타나는 이슈들을 어떻게 정의해 나가고 있는지에 대해 알 수 있었다.

### -Continuity Analysis of Programs, POPL

프로그램의 입력이 작은 값만큼 변했을 때, 출력도 작은 값만큼 변하는게 보장되는지를 분석하는 방법에 대해서 소개했다. GPS와 같은 장비의 입력은 항상 일정한 값이 들어오기 보다는 주변환경 등등의 영향으로 작은 값을 범위로 동요하게 되는데, 이 때 출력값도 안정적인 값 범위에서 바뀔것인지를 검증하는 것이 중요할 것이다. 이 논문에서는 이러한 문제를 프로그램의 robustness라 부른다. 이 논문에서는 이러한 성질을 증명할 수 있는 proof rule을 제시하고 관련된 증명과 중요한 몇몇 프로그램의 연속성을 제시한 룰들로 증명할 수 있음을 실험적으로 보였다. Swarat Chaudhuri는 이 룰들을 만드는데 있어서의 몇 가지 난점들을 소개하고 그것들을 어떻게 해결했는가를 위주로 발표했다. 그 중에서 연속성이라는 성질이 루프에 대해서 귀납적으로 정의되지 않기 때문에, 귀납적으로 정의되도록 루프의 iteration들을 묶는 과정이 필요했다는 것이 기억에 남는다.

### -A Simple, Verified Validator for Software Pipelining, POPL

내용은 별로 알아듣지 못했지만 Xavier Leroy가 직접 발표를 해서 기억에 남는다. 대표적인 OCaml 개발자로 OCaml을 쓰다보면 Xavier Leroy가 짜 놓은 라이브러리등의 소스 코드를 많이 보게 마련이다. 이번에 발표한 논문은 software pipeline을 검증하는 것에 관련된 것이었다. Software pileining으로 인해서 변형된 루프가 원래의 루프와 의미적으로 동일한지에 대해서 검증하는 것이었는데 software pipelining이 보통 상당히 복잡하게 루프를 변형하기 때문에 이 문제가 중요하다고 한다. 슬라이드가 간단했고 발표도 대가답게 매우 여유있고 부드러웠다. 디테일보다는 개념적인 내용들 위주로 강의를 듣는 것 같은 느낌이 들었다. 발표 후 질문도 많았다. 내가 잘 알아듣지 못한점만 뻘다면 상당히 인상적인 발표였다.

### -Compositional May-Must Program Analysis: Unleashing the Power of Alternation, POPL

Interprocedural analysis에서 함수의 summary를 그때그때(demand-driven) 만드는 경우 may와 must summary를 서로 잘 섞어서(alternation) 사용하면 속도가 더 빨라지는 경우가 있고 실제 코드를 분석하는데 효과적임을 보인 논문이다. 즉, 함수 f의 must summary를 만드는데 있어서 f가 호출하는 함수 g의 must summary만을 이용하기보다는 경우에 따라서 may summary를 이용하면 f의 must summary를 더 빠르게 만들 수 있는 경우가 있다는 것이다 (그 반대도 가능). 예를 들어서 f의 must summary를 만드는 중에 g의 must summary가 없어서 g를 다시 분석해야 하는 경우에, 경우에 따라서 g의 모든 패스를 살펴보아야 하는 경우가 있을 수 있는데, 만약 g의 may summary가 이미 만들어져 있었고 그 summary가 현재 관심있는 property에 대해서 말해주고 있다면 g의 must summary를 새로 만들 필요 없이 이미 만들어져 있는 may summary를 활용할 수 있다. 발표를 들으면서 든 생각은, 맞는 말이긴 한데 그러한 경우(may-must summary를 서로 서로 사용해서 좋은 일이 일어나는 경우)가 얼마나 되겠느냐는 의문이 들었다. 제한적인 경우에만 그런일이 일어날 것 같은데 (왜냐하면 내가 관심있는 property가 "우연히(앞으로 사용될지 모르고)" 만들어 놓았던 summary에 표현되어 있는 경우에만 적용가능하므로) 실제 코드에서 이런일이 얼마나 일어날지가 궁금했다. 그런데 실험결과를 보니 의외로 실제코드를 분석할 때 상당히 효과적이었다. may/must 둘 중 하나만 이용하는 분석에 비해서 같이 이용하는 경우가 두 배 이상 빨랐다.

### -Higher-Order Multi-Parameter Tree Transducers and Recursion Schemes for Program Verification, POPL

Naoki Kobayashi 교수님이 작년 APLAS에서 invited talk을 했을 때 재미있게 들었는데 그 내용에 대한 확장논문이었다. 이전 연구가 higher-order functional program을 검증(verification)하는 새로운 방법에 대한 제시였다면 이번 논문에서는 그것이 가지고 있던 제한점을 해결하여 좀 더 일반적인 방법을 제시하고 있었다. 이 방법이 흥미로웠던 이유는 higher-order function(함수를 인자로 받는 함수)의 검증은 일반적으로 생각하기 어려운데, 이를 recursion scheme이라는 것으로 매우 자연스럽게 표현되기 때문이다. 그리고 higher-order 프로그램의 검증은 프로그램을 recursion scheme으로 변환한 후 이를 모델체킹 하는 것이다. 이를 위한 검증 알고리즘에 대해서 작년 POPL에서 발표했고 이번 POPL논문에서는 기존의 결과를 확장하여 좀 더 넓은 도메인으로 이 방법을 적용할 수 있게 한 것이었다. recursion scheme으로 표현되지 않던 성질(주로 infinite domain)들을 표현하기 위해서 새로운 모델을 정의하고 verification은 작년에 제시된 알고리즘에 의존하였다. APLAS때와 마찬가지로 Kobayashi의 발표 스타일도 기억에 남는다. 영어가 유창하다기 보다는 군더더기 없이 짧게 짧게 핵심적인 요소들만 짚어서 말을 해서인지 전반적으로 알아듣기 쉽고 깔끔했다. 내용도 키 아이디어를 잘 보여주는 예제 위주로 구성하여 전체적인 아이디어가 어떻게 동작하는 것인지 이해하기 쉬웠다.

### 3. 스페인

낮선 곳을 방문한다는 것은 낯설음 하나만으로도 즐거운 일인 것 같다. 내가 사는 곳에서 수천-수만 킬로미터 떨어진 땅, 그리고 그곳에 사는 사람들의 모습을 직접 피부로 느낄 수 있다는 것 자체가 생각해보면 상당히 신기한 일이다. 스페인은 그런 느낌이 더욱 강하게 느껴진 곳이었다. 스페인의 점심식사는 2시쯤, 저녁은 9시쯤 시작된다. 그 사이에 간식류를 자주 먹으며 저녁은 다른사람들과 바르(bar)에서 이야기를 하면서 간단히 먹는다. 저녁은 거의 술잔에 안주 정도를 먹는다. 낮선 식사시간때문에 몸이 적응을 못하는지 스페인에 있는 동안 점심을 2시에 먹고 저녁에는 배가 고프지 않은 경우가 종종 있었다. 편차는 심하지만 음식값은 대체적으로 비싼 편인 것 같았다. 안주가 아닌 음식을 먹으려면 대체로 한끼에 10유로 이상씩 든다. 영어가 정말 안통하는 나라지만 사람들은 말이 통하든 안통하든 별로 신경쓰지 않는 것 같다. 밤에 음식점을 찾아 해매면서 만난 어떤 아저씨는 우리가 알아듣는 말든 스페인어와 아주 약간의 영어로 자신의 축구 사랑을 늘어놓았다. 중국 교수님들과 함께 갔던 스페인 본토 레스토랑에서도 음식점 종업원과 말은 전혀 통하지 않았지만 메뉴를 고를 때부터 후식을 먹을 때까지 계속 대화하며

즐거운 시간을 보냈다. 톨레도로 가는 기차에서도 우리자리를 차지한 스페인 할아버지 할머니들이 자리는 비켜주지 않으면서도 우리 자리를 확보해 주기위해서 분주히, 오래토록 참견해 주었던 것도 기억에 남는다.

