

Trip Report on POPL 2014

January 22 – 24, 2014, San Diego, CA, USA

이원열 (leewy@postech.ac.kr)

머리말

1월 21일 화요일 저녁, 미국 샌디에고로 출발하였다. 다른 일정 때문에, POPL 이전에 열리는 PLMW, POPL Tutorial, VMCAI 등에는 참석하지 못하고 POPL에만 참석하게 되었다. 미국으로 가는 비행기에서 제대로 자지 못하고 호텔에서조차 깊이 잠들지 못하는 바람에 시차적응을 하는데 상당히 애를 먹었다. (결국 한국에 돌아올 때쯤이 되어서야 겨우 시차적응을 할 수 있었다...) '시차적응이 잘 되었더라면 좀 더 많은 발표를 집중해서 듣고 더 유익한 시간을 보낼 수 있었을 텐데'라는 아쉬움이 많이 남는다.

POPL 2014

* POPL 2014 & ICDM 2012

POPL 2014는 내가 처음으로 참가하는 PL 분야 학회였다. 3일동안 아침부터 저녁까지 계속 진행되었으며, 올해에는 역대로 가장 많은 수인 51편의 논문이 발표되었다. 이전에 데이터마이닝 (DM) 분야 학회인 ICDM 2012에 참석했었는데, 그 때랑은 분위기가 사뭇 달랐다. 우선, PL community가 DM community보다는 상대적으로 작기 때문에, POPL 2014에 참석한 사람의 수뿐만 아니라 발표된 논문의 수와 동시에 진행되는 세션의 수도 더 적었다. (POPL 2014에서는 동시에 2개의 세션만 진행되었지만, ICDM 2012에서는 4~5개의 세션이 동시에 진행되었다.) 또한, 휴식시간이 POPL 2014에서는 45분으로 ICDM 2012에서의 30분보다 좀 더 길었으며, ICDM 2012와는 다르게 POPL 2014에서는 점심식사를 호텔 안에서 제공하였다. 아마도 참석자들 간의 토론을 활발히 만들고, 이것이 점심시간까지도 자연스럽게 이어질 수 있도록 하기 위한 목적이었다고 생각된다.

* 활발한 대화와 열띤 토론

학회기간 내내 참석자들끼리 서로 활발한 대화와 열띤 토론을 끊임없이 주고 받는 모습이 이번 학회에서 가장 인상 깊게 느껴졌던 점이다. 발표장이 두 개 밖에 없었기 때문에 모든 발표는 호텔의 한 층에서만 진행되었는데, 휴식시간만 되면 거의 모든 사람들이 발표장 밖의 홀로 나가서 시끌벅적하게 이야기를 나누었다. 생전 처음 보는 광경이었다. 홀이 그렇게 크기 않았기 때문에 사람들로 가득 붐볐으며 45분의 긴 시간 동안 이러한 풍경이 지속되었다. 특히 유명하신

교수님이나 연구자분들 주위에는 항상 많은 사람들이 몰려서 그 분들과 서로 이야기를 하고 싶어하는 듯한 느낌을 받았다. 또한 발표가 끝난 뒤 질문을 하고 싶다면 양 옆 통로에 설치된 마이크 앞으로 직접 가서 말해야 하는데도, 마이크 앞에 줄을 서면서까지 많은 사람들이 흥미로운 질문들을 던졌다. 질문자가 너무 많아서 다 질문을 하지 못하고 다음 발표로 넘어간 적도 종종 있었다. 이렇듯 3일간의 학회 기간 동안 참석자들의 연구에 대한 엄청난 열기를 느낄 수 있었다.

* 학회 프로그램

POPL이 PL 분야의 최고 학회인 만큼 발표 하나하나를 제대로 이해하기에는 무리가 있었다. 특히 어떤 세션에서는 introduction에서부터 이해할 수 없는 용어들이 난무하는 경우도 있었다. 역시나 아직도 배워야 할 것이 정말 많은 것 같다.

다행히 몇몇 세션에서는 어느 정도 이해는 할 만 내용의 발표가 있었다. 개인적으로 Type System Design 세션과 Separation Logic 세션이 가장 흥미로웠고 많은 부분을 알아들을 수 있었다. 첫 날 첫 번째 순서였던 Type System Design 세션에서는 polymorphic function, Haskell with Interface, dependently typed language에 관한 발표가 있었고, 각각이 재미있는 내용들을 다루고 있었다. 둘째 날 마지막 순서였던 Separation Logic 세션에서는 separation theory, proof theory for separation logic 등에 관한 발표가 있었으며, separation logic에 대해 배경지식이 있었기에 발표들을 수월하게 이해할 수 있었다. 특히, 작년 POPL에 이어서 올해 POPL에서도 separation logic만을 다룬 세션이 있었다는 점이 눈 여겨 볼만 하다. separation logic에 대한 사람들의 관심도가 점점 더 높아져가는 것 같았다.

POPL 바로 전 주에 있었던 ROSAEC 워크샵에서 처음으로 abstract interpretation과 probabilistic programming에 대해서 알게 되었는데, 이번 POPL에서도 이와 관련된 많은 논문들이 발표되었다. 특히, Probability라는 세션 자체가 있었다는 것이 신기했다.

I. Type System Design

1. Polymorphic Functions with Set-Theoretic Types. Part 1: Syntax, Semantics, and Evaluation --- Giuseppe Castagna

이번 학회의 가장 첫 발표였으며, PL 연구실에서 PhD를 받으신 현승이형도 같이 쓰신 논문이어서 가장 집중도 있게 들은 발표 중 하나였다. 이 연구의 간략한 동기는 다음과 같다.

```
map f l = case l of
  | [] -> []
  | (x : xs) -> (f x : map f xs)

even x = case x of
  | Int -> (x `mod` 2) == 0
  | _ -> x
```

우선, 1) 위와 같은 프로그램을 정의할 수 있으며, 2) `map`의 type이 $(\alpha \rightarrow \beta) \rightarrow [\alpha] \rightarrow [\beta]$ 이고 `even`의 type이 $(\text{Int} \rightarrow \text{Bool}) \wedge ((\alpha \setminus \text{Int}) \rightarrow (\alpha \setminus \text{Int}))$ 인 것을 유추할 수 있으며, 3) 더 나아가 `(map even)`의 type이 $([\text{Int}] \rightarrow [\text{Bool}]) \wedge ([\alpha \setminus \text{Int}] \rightarrow [\alpha \setminus \text{Int}]) \wedge ([\alpha \vee \text{Int}] \rightarrow [(\alpha \setminus \text{Int}) \vee \text{Bool}])$ 인 것까지 유추할 수 있는 calculus를 만드는 것이다.

좀 더 정확하게 말하자면, <higher-order polymorphic function>과 <union, intersection, negation이 포함된 recursive type>을 지원하는 lambda-calculus를 이 논문에서 제시하였다. 특히, 3번이 가능하도록 하는 것이 매우 어려운 문제이며, substitution과 instantiation이 복잡하게 적용되기 때문에 상당히 힘든 문제였다고 한다. 아쉽게도 뒷부분에 설명된 자세한 사항들은 잘 이해하지 못했다.

2. Combining Proofs and Programs in a Dependently Typed Language --- Chris Casinghino

지금까지의 dependently-typed language는 모든 expression이 terminate하도록 강제하거나 (Coq, Agda 등), 아니면 무한 루프를 허용하는 대신 logic으로서는 inconsistent 했었다 (Haskell 등). 이 논문에서는 이 두 가지 종류의 언어가 가지는 장점들을 어떻게 하면 합칠 수 있는가에 대해서 논의하고 있다.

제안하는 언어는 logical fragment와 programmatic fragment로 구성된다. logical fragment에서는 모든 expression이 terminate해야하지만, programmatic fragment에서는 그럴 필요가 없다. 예를 들어 다음과 같은 `not` 함수는 logical fragment에서 정의되지만, `div` 함수는 종료되지 않을 수 있기 때문에 programmatic fragment에서만 정의될 수 있다:

```

log not : Bool -> Bool
not b = if b then False else True

prog div : Nat -> Nat -> Nat
rec div n m = if lt n m then 0
              else plus 1 (div (minus n m) m)

```

이 논문에서 제안하는 언어의 특징적인 점은, programmatic function에 대한 사실들을 증명해낼 수 있다는 것이다. 예를 들어, `div` 함수에 대한 다음과 같은 사실을 증명할 수 있다 (이는 F* 등에서는 불가능하다):

```

log div63 : div 6 3 = 2
div63 = refl

```

평소에 dependently-typed language에 관심이 있었기에, 시간이 날 때 이 논문뿐만 아니라 이와 관련된 연구들을 공부해보고 이들이 어떤 제한점을 가지고 있는지도 생각해보면 좋을 것 같다.

II. Separation Logic

3. Proof Search for Propositional Abstract Separation Logics via Labelled Sequents --- Zhe Hou

Points-to 등이 없고 대신 propositional variable만 있는 <abstract한 separation logic>의 proof system을 제안한 논문이다. 우리 논문과 비슷하게 labelled sequent를 사용하였지만, 우리와는 접근 방법이 조금 달랐다. 우리는 BBI의 sequent calculus와는 상관없이, separation logic을 위해서 완전히 새로 proof system을 만들었다. 하지만, 이 논문에서는 BBI와 sound & complete한 sequent calculus에서 시작해서 sound한 rule들을 추가하면서 이를 확장하였다.

아직 points-to 등을 지원하지 않긴 하지만, 우리 system과는 다르게 어느 정도 modular한 강점을 가지고 있는 system이라고 생각된다. 사용하려는 application에 따라서 약간씩 다른 semantics of separation logic이 필요한데, 우리 system을 변형해서 이러한 여러 semantics를 모두 지원하기는 힘들 것이다. 하지만, 이 논문의 방식을 비슷하게 따른다면 sound한 rule들을 조금씩 다르게 추가해서 각각의 semantics와 complete한 system들을 쉽게 얻는 것이 가능할 것이다.

4. A Proof System for Separation Logic with Magic Wand --- Sungwoo Park

지난 1년간 교수님과 함께 연구한 것의 최종 결과물이다. 발표는 교수님께서 하셨고, 나는 첫 날 저녁에 있었던 포스터 발표를 맡았었다. 역시나 교수님께서도 우리 논문의 핵심 내용을 간결하면서도 이해하기 쉽게 발표를 해주셨다. 발표가 끝나고서 질의응답 시간에는 생각보다 많은 분들께서 질문을 해주시고 comment를 해주셨다. 작년에 열심히 연구한 결과에 많은 사람들이 관심을 가져주어서 내심 뿌듯했다. 그리고 교수님께서 (작년 POPL 발표에 이어서 이번에도) 맨 마지막 슬라이드에 한국어로 중요한 내용을 적으셨는데, 뒤에 앉아있던 외국인들이 "저 korean이 무슨 뜻이지?"라고 궁금해하기도 했다 :) 발표자로서 POPL에 참석할 수 있도록 더욱 노력해야겠다!

III. Invited Talk

5. From the Trenches: Static Analysis in Industry --- Andy Chou

셋째 날 invited talk에서는 Coverity라는 회사의 Cofounder이자 CTO인 Andy Chou가 자신의 회사에 대해서 발표를 했다. Andy Chou는 Stanford CS에서 PhD 학위를 받는 도중, 자신이 연구하던 내용을 바탕으로 <프로그램 정적 분석 툴>을 만드는 Coverity라는 회사를 cofound했다고 한다. Stanford CS PhD 학생 네 명과 Dawson Engler 교수가 함께 창업한 이 회사는 현재 1100 customer를 가지고 있다고 한다.

Static analysis에 대해서 조금이나마 자세하게 알게 된 것이 ROSAEC 워크샵이었기 때문에, 이것이 실제 industry에서 얼마나 유용하게 사용되고 있는지에 대해서는 전혀 감이 없었다.

하지만 발표를 들으면서 실제 소프트웨어 개발을 하는데 있어서 static analysis 기술이 상당히 유용하게 이용될 수 있음을 느꼈다. 단순한 copy-and-paste 오류에서부터 시작해서 memory leak 등을 꽤나 정확하게 잡아줄 수 있는 그런 분석 툴이었다. 또한 코드의 크기가 큰 linux kernel도 빠른 시간 안에 분석이 가능하다고 한다. PL 분야가 이론에만 머무르지 않고 practical하게 사용되는 예를 직접 보게 되니까 감회가 새로웠다.

맺음말

학회 기간 동안 다양한 것들을 배울 수 있었지만, 여러 사람들도 만날 수 있었다. 우선, 서울대 이광근 교수님과 허충길 교수님을 비롯하여, ROPAS 연구실의 김진영 형과 강동욱 형도 만날 수 있었다. 또한, 우리 연구실을 졸업하시고 그 동안 말로만 전해 들던 현승이 형도 뵈 수 있었다. 안타깝게도 학회 마지막 날 저녁에 먼저 LA로 가봐야되서, 이 분들과 함께 저녁 식사를 하지는 못했다.

LA에 가서는 칼텍에서 3일정도 머무르다 왔다. 칼텍에서 재학 중인 고등학교 동기가 있어서, 잠깐 신세를 지고 그 근처를 구경했다. 특히, 칼텍 학생이랑 같이 캠퍼스 투어를 하니 건물 내부까지 들어가볼 수 있어서 더욱 유익한 투어였다! 포스텍을 설립할 때 모델로 삼았다는 칼텍을 직접 돌아보니까 그 느낌이 포스텍과 상당히 비슷했다. (단, 칼텍에 잔디가 훨씬 많고, 포스텍과는 다르게 모두 평지였다는 점은 제외하고.) 특히나 2009년에 지어졌다는 CS 건물은 정말로 인상적이었다! (학회장에서 찍은 사진이 없어서 칼텍 사진을 대신 첨부했다.)

이번 학회 참석을 가능하게 해주신 박성우 교수님께 감사 드립니다.

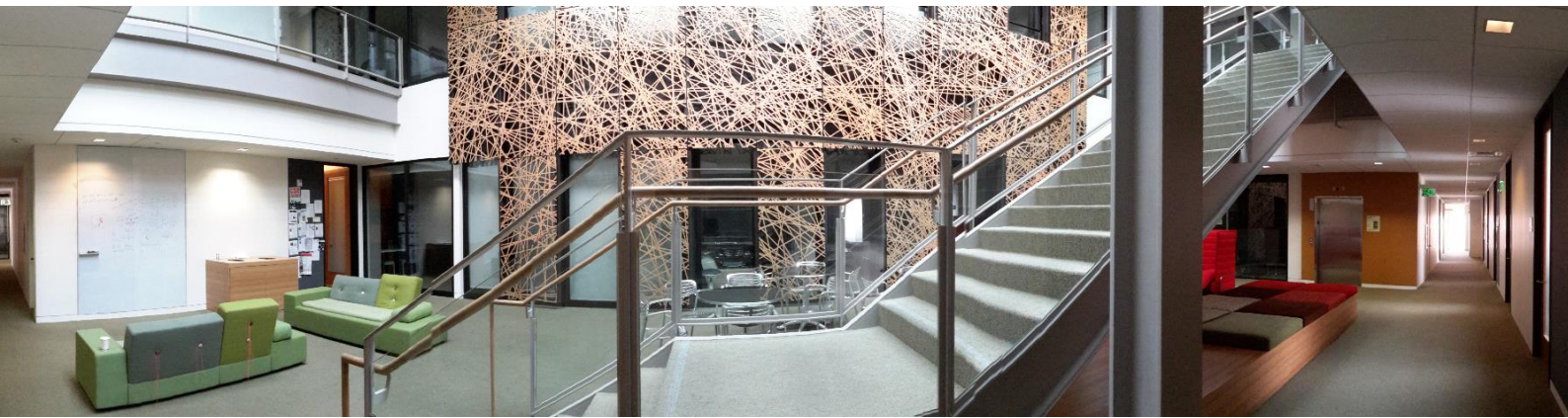


그림 1. 칼텍 (Caltech) Annenberg Center의 외부와 내부 사진 (총 3층).

다른 학교들의 CS Department에 해당하는 CMS (Computing + Mathematical Sciences) Department의 건물이다. 2009년 10월에 완공되었으며, 건물의 모든 면이 투명한 창문으로 둘러싸여있어서 굉장히 멋있다! 뿐만 아니라 최근에 지어져서 그런지 건물 내부도 상당히 쾌적하며 멋있고, 휴식을 취할 수 있는 공간이 많이 있다. 특히 건물 내부의 벽 곳곳에 화이트보드가 설치되어 어디서든 자유롭게 토론을 할 수 있도록 되어있다. 칼텍에서 가장 인상 깊었던 건물이었다! 우리나라에도 이런 멋진 CS 건물이 지어졌으면 좋겠다 :)