
APLAS 2014

싱가포르

2014년 11월 17일 - 2014년 11월 19일
서울대학교 허기홍



싱가포르 국립대학에서 열린 APLAS 2014 학회에 참가하였다. 추운 한국을 벗어나 따뜻한 날씨도 즐기고 훌륭한 음식도 맛보면서 즐거운 시간을 보냈다. 프로그래밍 언어의 다양한 분야에서 많은 연구자들이 자기들이 맞닥뜨린 문제와 해결책을 이야기하는 가운데 나도 포스터 발표자로 참가하게 되어 더욱 뜻깊은 학회였다. 2009년 서울대에서 열린 APLAS 에서, 혹은 그 이후 여러 학회에서 안면이 있는 사람들도 보여서 친숙한 분위기였다.

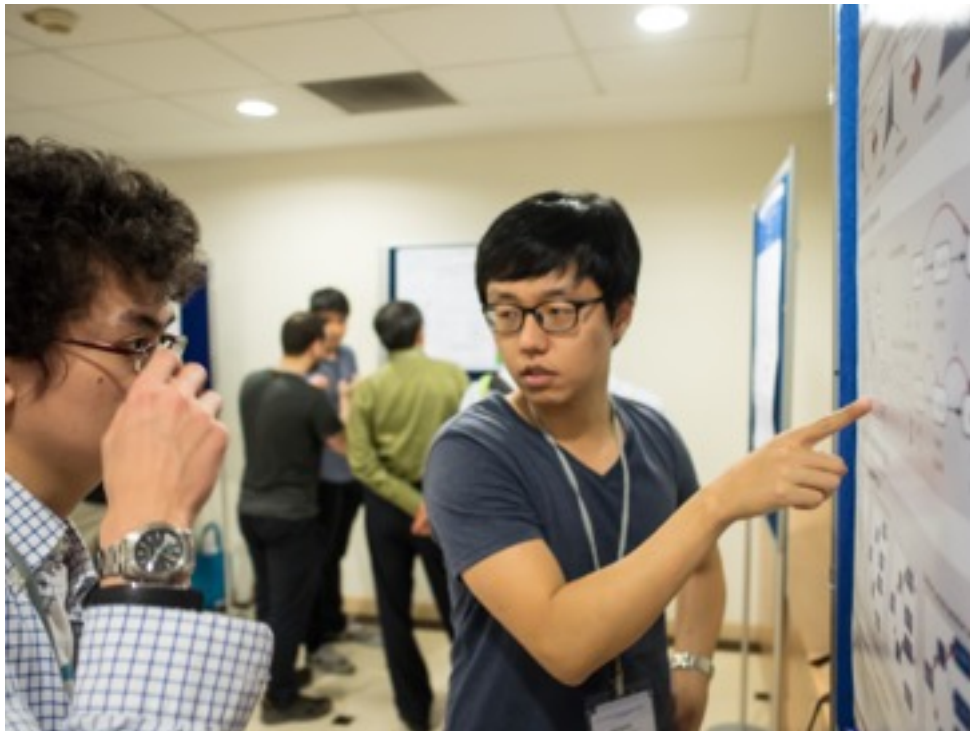
포스터 발표

Sound, Precise & Scalable Static Analysis by Sparsity and Selectivity

허기홍, 오학주, 이광근

지난 몇 년 동안 연구한 결과를 한데 모아서 포스터 발표에서 소개하였다. 그동안 연구실 동료들과 안전하면서도 빠르고 정확하게 정적 분석을 하는 기술을 만들고 PLDI'12, PLDI'14, TOPLAS'14 등 에서 좋은 평가를 받았는데 이를 큰 틀에서 쉽게 설명하려고 노력했다. 이 내용으로 이전에도 여러 번 발표를 해보았고, 아이디어는 아주 간단하였기 때문에 큰 무리 없이 내용을 전달할 수 있었다.

ROSAEC 센터의 포스터 발표장처럼 모두들 다과를 먹으면서 즐거운 분위기였다. 심지어 첫날 오후 두세시쯤이었음에도 학회측에서 맥주를 제공해주어서 더욱 열띤 토론이 이어졌다. 정적분석기를 만들어본 경험이 있는 사람들이 많이 관심을 보였고 자기네들 분석기에 적용해 보겠다는 사람도 있어서 재미있었던 발표였다.



초청 발표

Incremental Adoption of Static-Typing

Julien Verlaquet, Facebook

요즘 세계에서 가장 주목받는 IT 기업중 하나인 페이스북에서 정적 타입 (static type) 을 도입한 경험담을 이야기해 주었다. 원래는 페이스북도 여느 웹 기반 소프트웨어 회사들처럼 PHP 를 주로 이용했다고 한다. 하지만 PHP 같은 동적 타입 언어는 그때 그때 손쉽게 프로그램을 만들기는 쉽지만 쉽게 버그가 발생하고 버그를 찾기도 어렵다. 이를 해결하기 위해 정적 타입 시스템을 이용하기로 결정을 내렸다고 한다.

정적 타입 시스템을 회사 전체에 자연스럽게 도입하기 위하여 페이스북에서는 자체적으로 HACK (<http://hacklang.org>) 이라는 언어를 만들었다. 이 언어는 우선 PHP 에 뿌리를 두고, 호환도 된다. 또한, HACK 프로그램에서는 정적 타입 코드와 동적 타입 코드가 동시에 존재할 수 있다. 이런 특징 때문에 기존 PHP 코드를 한 번에 갈아치울 필요가 없고, 서서히 HACK 으로 바꾸어 갈 수 있다.

거대 공룡 기업의 문제 해결력을 볼 수 있었다. 회사의 생산성 향상을 위해서 관성에 얽매이지 않고 스스로 과감하게 문제를 해결하는 모습이 인상적이다.

NetKAT - A Formal System for the Verification of Networks

Dexter Kozen, Cornell University

이 발표에서 소개하는 NetKAT 은 네트워크 시스템을 기술하고 검증하는데 쓰는 프로그래밍 언어이다. 특화된 하드웨어 장치들로 구성하던 전통적인 네트워크와 달리, 최근에는 범용 장치에다가 특정 기능을 소프트웨어로 만든 네트워크 (software-defined network) 가 널리 쓰인다고 한다. 이때 네트워크를 기술할 수 있는 프로그래밍 언어가 필요한데 여기서 말하는 NetKAT 이 그 중 하나이다.

NetKAT 의 특징 중 하나는 Kleene 대수를 기반으로 만들어진 언어라는 점이다. Kleene 대수는 잘 알다시피 정규식의 성질을 기술하는 전통적인 방식이다. 네트워크는 거시적으로 보았을때 큰 오토마타라고 여길 수 있기 때문에 오토마타를 기술하는 Kleene 대수로 네트워크를 표현하는 방식은 지극히 자연스럽다. 하지만 이것으로는 부족하고, 미시적으로는 스위칭 프로세스를 잘 기술할 방법이 필요하다. 이를 위해서는 참/거짓을 판단하는 장치가 필요한데, 이 때문에 도입한 것이 KAT (Kleene Algebra with Test) 라는 대수이다. 이 또한 오래전부터 사용된 전통적인 기술이다. NetKAT 은 이 KAT 을 기반으로 하고, 네트워크를 효과적으로 만들고 검증하는데 사용될 수 있다.

교과서에 나올 정도로 보편적이고 전통적인 기술을 다시 되살려 최신 기술에 도입한 아주 멋진 연구라고 생각한다. 교과서에 실릴 정도로 잘 정리된 이론이면 이해하기도 쉽고 남에게 설명하기도 쉬울 것이다. 우리가 하는 연구에도 그럴 가능성은 없을지. 늘 기본에 충실할 필요가 있다.

Automatic Memory Management Based on Program Transformation Using Ownership

Tatsuya Sonobe, Kohei Suenaga, and Atsushi Igarashi

C 같이 메모리 관리를 수동으로 해야하는 언어로 작성된 프로그램에서 메모리 누수 (memory leak) 가 발생하는 부분을 찾아 메모리를 해제하는 코드 (예, free()) 를 자동으로 삽입해주는 기술이다. 핵심은 순서를 구분하고 (flow-sensitive) 하고, 각 포인터 변수의 메모리 소유권 (ownership) 을 나타내는 타입 시스템을데 이는 2009년 APLAS에서 발표를 들었던 기억이 난다. 이 타입 시스템에서 각 포인터 변수의 타입은 그 포인터가 가리키는 메모리 영역의 타입 뿐만 아니라 그 메모리 영역의 소유권 정보까지 내포하고 있다. 소유권이란 해당 메모리 영역을 가리키는 포인터 변수의 개수에 따라 달라지는 0과 1사이의 실수이다. 만약 한 포인터가 유일하게 한 메모리 영역을 가리킨다면 그 포인터는 그 메모리영역의 전적인 소유권 (= 1) 을 갖는다 (아래 그림 1,2번 줄). 그렇지 않고 여러 포인터가 영역을 동시에 가리킨다면 1보다 작은 실수로 소유권을 나누어 갖게된다. free() 로 메모리 영역이 해제되면 포인터는 해당 메모리 영역의 소유권을 놓는다 (아래 그림 4,5번 줄).

Program	Type env. after each instruction
0 :	<i>/* ∅ */</i>
1 : <code>let x = malloc() in</code>	<i>/* x:int ref₁ */</i>
2 : <code>let y = malloc() in</code>	<i>/* x:int ref₁, y:int ref₁ */</i>
3 : <code>*x ← y;</code>	<i>/* x:int ref₁, y:int ref₁ */</i>
4 : <code>free(x);</code>	<i>/* x:int ref₀, y:int ref₁ */</i>
5 : <code>free(y)</code>	<i>/* x:int ref₀, y:int ref₀ */</i>

그런데 만약 어떤 포인터가 소유권을 놓아야 할 지점에서조차 여전히 소유권을 갖고 있으면 메모리 누수라고 판단하고 그 부분에 free() 명령을 삽입하는 것이다.

메모리 누수 문제를 타입으로 풀려는 시도가 특이해 보였다. 간단한 아이디어가 재미있었지만, 구현과 실험이 없어서 성능이 궁금한 연구이다.

A Method for Scalable and Precise Bug Finding Using Program Analysis and Model Checking

Manuel Valdiviezo, Cristina Cifuentes, and Padmanabhan Krishnan

이 논문은 모델 체킹과 정적 분석을 결합하여 효과적으로 버그를 찾는 방법을 제시한다. 보통, 모델 체킹은 정확한 반면 큰 프로그램을 다루지 못하고, 정적 분석은 큰 프로그램을 빠르게 분석할 수 있지만 정확도가 떨어지는 단점이 있다. 이 논문에서 제시하는 방법은 정적 분석을 통해 모델 체킹 문제 크기를 줄이는 것이다. 먼저 정적 분석을 사용해서 버그가 예상되는 후보를 빠르게 선별한다. 그리고 각 후보와 관계되는 부분만 프로그램을 도려낸다. 그리고 그렇게 도려낸 각 프로그램 조각들에다 모델체킹을 적용하여 빠르고 정확한 결과를 기대하는 것이다.

전통적인 두 분석 기법의 장점을 잘 결합한 재미있는 연구라고 생각한다. 저자들은 오라클에서 일하는 사람들인데 아무래도 큰 소프트웨어를 다루는 거대 회사이다보니 빠르고 정확한 분석에 대한 간절함이 느껴진다.

싱가포르

싱가포르는 일년내내 30도 안팎을 유지하는 적도 부근의 따뜻한 도시국가이다. 아시아의 네마리 용이라고 해서 우리나라와 같이 신흥 경제 강국으로 꼽히기도 했고, 범죄자에게는 가혹한 처벌을 내리는 것으로도 유명하다. 숙소는 도심 한복판, 우리로 따지면 서울 명동쯤 되는 곳에 있었는데 그 거리는 끝도 없이 명품 매장이 이어져 있었다. 유럽을 가보면 각 도시마다 건축 양식이나 역사적 상징물 등이 그 도시의 색을 결정하는데, 그에 비해 아시아 국가의 색깔은 그냥 고층 빌딩인 듯하다. 서울이나 홍콩이나 북경이나 싱가포르나 큰 차이를 느끼지 못하겠다. 싱가포르에는 옥상 수영장으로 유명한 마리나베이샌즈 호텔이 명소인것 같은데 호화스럽기만 할 뿐 깊이는 느껴지지 않는다. 출장기간이 짧고 대부분 학회장에서 보냈기 때문에 싱가포르의 진가를 못보고 왔을 수도 있다.

학회의 운영은 정말 훌륭했다. 학회에 많이 참석한 것은 아니지만 다녀본 학회중에 최고라 말할 수 있겠다. 학회장소인 싱가포르 국립대와 숙소를 오가는 셔틀 버스를 매일 제공했고, 아침, 점심, 저녁과 간식까지 제공해 주었는데 맛이 기가 막혔다. 쌀국수 등 동남아 음식을 비롯하여 인도 음식, 채식주의자를 위한 음식도 마련해 주었다. 저녁 만찬과 산책 프로그램도 학회를 더욱 즐겁게 한 요소였다.

마무리

학회는 갈때마다 좋은 자극을 받고 돌아온다. APLAS 는 아시아 학회이지만 전세계에서 PL 을 공부하는 훌륭한 사람들을 만날 수 있었다. 출장때마다 지원을 아끼지 않으시는 지도교수 이광근 교수님과 소프트웨어 무결점 연구센터 행정실에 감사드린다.

