



ETAPS 2011

European Joint Conferences on Theory & Practice of Software

March 26 - April 3, Saarbrücken, Germany

ETAPS는 유럽에서 열리는 학회들(CC, ESOP, FASE, FoSSACS, TACAS)을 포함하는 큰 학회이다. 올해에는 무려 18개의 위성 워크샵들을 거느리고 있다. 참석인원도 약 500명 가량 된다. 이렇게 큰 학회를 같은 연구실에 있는 이원찬 연구원과 함께 다녀왔다.

Invited Talks

전반적으로 초청 발표 (invited talk)가 많고 수준은 높아서 유익했다. 대가들의 깊은 이해에서 나오는 명쾌한 설명이 돋보이는 발표가 많았다. 인상 깊었던 초청 발표들을 소개하겠다.

The Search for Structure in Quantum Computation

Prakash Panangaden (McGill, Canada)

APLAS때도 양자 계산 (Quantum Computation)에 대한 발표를 들었는데 그때와는 다르게 (^_^) 무척 재밌게 들었다. 옛날 사람들이 항상 구조를 잘 갖춘 (structured) 프로그램만을 생각하다가 병행성 (concurrency) 불결정적인 (nondeterminism)을 생각할 때 느꼈을 당황스러움을 양자 계산이라는 새로운 계산 모델이 나오면서 우리가 비슷하게 당황하는 것이라한



다. 실제로 정말 당황스럽다 양자계산은... 상자 두개가 있을때 두 입자가 다른 상자의 있을 확률은 $1/2$ 인데 양자 계산에서는 $1/3$ 이거나 1 이라고 한다. 도대체 어떻게? 정확하게 설명은 해주지 않아서 모르지만 그렇다고 한다. 양자 계산을 이해하려면 직관을 버려야 한다는... 양자 계산을 사용하는 두가지 방법으로 사람들이 흔히 생각하는 것은 알고리즘의 계산 속도를 올리기와 통신에서 보안문제라고 한다. 항상 빠르게 계산하는 것만을 양자 계산의 장점이라 생각했기에 보안문제가 양자 계산과 관련이 있는지 몰랐다. 발표자는 여기에 지역적이지 않은 (non-local) 양자 계산의 특성을 분산 연산에 사용할 수 있다고 했다. 그러면서 강조한 사실! 양자 컴퓨터를 만들 수 있는가 없는가는 중요하지 않다. 새로운 계산 모델을 어디에 어떻게 적용할 지 알아보는 것 자체가 의미 있는 일이라고 했다. 양자 모델의 놀라운 점! 1) 중첩 (superposition) 하나의 시스템이 두가지 상태를 동시에 가질 수 있고, 이는 확률적인 합이 아니다. 그러니까 $1/2$ 로 A 상태 $1/2$ 로 B 상태를 갖는다는 그런 종류가 아님. 말 그대로 동시에 두가지 상태를 가질 수 있고 어떻게 관찰하느냐에 따라 우리가 그 상태 중에 한가지를 볼 수 있는 것이란다. 게다가 그 관찰하는 행위가 그 상태를 변화시킬 수 있다는 잘 알려진 사실. 따라서, 양자 계산에는 지역적인 (local), 인과관계가 뚜렷한 (causal), 결정적인 상태에 기반하는 논리는 존재할 수 없다고 한다. 그 후에 기술적인 내용을 많이 설명했는데 category theory 관련 내용이라 자세히 이해하지는 못했다. 하지만, 재밌는 예 하나는 이해했다. 정보의 순간이동 (teleportation). 일반적인 통신과는 다르게 내가 아는 정보를 다른 사람이 가져가면 내가 가진 정보는 더 이상 내가 알던 정보가 아니다. 정보의 이동만 가능하고 복사가 불가능한 통신체계를 만들 수도 있다. 양자 계산에 관한 내용은 들으면 들을수록 어렵지만 흥미롭다.

Future-proofing collections: from mutable to persistent to parallel

Martin Odersky (EPFL, Switzerland)

Scala 언어를 처음으로 접하는데 인상적이었다. Scala는 객체 지향 프로그래밍 언어와 함수형 프로그래밍의 요소가 결합된 언어이다. 같은 일을 하는 코드를 여러 언어에 대해서 비교해서 보여 주었는데 Scala로 구현된 코드의 크기가 전체적으로 짧았다. 또, 최신의 프로그래밍 언어답게 멀티 프로세서를 적극적으로 활용할 수 있도록 언어가 디자인 되었다는 느낌을

받았다. 병렬 프로그래밍은 피할 수 없이 어려운데 이 것을 Scala 언어의 collection을 이용하여 잘 지원하도록 하는 방법을 이야기 했다. Collection은 효율적으로 구현되었고, 거의 모든 타입에 대해서 지원을 한다. 예를 들어 `val (minors, adults) = people.partition(_.age < 18)`는 직관적으로 알 수 있듯이 people이 collection 타입이면 주어진 조건 함수를 사용하여 같은 타입의 두 minors, adults로 나누는 일을 한다. Scala라는 이름이 scalable 언어에서 나왔다고 한다. 짧은 코

드로도 많은 일을 할 수 있는 훌륭한 언어로 보인다. 참고로 우리 연구실에 있는 Bruno Oliveira가 가장 좋아하는 언어이기도 하다.

An Automata-Theoretic Approach to Program Analysis

Andreas Podelski (Freiburg, Germany)

Andreas Podelski 교수님이 대가라는 것을 다시 한번 느낄 수 있는 발표였다. 요약 해석 기반의 정적 분석과 모델 체킹의 차이를 보여주는 슬라이드가 가장 기억에 남는다. 교수님 설명에 따르면 프로그램 분석을 가르치는 학교를 두 종류로 나눌 수 있는데, 어떤 학교에서는 프로그램의 실행 의미를 정의하고 그것의 요약 의미를 정의해서 요약 의미로 프로그램을 분석한다. 반면, 다른 학교는 프로그램 자체를 어떤 요약된 프로그램 혹은 오토마타로 요약 변환을 한 후에 그 요약된 대상을 분석한다. 전자가 요약 해석에 기반한 정적 프로그램 분석이고, 후자가 소프트웨어 모델 체킹이다. 이 발표에서는 후자의 한 방법으로 프로그램을 오토마타로 변환하여 프로그램의 안전성과 종료성을 분석하는 방법을 설명했다. 내가 특히 관심이 있었던 것은 프로그램 종료성 문제를 해결하기 위해 사용한 전이 불변성 (transition invariant)이었다. 왕교수님이 우리의 알고리즘적인 학습으로 불변성 찾기의 후속 연구 주제로 발표 전에 제안했었다. 프로그램의 상태에 대한 불변성으로는 프로그램의 종료성을 증명할 수 없기에 프로그램의 전이 상태에 대한 불변성을 찾고, 그 전이 불변성이 바닥이 있는 관계인지 (well-founded relations) 검사함으로써 프로그램의 종료성을 증명한다. 여기서 바닥이 있는 관계는 전이의 한계 (bound)가 있고, 그 한계에 유한번의 전이만에 도달하는 관계를 뜻한다. 예로, $x > 1 \wedge x' < x - 1$ 는 바닥이 있는 관계이다. 여기서, x 는 입력 상태에서의 x 의 값을 x' 은 출력 상태에서의 x 의 값을 뜻한다. x 의 값은 항상 1보다 큰 차이로 감소를 하고 x 의 값의 최소 한계가 1이기 때문이다.



Automated Design and Verification of Security Protocols based on Zero-Knowledge Proofs

Michael Backes (Saarbrücken, Germany)

아무것도 모르고 증명하기 (zero-knowledge proof) 라는 것을 알게 되어서 유익했다. 이것은 예를 들어 A가 어떤 비밀을 알고 있다고 할때 B가 A가 그 사실을 실제로 알고 있는지를 검증하는 것을 말한다. 다만 검증을 할때 B가 A가 그사실을 알고 있다는 사실을 제외한 그 어떤 정보도

알아내서는 안된다. 또, 이 검증 시스템은 안전하고 완전해야 한다. 실제로는 거의 불가능하기 때문에 아주 작은 확률로 안전성이 깨지는 것은 감수한다. 이런 시스템은 전자투표와 같이 익명성이 보장되어야 하는 곳에 꼭 필요하다. 이 발표에서는 그런 시스템이 주어졌을 때 그 위에서 어떻게 안전성이 보장되는 프로토콜을 자동으로 만들어 낼 지에 대해서 설명을 했다. 발표자료가 직관적이고 체계적이었다. 발표자에게 메일을 보내서 자료를 요청하려고 한다.

Talks

아쉽게도 일반 논문 발표는 초청발표와 다르게 발표를 못 해서인지 내용이 없는 것인지 뭔가 훌륭한 일을 했다는 느낌을 주는 발표가 너무도 적었다. 그저 자신이 계속 해오던 조금씩 잘라서 논문 한편 한편으로 만든다는 느낌. 논문을 위한 논문. 뭔가 유용한 것은 한참이 지나서야 나오겠구나 생각이 들게 만드는... 또, 멋있어 보이는 연구 주제는 몇몇 그룹이 나눠서 짬 해두고 다른 그룹의 침범은 녹록지 않은 분위기. Andrew Appel 교수님, Viktor Kuncak 교수님, Patrick Cousot 교수님, Naoki Kobayashi 그룹등등 많은 그룹들은 뭔가 앞으로 할 일이 쭉 정해져 있고, 그 그룹에 속하면 계속 그 일을 하고 좋은 곳에 논문을 낼 수 있어 보인다. 오히려 다른 그룹, 이름이 생소한 그룹에서 나온 논문들이 더욱 관심이 가는 것은 이유는 무엇일까? 저런 주도적인 연구 그룹들을 부러워 하는 것인지 맘에 안 들어 하는 것인지 나조차도 모르겠다.

이렇게 큰 학회에서 중요한 학회들에서 열리는 발표들을 보면서 과연 저 중에 정말 사람들을 기분 좋게 하고 연구하는 사람들도 가슴이 뛰는 연구가 얼마나 될까 생각을 했다. 굳이 일반인들까지 행복하게 하지 못 하더라도 연구하는 사람들한테라도 인정을 받고 기분을 좋게 하는 연구가 몇이나 될까? 나의 연구는 어떤가? 내 스스로도 가슴 뛰게 연구를 하고 있는가 하는 의문이 들었다.

그렇다, 주도적인 그룹들이 자신들이 하고 있는 연구를 하면서 가슴이 뛰고 보람을 느끼며 다른 연구자들에게 자랑스럽다면 부러워할만한 것이고, 단지 논문을 위해서 연구를 하고 있다면 부러워할 필요가 없으리라. 막연하게나마 가슴 뛰는 연구를 하고 싶다는 절실함을 가지게 되었다.



TACAS (Tools and Algorithms for Construction and Analysis of Systems)

Predicate Generation for Learning-Based Quantifier-Free Loop Invariant Inference

Yunbum Jung, Wonchan Lee, Bow-Yaw Wang, and Kwangkeun Yi

TACAS에 실린 우리 논문이다. 이전 VMCAI 논문에서 요약 (predicate abstraction)에 쓰이는 요약대상들 (predicate)을 프로그램 코드로부터 가져오던 것을 자동으로 생성하는 방법을 제시한 논문이다. 프로그램 코드상에 없는 새로운 요약대상들을 만들 수 있다는 것과 필요한 만큼만 만든다는 것이 장점이다. 발표는 왕보요 교수님이 했다. 같은 세션에 묶였던 다른 발표가 도구 논문 (tool paper)여서 그런지 듣는 사람들이 생각보다 많지 않았다. 발표시간에 쫓겨서 발표를 하신 것도 아쉬움으로 남는다.

Loop Summarization and Termination Analysis

Aliaksei Tsitovich, Natasha Sharygina, Christoph M. Wintersteiger, and Daniel Kroening

종료 분석에 관한 논문이다. Andreas Podelski 등등이 만든 Terminator와 다른 점은 종료 조건을 한번에 만든다는 것이다. Terminator는 한가지 경로가 끝난다는 것을 보이고, 그 것과 같은 조건으로 끝나는 다른 경로들까지 한꺼번에 끝난다는 것을 보인다. 계속 실행가능한데 아직 포함되지 않은 경로를 하나 뽑아서 다시 종료 조건을 찾고 확장해나간다. 만약 모든 실행 경로가 끝난다고 하면 그제서야 그 프로그램이 끝난다고 결론을 내린다. 그리고, 이 과정이 전체시간의 대부분을 차지한다. 이 논문은 그 과정을 한번에 끝내고 싶어한다. 방법은 루프의 실행 의미를 포함하는 루프 요약 (Loop Summarization)을 만들어서 다시 루프를 분석하는 일이 없도록 하는 것이다. 이것을 계속 적용해나가면 결국 프로그램 전체가 루프를 포함하지 않는 순차적인 프로그램이 된다. 재밌는 것은 논문에서 SNU 벤치마크 프로그램이라고 나와서 서울대랑 이름이 같다고 생각했는데 실제로 서울대학교 민상렬 교수님 연구실에서 만든 벤치마크 프로그램이었다. 반가운 일이다.

Efficient CTMC Model Checking of Linear Real-Time Objectives

Benoit Barbot, Taolue Chen, Tingting Han, Joost-Pieter Katoen, and Alexandru Mereacre

Katoen 교수님의 발표는 처음 들었는데 정말 발표를 잘 하셨다. 내용부터가 전혀 배경 지식이 없는 나에게는 전혀 생소한 CTMC (continuous-time Markov chain)을 DTA (deterministic timed automata)에 대해 증명을 할 때 어떻게 하면 효율적으로 하는 가였다. Katoen 교수님이 훌륭하다는 사실은 왕교수님께 익히 들어 알고 있어서 논문을 좀 읽어보려고 했으나 너무 어려웠다. 그래서, 포기하고 있었는데 발표를 환상적으로 잘하셔서 마치 이해한 듯한 느낌을 주었다. 결국 지금 다시 봐도 전혀 이해가 가지 않지만 발표를 듣는 중에는 정말 어려운 문제를 훌륭하게 해결했다는 확신을 받았다. 정말 신기한 일이다. 다음에 또 발표를 들을 기회가 있다면 논문을 다시 한번 제대로 이해하고 들어보고 싶다.

Semantics of Concurrent Revisions

Sebastian Burckhardt and Daan Leijen

프로그램 관리 시스템 (예, SVN, CVS, git)에서는 하나의 프로그램이 다양한 용도로 쓰이기 위해서 중간에 가지를 치고 나중에 여러 가지의 프로그램들이 하나로 합쳐지기도 하는데 이런 시스템의 의미를 기술하는 프로그래밍 모델에 관한 논문이다. 또, 이 프로그래밍 모델을 가지고 병렬프로그램의 의미가 결정적 (deterministically)이고 일관성 있도록 (consistently) 기술한다. 내용은 사실 간단해서 이해하기가 쉽다. 사실 병렬적으로 다르게 변경되는 두 프로그램의 실행 결과를 어떻게 결정적으로 합치는지가 중요한데, 서로 겹치는 부분은 연산을 지연시켰다가 합쳐질때 한꺼번에 같이 처리하기에 크게 어려워 보이는 점이 없다. 다만 처음으로 이런 시도를 한 점은 높이 사야할 것 같다.

Linear Absolute Value Relation Analysis

Liqian Chen, Antoine Mine, Ji Wang, and Patrick Cousot

학회에서 벌써 세번째로 만나는 친구 Liqian Chen의 또 다른 도메인 관련 논문이다. 이 친구는 새로운 도메인만 연구하고 있다. 발표가 끝나고 무슨 연구를 하고 있다고 물어보니 또 다른 도메인을 연구하고 있다고 한다. 저번 APLAS의 여행기에서도 썼지만 사실 이런 도메인이 일반적으로 쓰이는 것도 아니고 특정 프로그램 몇개에 대해서만 의미가 있을텐데 좀 비판적인 생각이 드는 것도 사실이다. 다면 (polyhedral) 도메인은 볼록한 (convexity) 형태만을 표현할 수 있다는 한계를 가진다. 절대값 $|x| = 1$ 의 해가 $x = -1, 1$ 과 같이 non-convexity를 표현할 수 있다는 점에 착안해서 linear absolute value inequalities라는 도메인을 만들었다. 그래서, 꼭 이 도메인이 필요한 프로그램의 불변성을 찾았다는 것이 논문의 요지이다.

Typing Local Control and State Using Flow Analysis

Arjun Guha, Claudiu Saftoiu, and Shriram Krishnamurthi

자바 스크립트와 같이 정적 타입이 없는 언어에서 기존에 잘 돌아가는 프로그램들이 타입 검사가 되지 않는 경우는 제외하면서도 어느 정도 타입 검사를 해주는 방법을 제안하고 있다. 방법은 실행 중에 만족해야 하는 타입의 조건을 코드에 자동으로 주석으로 달고, 함수 내부에서는 간단한 흐름 분석을 통해 그 주석들이 타입과 잘 맞아떨어지는 지를 검사한다. 이후에 함수를 호출하는 부분에서는 그 함수의 타입만을 보고 타입의 조건을 달고 검사한다. 방법이 깔끔하고 발표를 잘 해서 마음에 드는 논문이다.

Probabilistic Points-to Analysis for Java

Qiang Sun, Jianjun Zhao, Yuting Chen

학회에 와서 제일 처음 들었던 논문이다. 기존의 포인터 분석과는 다르게 포인터가 어떤 주소를 가리킬 수 있는 확률을 분석하는 방법을 제시한다. 들어보면 그 확률로 알아서 무엇을 하겠다는 생각이 든다. 발표자는 컴파일러의 최적화를 더 해보기 위한 분석이라고 한다. 예를 들어, 중복되는 코드를 모험적으로 없애기 (speculative redundancy elimination) 같이 확률이 어느 정도 이상이면 없앴다가 실체가 아니면 다시 이전으로 돌아가서 계산을 진행하는 최적화까지 생각한다는 것이다. 어저지같다는 느낌을 받았다. 실험 결과에서는 확률 포인터 분석이 기존의 포인터 분석보다 약간만 시간과 메모리만을 더 쓰면서 더 정확하다는 (저자들이 정의한 측정법에 따라) 내용만을 포함하고 있다. 이 연구가 의미있다는 것을 보이려면 당연히 이 분석을 통해 얻은 정보를 가지고 최적화를 더 해서 생성된 최종 코드가 실제로 더 빨라진다는 것을 보여야 하는 것 아닌가? 그리고, 분기문에서 실행흐름이 어디로 갈지에 대한 확률을 알아야 분석 결과가 의미가 있을텐데 `if (p != null)`와 같은 분기문에서 `p`가 `null`이 아닐 확률을 도대체 무슨 수로 알아낸단 말인가?

Faster Alias Set Analysis Using Summaries

Qiang Sun, Jianjun Zhao, Yuting Chen

요약 (Summary)를 이용한 분석은 항상 나의 관심을 끈다. 내가 박사과정 중에 많은 시간을 들여 만든 메모리 누수 탐지기가 함수 요약 기반의 분석기이기 때문이다. 이 논문에서는 포인터 분석을 함수 요약을 가지고 한다. 한가지 특이한 점은 호출되는 함수와 호출하는 함수의 요약 두 가지를 쓴다는 점이다. 호출하는 함수의 요약은 분석 속도가 너무 느려지는 경우에 정확도를 잃으면서 어림잡기 위해 사용한다. 그런데, 사실 호출되는 함수의 요약은 내가 ISMM'08에 쓴 논문에서 제시하는 방법과 굉장히 유사하다. 하지만, 아쉽게도 참고 문헌에는 나의 논문이 없었다. 나 스스로도 참고해야 할 논문을 모르고 빠뜨린 적이 있기 때문에 이해가 가지 않는 것은 아니지만 마음이 상했다.



MPI에서의 발표

MPI에서 연구하고 있는 허충길 박사님의 소개로 원찬이와 나는 MPI-SWS에서 각자의 연구주제를 발표했다. 도착한 날 MPI 사람들과 볼링을 치러 가서 즐거운 시간도 보냈다. 학회가 끝난 후에는 주말이 시작되어 어쩔 수 없이 학회 중에 발표를 하게 되었다. MPI연구소는 두 곳으로 나뉘어 있는데 한 쪽에서 세미나가 열리면 다른 쪽에서도 화상을 통해 참석할 수 있게 되어 있다. 발표자는 청중이 앉는 쪽 벽에 있는 스크린을 통해 저쪽에 있는 청중을 볼 수 있고, 그 쪽 사람들도 언제든지 궁금한 것이 있으면 질문을 할 수 있다. 나는 왕교수님과 지금까지 한 일들을 한데 묶어서 발표를 했다. 듣는 사람들 대부분 재밌게 들었던 것 같은데 특히 Viktor Vafeiadis가 관심을 보이고 질문을 여러번 했다. 가장 기억에 남는 좋은 질문은 알고리즘 학습의 질문을 답할 수 없을때 간단하게 동전 던지기를 통해 답을 하는데 만약 확률을 50대 50으로 하지 않고, 확률을 한 쪽으로 치우치게 하면 실험 결과가 어떻게 되냐는 것이었다. 사실 이 질문은 예전에 카이스트의 정교민 교수님도 했었다. 확실히 실험을 통해서 확인을 해볼만한 문제이다. 만약 우리가 찾는 루프의 불변성들의 분포가 제공한 over-approximation에 보다 가깝다면 멤버십 질문에 대해 예를 더 많이 대답하는 쪽이 더 빨리 해를 찾을 확률을 높일 것이다. BLAST를 만드는 데 기여한 Rupak Majumdar가 학회 발표를 들으러가서 내 발표를 듣지 않았다.

원찬이는 암시적 프로그래밍 (implicit programming)의 핵심 (core calculus)을 찾는 연구에 관해 발표했는데 많은 사람들의 관심을 받았다. 특히 Derek Dreyer는 이해가 무척 빨랐다. 열심히 듣는 것 같지도 않았는데 원찬이 발표를 중간에 다 이해하고 질문을 했다. 또, 논문의 방향에 대해서도 비판적이면서도 도움이 되는 이야기를 많이 해주었다. Derek Dreyer는 홈페이지에서 볼 때보다 더 나이가 많아 보였는데 실제 나이는 80년생으로 나와 동갑이라고 해서 놀랐다. 근데 벌써 POPL에 6개 ICFP에 5개나 논문이 있다. 17살부터 박사학위를 시작했다고 한다. 이런 사람들을 만나면 내가 계속 학계에 있어도 되는지 회의적인 생각이 들기도 한다.

Banquet

Banquet으로 UNESCO 세계 문화 유산으로 지정된 오래된 제철소에 갔다. 쇠 냄새가 은은하게 퍼지는 곳에서 연주가들이 클래식 음악을 연주하며 우리는 식사를 했다. 전반적으로 독일 음식치고는 짜지 않아서 먹을만 했다. 원래는 학생으로 등록을 해서 100유로를 따로 내고 티켓을 구매해야만 했지만 충길이형이 바쁘다고 감사하게도 티켓을 쥐서 갔다 올 수 있었다. 충길이형이 이 외에도 여러모로 도와줘서 편하게 다녀 올 수 있었다. 충길이형에게 정말 고맙다는 말을 전하고 싶다.

