2010 ROSAEC CENTER NEWSLETTER

소프트웨어무결점 연구센터 뉴스레터

No . 1 April , 2010

rosaec.snu.ac.kr

교육과학기술부 / 한국 연구재단 지정 우수연구센터

ROSAEC center

Research On Software Analysis for Error-free Computing

소프트웨어 무결점 연구센터 NRF ERC

목차

1 인사말

안녕하십니까.

저희 소프트웨어무결점연구센터(ROSAEC Center, Research On Software Analysis for Error-free Computing)는 교육과학기술부와 한국연구재단 우수연구센터(ERC) 지원사업을 통해 2008년 9월 시작된 연구센터 입니다.

무결점 소프트웨어가 시장에서 절실히 필요해지는 이유를 따로 부연할 필요는 없을 것입니다. 더군다나 요즈음 "녹색 기술"이라는 화두에 컴퓨터 소프트웨어를 대입해 보면, 그 핵심 요소기술에는 오류 없는 소프트웨어를 제작하는 기술이 있게 됩니다. 낭비 없이 안전하게 지속하는 세상은 그 세상을 지탱시키는 소프트웨어들의 무결점과 직결되기 때문입니다.

저희의 목표는 비유하자면 "소프트웨어 MRI", "소프트웨어 fMRI", "소프트웨어 PET"등을 연구개발하는 것입니다. 저희는 오류 없는 소프트웨어를 저렴하게 생산 가능하게 하는 원천기술의 한 축을 세계적으로 선도하려고 합니다. 그래서 이 기술을 기반으로 실용적인 소프트웨어 소스 오류 자동 검출 및 검증 도구들을 개발할 것입니다. 그리고 이 도구들을 순수 소프트웨어 개발뿐 아니라 지능형 로봇/무인 비행체/금융 공학/인공 위성/의료 기기등의 소프트웨어에 특화시키는 과정을 밟으려고 합니다.

많은 관심과 격려를 부탁드립니다.

감사합니다.

센터장 이광근.
서울대학교 컴퓨터공학부 교수

2
센터 연구소개

센터소개

제대로 작동할 지를 미리 검증할 수 없는 기계설계는 없다. 제대로 서 있을지를 미리 검증할 수 없는 건축설계는 없다. 인공물이 자연세계에서 문제없이 작동할지를 미리 엄밀하게 분석하고 검증하는 기술은 잘 발달해 왔다. 이러한 분석 검증 기술은 다른 엔지니어링 분야에서는 당연히 확립된 기술이다. 왜냐하면 모든 엔지니어링의 제일 근본적인 질문이 바로, 우리가 만든 것이 우리가 의도한대로 움직인다는 것을 어떻게 확인할 수 있는가? 이기 때문이다.

컴퓨터 소프트웨어에 대해서는 어떤가? 작성한 소프트웨어가 제대로 실행될지를 미리 엄밀하게 확인해 주는 기술은 있는가? 그래서 작성한 소프트웨어가 가질 수 있는 오류를 자동으로 미리 모두 찾아주거나, 없으면 없다고 확인해 주는 기술은 있는가? 그래서 소프트웨어의 오류 때문에 발생하는 개인/기업/국가/사회적 비용을 절감시켜주는 기술은 있는가?

오류 검출 및 검증 기술의 첨단에서 실용가능한 기술을 규명해내고 남들보다 먼저 산업화해내는 그룹은 확산 직전에 있는 거대한 무결점 소프트웨어 개발 도구 시장의 주도권을 잡게 될 것이다.

01 SW 분야의 근본문제

작성한 SW의 오류를 자동으로 미리 모두 찾아주거나, 없으면 없다고 확인해주는 기술들은 있는가?

02 무결점 SW의 시장가치

점점 막대해지는 SW 오류의 비용 (개인/기업/사회/국가)
미국: 60조원/년
모든 제품 경쟁력 = SW품질
가전,기계,통신,교통,국방,의료,에너지
제품 리콜, 사회 혼란, 국가 손실
SW 제작중 오류수정에 쓰는 비용 : 20조원/년
2010 SW 시장 : 310조원

03 센터의 목표

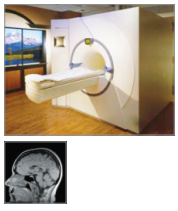
원천기술 선도 정적분석 기술 심화 + 혁신 신기술 연구

실용적 도구 개발 SW 오류 자동 검출/검증기 개발

특화된 산업화 순수SW, 지능형 로봇, 무인 비행체, 금융 공학에 특화시켜 산업화

04 센터의 목표 도구

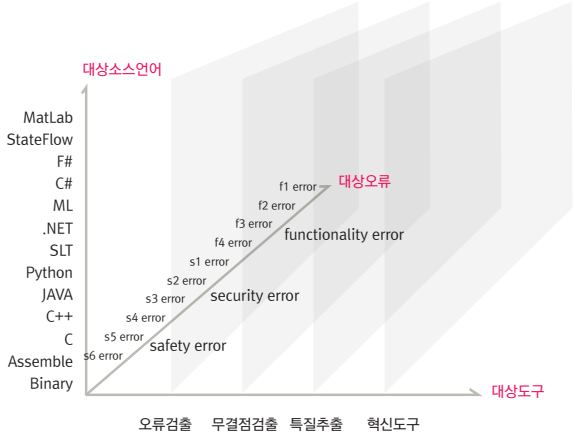
소프트웨어 MRI



소프트웨어 fMRI

소프트웨어 PET

05 목표 도구의 공간



06 세부 초점사항 : 원천기술

정적분석(static analysis)

이 기술은 엄밀히 예측한다.

이 기술은 테스팅의 단점을 보완한다.

이 기술의 실용성은 확인되었다.

산업화 완료한 SPARROW를 통해

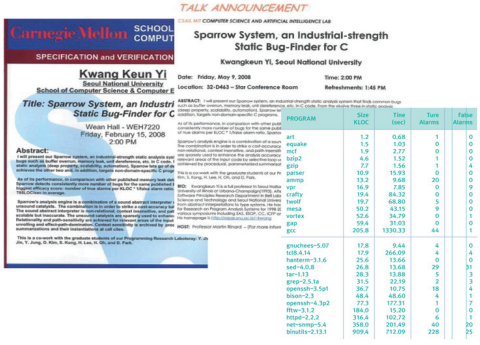
static analysis, abstract interpretation, type system, program logic, theorem proving, model checking

www.spa-arrow.com

Sparrow
The Early Bird

07 원천기술 실용성 예시 : SPARROW

대상 : C, memory leak/buffer overrun,
오류 검출율 6/KLoc, 속도 10Loc/sec


TALK ANNOUNCEMENT

Carnegie Mellon SCHOOL COMPUT

SPECIFICATION and VERIFICATION

Kwang Keun Yi
Seoul National University
School of Computer Science & Computer E

Title: Sparrow System, an Industri
Static Bug-Finder for C

Wean Hall - WEH7220
Friday, February 15, 2008
2:00 PM

Abstract:

Sparrow System, an Industrial-strength
Static Bug-Finder for C
Kwangkeun Yi, Seoul National University

Date: Friday, May 9, 2008
Location: 32-D463 – Star Conference Room
Time: 2:00 PM
Refreshments: 1:45 PM

ABSTRACT:

PROGRAM	Size KLOC	Time (sec)	Ture Alarms	False Alarms
art	1.2	0.68	1	0
equake	1.5	1.03	0	0
mcf	1.9	2.77	0	0
bilp2	4.6	1.52	1	0
gzip	7.7	1.56	1	4
parser	10.9	15.93	0	0
ammp	13.2	9.68	20	0
vpr	16.9	7.85	0	9
crafty	19.4	84.32	0	0
twolf	19.7	68.80	5	0
mesa	50.2	43.15	9	0
vortex	52.6	34.79	0	1
gap	59.4	31.03	0	0
gcc	205.8	1330.33	44	1
gnuchess-5.07	17.8	9.44	4	0
tcl8.4.14	17.9	266.09	4	4
hanterm-3.1.6	25.6	13.66	0	0
sed-4.0.8	26.8	13.68	29	31
tar-1.13	28.3	13.88	5	3
gzep-2.5.1a	31.5	22.19	2	3
openssh-2.5pl	36.7	10.75	18	4
bison-2.3	48.4	48.60	4	1
openssh-4.3p2	77.3	177.31	1	7
ffhw-3.1.2	184.0	15.20	0	0
httpd-2.2.2	316.4	102.72	6	1
net-snmp-5.4	358.0	201.49	40	20
binutils-2.13.1	909.4	712.09	228	25

08 5대 선도 프로젝트(flagship projects) 중심

국가 성장동력 산업과 융합

시스템 SW 오류 자동 검증 도구셋 개발

웹 및 그래픽 SW 오류 자동 검증 도구셋 개발

지능형 로봇 SW 오류 자동 검증 도구셋 개발

무인 비행체 SW 오류 자동 검증 도구셋 개발

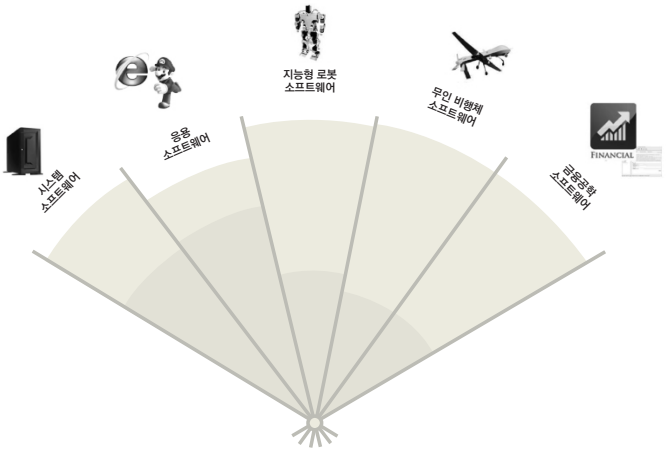
금융공학 SW 오류 자동 검증 도구셋 개발

09 총괄과제 구성 및 연계

1총괄 : 정통 첨단 기술팀 (leading analysis tech.)

2총괄 : 도메인 특화 기술팀 (domain-specific analysis tech.)

3총괄 : 이론 및 혁신 기술팀 (theory & innovative tech.)



- 오류 및 특질 검출기 (bug-finder, property-finder)
- 무결점 검증기 (verifier)

1총괄 : 정통 첨단 기술팀 (leading analysis tech.)

2총괄 : 도메인 특화 기술팀 (domain-specific analysis tech.)

3총괄 : 이론 및 혁신 기술팀 (theory & innovative tech.)

10 센터의 차별성

SW 원천 기술 연구센터
있는 기술의 조합, 산업화 센터가 아님

SW 실용기술 선도센터
이론 논문 센터가 아님

SW 고부가 산업창출 원동력센터
기존 산업 보조 센터가 아님

SW 융합,혁신 연구 추진센터
우물안의 센터가 아님

SW 기초 첨단 인력 양성센터
꽃을 가꾸는 센터가 아님

3
대표논문과 이야기

Deriving Invariants by Algorithmic Learning, Decision Procedures, and Predicate Abstraction

Yungbum Jung, Soonho Kong, Bow-Yaw Wang, and Kwangkeun Yi.
Proceedings of the 11th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'10). Madrid, Spain. January 2010.

지난 7~9월간 센터를 방문했던 Bow-Yaw Wang 교수(Academia Sinica, Taiwan)와의 공동연구 이야기입니다.
좋은 연구를 짧은 기간에 같이하게 된 아주 즐거운 경험이었습니다.
Algorithmic Learning이라는 분야를 프로그램 검증에 사용할 수 있는 시초를 만든 연구를 같이 하고 갔습니다.
키워드는 algorithmic learning, randomness, invariant inference입니다. 정영범, 공순호 학생과의 팀웍이 완벽했습니다.
연구를 하면서 "이거다"싶은 느낌이 가끔 있습니다. 그 경우였습니다.
내년 1월 VMCAI(Verification, Model Checking, Abstract Interpretation) 학회에서 발표합니다.
논문리뷰중에 "The paper presents a fresh/novel perspective to invariant inference and has a potential to lead to a new line of subsequent work in this direction." 라는 평을 받았습니다. 드문 리뷰 기분 좋았습니다. 논문 마지막 손보기위해 모여 앉았던 이른 아침의 캠퍼스 벤치, 그 추억이 소중합니다.

Adaptive Scratch Pad Memory Management for Dynamic Behavior of Multimedia Applications

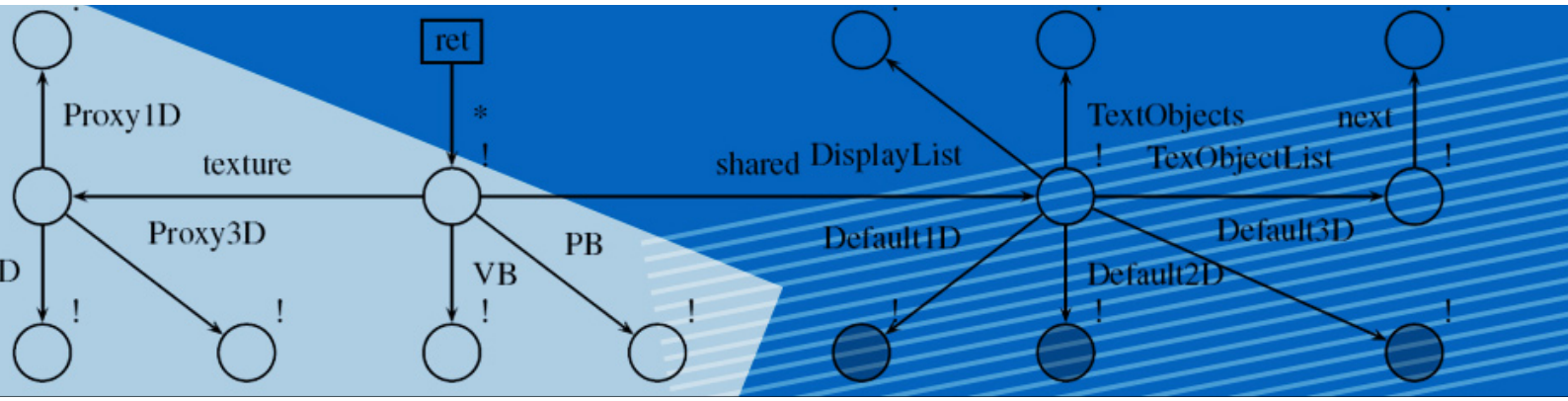
Doosan Cho, Pasricha, S., Issenin, I., Dutt, N.D., Minwook Ahn, and Yunheung Paek.
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 28 (4). April 2009. pp. 554-567.

본 논문은 프로파일링을 이용한 메모리 접근 패턴 최적화와 관련된 것으로 주어진 메모리 아키텍처에 맞게 코드를 최적화하거나 설계단계에서 주어진 어플리케이션에 맞게 하드웨어를 최적화하는데 활용될 수 있는 기법을 제안하고 있다.
제안하는 기법은 미국UCI대 Nikil Dutt교수진이 벨기에의 유명 연구소인 IMEC과 공동연구를 진행했던 결과에 기반하고 있다. 본래 UCI의 연구는 정교한 소스레벨 분석을 통한 코드 최적화에 있으며 이러한 기술은 소스단계에서 메모리 접근 오류 분석에 활용하는 것이 가능하다. 최적화 관련된 분석기술을 오류검출도구로 응용하는 것은 매우 흥미로운 작업이 될 것으로 기대하고 있다.

Abstract Parsing Static Analysis of Dynamically Generated String Output Using LR-Parsing Technology

Kyung-Goo Doh, Hyunha Kim, and David A. Schmidt.
Proceedings of the 16th International Symposium on Static Analysis (SAS'09). August 2009. pp. 256-272.

학자에게는 좋은 연구거리를 찾는게 가장 큰 고민거리 중 하나다. 그런데 우연히 운 좋게도 평소에 잘 알고 지내던 소프트웨어관리도구 개발업체 사장과 차 한잔하면서 잡담하다가 연구거리가 하나 굴러들어왔다. 프로그램 실행 중에 만들어지는 문자열을 실행해보지 않고 이해해보고 싶다고 했다. 내가 즉흥적으로 한 대답은 "허! 그거 간단해요. 문자열 데이터 흐름 분석하면 바로 답이 나와요."였다.
그런데 전통적인 방식으로 분석해보니 기껏해야 정규표현식 정도의 정밀도로 문자열을 요약할 수밖에 없었다. 나름대로 회사에서 원하는 정도의 정보를 얻기에는 충분했으나, 그래가지고는 실행중에 만들어내는 문자열의 구문구조는 이해할 수 없었다. 누가 해놓은 연구가 없는지 뒤져보았으나 다른 사람이 해놓은 것도 같은 한계에 직면하고 있었다. 별로 돌파구가 보이지 않는 듯 했다.
그러던 중 우연히 Dave Schmidt 교수와 만나서 이 문제를 이야기했고, 얼마 후 파싱이론을 적용해보면 어떨까하는 제안을 해왔다. 쌓아놓은 도가 웬만큼 깊지 않고는 쉽게 생각해낼 수 없는 발상이었다. 그 순간. 아! 바로 이거다 싶었다. 오래 묵은 LR(k) 파싱알고리즘을 적용했더니 문제가 술술 풀렸다. 파싱이론을 정립해놓은 선구자들이 위대해 보였다. 실행중에 만들어내는 문자열의 구문구조를 이해할 수 있으니 이제 의미구조의 이해에 도전할 차례다.
논문심사평 중에서 좋은 말 몇 마디 붙이면...
하나, This paper is an innovative step forward in string analysis with many applications. The use of parsing stack as an abstract domain has other static analysis applications in estimating values that can be characterized by LR(k) grammars.
둘, First notion of "analyze-and-parse" style string analysis and use of the set of parsing stacks as the abstract domain.



Improving Bug Triage with Bug Tossing Graphs

Gaeul Jeong, Sunghun Kim, and Thomas Zimmermann.
Proceedings of the European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'09). Amsterdam, The Netherlands. 2009. pp. 111–120.

이광근 교수님의 지도 및 조언, 김성훈 박사님의 아이디어 및 추진력, Thomas Zimmermann의 경험 그리고 나의 실험이 잘 조화를 이루었던 연구다.

당시 "This is a quite good paper, based on a simple but clever idea, and with practical applicability." 라는 논문 리뷰가 있었다. 리뷰가 이야기하는 것처럼 우리의 아이디어는 작고 단순했지만, 신선하고 분명한 방안을 제시함으로써 좋은 결과를 얻어낼 수 있었다. 나는 이 논문을 볼 때마다 버스에서 문득 떠오르는 생각들, 깊은 밤 잠들기 전에 번뜩이는 아이디어들을 소중하게 여겨야겠다고 새삼 다짐하게 된다.

이 논문은 내가 대학원에 입학해 처음 참여했던 논문이자 처음으로 채택되었던 논문이다. 또한 이 논문을 통해 처음으로 국제학회에서 발표할 기회를 가질 수 있었다. 이제 시작이라는 기분이 든다. 앞으로 더 많은 모험이 나를 기다리고 있다는 사실이 나를 설레게 한다..

GeneShelf A Web-based Visual Interface for Large Gene Expression Time-Series Data Repositories

Bohyoung Kim, Bongshin Lee, Susan Knoblach, Eric Hoffman, and Jinwook Seo.
IEEE Transactions on Visualization and Computer Graphics. 15 (6). 2009. pp. 905–912.

정보의 홍수로부터 생의학 연구자들을 구하자

최근 10여 년 동안 생의학 연구는 첨단 유전자 분석 기술의 발달로 눈부신 발전을 이루었다. 하지만 값비싼 첨단 장비를 이용하여 만들어낸 방대한 데이터들은 아직 효과적으로 공유되고 활용되지 못하고 있는 실정이다. 본 논문에서는 전 세계 생의학 연구자들이 유전자 칩 실험 결과를 공유하는 웹상의 데이터베이스에서, 유전자 발현에 관련된 방대한 시계열 자료를 효과적으로 열람/검색할 수 있는 시각적 인터페이스를 제시하였다. 특히 timeline과 bar chart를 자연스럽게 통합한 새로운 시각화 도구인 nTimeLines를 제안하여 다음과 같은 아주 긍정적인 리뷰 코멘트를 받았다.

"The described software is innovative and I have no doubt that it will be appreciated by its targeted audience of biologists working with database repositories of microarray data. I agree that a lightweight visualization tool is much needed."

"The software provides a nice-looking and easy to use interface for visualizing the behaviour of top changing genes and pathways across treatments and time points."

Abstract Parsing for Two-staged Languages with Concatenation

Soonho Kong, Wontae Choi, and Kwangkeun Yi.
Proceedings of the 8th International Conference on Generative Programming and Component Engineering (GPCE'09).
Denver, Colorado, USA. 2009. pp. 109–116.

다단계 언어는 서울대학교 프로그래밍 연구실에서 흥미를 가지고 꾸준히 연구하는 주제다.
2008년 11월에 있었던 첫 번째 ROSAEC 워크숍에서 도경구 교수님의 Abstract Parsing 발표를 듣고 우리는 이것을 이용해 다단계 언어의 분석에 도전해보기로 하였다.

본격적으로 연구를 시작하기에 앞서 사용할 무기를 이론적으로 정리하고 다듬기 시작했다. 3개월 정도에 걸쳐 Abstract Parsing을 요약 해석의 틀 안에서 이해하고 정리하여보니 이것만으로도 작고 단단한 결과물이 되었다. 이것을 GPCE'09 학회에 제출하였고 논문이 채택되었다.

아쉽게도 다단계 언어를 공략하는데 지금은 Abstract Parsing이 아닌 다른 방법을 사용하는 연구를 진행하고 있다. 하지만, 초보연구자인 우리(공순호, 최원태)는 목표를 향해 갈 때 신중하게 내디딘 발자국은 그것만으로 가치가 있다는 배움을 얻을 수 있었다.

키워드 : 요약 해석(Abstract Interpretation), 다단계 언어(Multi-staged Language), 프로그램 분석(Program Analysis) 구문 분석(Parsing), 문법(Grammar)

Large Spurious Cycle in Global Static Analyses and Its Algorithmic Mitigation

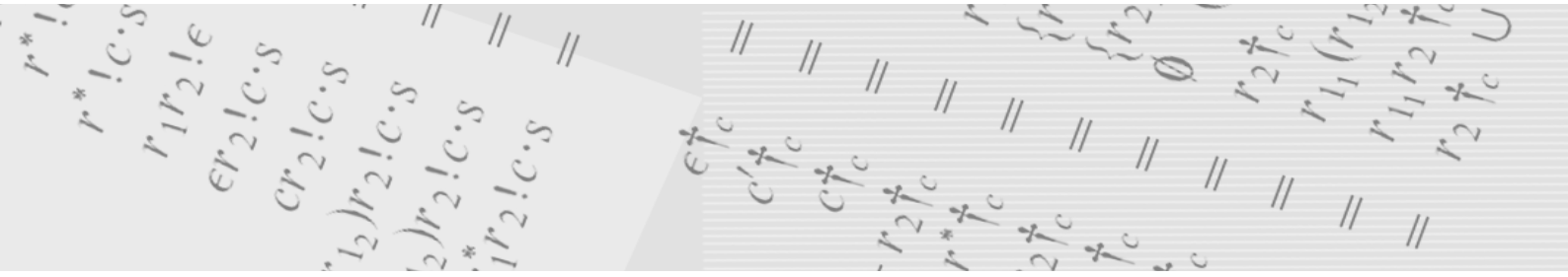
Hakjoo Oh.
Proceedings of the Asian Symposium on Programming Languages and Systems (APLAS'09). Seoul, Korea. December 2009.
pp. 14–29. Software: Practice and Experience. accepted.

이 짧고 간단한 논문을 완성하기 위해서 거의 2년이 걸렸다.

2007년 가을, 우리 연구실에서 개발해오던 프로그램 분석기를 가지고 이리저리 뜯어보던 도중에, 분석을 잘 아는 사람들에게는 당연했지만 그 당시 아무것도 모르던 나에게는 도무지 상식적이지 않던 현상을 하나 발견하게 되었다. 내 나름대로의 방식으로 다시 구현하고 실험을 해보니 분석기의 성능이 몇 배 좋아지는 결과를 얻었다.

그 당시에는 이런 아이디어로 논문을 쓸 수 있는건지에 대한 감도 없던 시기였기에 별 생각없이 랩 세미나 시간에 발표를 했었는데, 교수님께서 논문으로 정리해보자고 하셨다. 아이디어 자체를 내 방식으로 정리하는것은 쉬웠지만, 이 아이디어가 기존의 관련연구들 속에서 어떤 위치에 있는지, 어떤 의미를 가지는지를 명확히 하며, 이 분야의 전문가들이 이해할수 있는 형식과 내용으로 표현하기에는 내공이 많이 모자랐다.

이 논문을 통해 교수님께 정말 많은 것을 배웠다. 커뮤니티의 사람들이 이해할수 있도록 내용을 구성하는 방법을 점차 알게되었고 영어 표현을 비롯한 논문의 마무리에도 감이 조금 잡히게 되었다.



A Practical Memory Leak Detector Based on Parameterized Procedural Summaries

Yungbum Jung and Kwangkeun Yi.
Proceedings of the 7th International Symposium on Memory Management (ISMM'08). ACM. 2008. pp. 131–140.

우리 연구실에서 시간과 공을 들여 개발한 Sparrow의 한 엔진인 memory leak detector에 관한 논문이다. PLDI에 제출했었다가 성능은 좋은데 어떤 점이 새로운 것인지 잘 부각시키지 못해서 떨어졌었다. 이때 ISMM에 동시에 제출이 가능해서 새롭게 써서 제출하기로 했다. 처음으로 혼자 쓰는 논문은 교수님을 참으로 여러번 실망시켜 드렸었다. 이 논문은 교수님께 어떻게 논문을 써야 하는지 배우는 좋은 배움터였다. 논문 제출 마감 때는 교수님께서 안식년으로 미국에 계시고, 구정과 겹쳐서 처가가 있는 경주에서 얼마 없는 PC 방을 찾아 새벽에 돌아다니던 기억이 난다. CMU에 있는 박사과정 학생 Will Klieber의 자세한 교정을 생각하니 또 한번 고맙다.

PLDI에 떨어지고 나니 프로그램 버그를 찾는 도구들이 모두 모여서 겨루는 competition이 있으면 재밌겠다는 생각을 했었다. 프로그램에 memory leak이나 buffer overrun 같은 버그들을 심어놓고 프로그램을 공개하면 각자 다운받아 제한된 시간(일주일?) 안에 수단과 방법을 가리지 않고 누가 가장 많은 버그를 찾는 지를 겨루는 것이다. 성능이 뛰어난 분석기를 만드는 것이 논문에 새로운 아이디어를 제출하는 것보다 실제 현장에 있는 사람들에게는 더욱 절실하다고 본다.

A Polymorphic Modal Type System for Lisp-Like Multi-Staged Languages

Ik-Soon Kim, Kwangkeun Yi, and Cristiano Calcagno.
Proceedings of The ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'06). 2006. pp. 257–269.

2년간의 긴 터널이었다. 2003년 여름부터 2005년 여름까지. 김익순 박사와 함께한 연구 결과였다.

김익순 박사가 어느 날 내게, "스스로 진화하는 프로그램에 대해서 연구해 보고 싶다"고 했다. 김박사는 LISP의 매크로 시스템에 매혹되 있었다. 내 답은, "그게 요즘 이야기되는 다단계 프로그래밍(multi-staged programming)이란 것이다. 우선 그러한 프로그래밍 언어의 좋은 static type system을 만들어 보자." LISP정도의 풀 다단계 언어를 위한 정교한 타입 시스템이 목표였다.

김박사는 곧 타입 룰을 가지고 왔다. 맞을 수 밖에 없다는 직감이 왔다. 그러곤 같은 연구를 앞서 한 대표 주자들을 접촉했다. Imperial의 Cristiano Calcagno을 2004년 1월 POPL에서 만나 이야기했다. 현재 다단계 언어의 타입시스템에서 제일 문제되는 것을 우리가 푼 것일 지 모른다는 확신을 얻었다. CMU의 Frank Pfenning을 2004년 APLAS에 초청강연 연사로 불러서, 김박사 일을 소개하고 김박사와 이야기 붙였다. Rice의 Walid Taha와는 수시로 email하면서 우리 아이디어를 흘려보았다. 이러면서 좋은 결과를 우리가 만들어 냈다는 자신감이 쌓였다.

직관적인 타입 룰. 단순 타입 시스템의 안전성 증명은 수월했다. 다형타입시스템(polymorphic type system)으로 정교하게 확장했다. 그 안전성 증명은 지난했다. 이번엔 메모리 반응(imperative features)을 지원하도록 확장했다. 그 안전성 증명도 또 지난했다. "이렇게 계속 증명만 하고 시간이 가도 되는 건가요." "확인하지 않고는 굴 밖으로 나갈 수 없지 않겠냐." 거의 매주 토요일 오전에 만나서 증명을 확인해보고 수정했다. 토요일 점심을 참 많이 같이했다. 캠퍼스 솔밭식당으로 자주 갔다.

김박사는 이렇게 논문없이 세월이 가도 되는지 초조해했다. 나는 소총 100발 보다는 미사일 1방이 더 가치있다고 했다.

POPL 2006을 위해 2005년 7월 논문을 제출했다. 2달여 후 억셉트 email이 왔다. 기뻤다. 박사논문을 POPL에 내고 두 번째였다. 특히, 연구동기부터 마무리까지 순수 국내 연구로 POPL에 낸 것이 자랑스러웠다. 이 소식을 받을 때 김익순 박사는 파리 Cousot그룹에서 1년간 방문연구를 막 시작한 중이었다.

논문 리뷰중:
"The article is a significant step towards a practical multi-stage extension of ML."
"I found this to be original and significant work, and paper is well written. Much effort has gone into presenting the exact relation of this work with related work, which helps view their contribution in a much clearer perspective."
"Overall, this looks like a clean design. It seems to have all of the components that one would want for multi-stage programming. I did not check the proofs carefully, but they seem plausible. I think that this paper should be published."

Automatic Reproduction of a Genius Algorithm Strassen's Algorithm Revisited by Genetic Search

Seunghyun Oh and Byung-Ro Moon.
IEEE Transactions on Evolutionary Computation. accepted.

알고리즘으로 천재에 맞서 보자

스트라센! 듣기만 해도 주눅이 드는 이름이다. 행렬의 곱셈 연산 시간은 무조건 $\theta(n^3)$ 이라는 고정관념을 깨버려 20세기의 명사중 한 사람이 된 천재수학자 스트라센. 사람들이 그의 알고리즘에 놀라는 것은 어떻게 그 방대한 문제 공간에서 그런 해를 찾아내었느냐 하는 것이다. 나에게 그를 옆자리에 앉혀놓고 경쟁을 하라고 한다면 찻집이나 하나 차려 여생을 편히 사는 길을 택하겠다.

10여년 전부터 유전 알고리즘을 이용하여 스트라센의 알고리즘을 재현하는 것을 목표로 작업을 시작했다. 학회에서 만난 외국인 동료들에게 그 작업을 하고 있다고 하니까... "... looks interesting..." 재미있어 보이지만 되겠냐? 이런 표정들이다. 그래 쉽지는 않겠지. 그렇지만 우리 전공이란 게 밑져봐야 재료비가 드는 것도 아니고... 지가 아무리 천재라도 사람인데... 우리에겐 컴퓨터와 공간탐색 기법이 있지 않은가? 결과는... 실패, 또 실패... 훌쩍 5~6년.

이러던 어느날, 오승현 학생으로부터 흥분이 담긴 이메일이 날아왔다. "교수님, 찾았어요." 스트라센과 같은 성능의 알고리즘이 드디어 발견되다! 좀 더 나가니.. 스트라센의 알고리즘도 발견. 현재까지 대칭과 중복을 다 제외하고 적어도 608개의 서로 다른 스트라센급 $\theta(n^{2.81})$ 알고리즘을 발견하였다. 공간탐색 알고리즘의 위력에 새삼 놀란다.

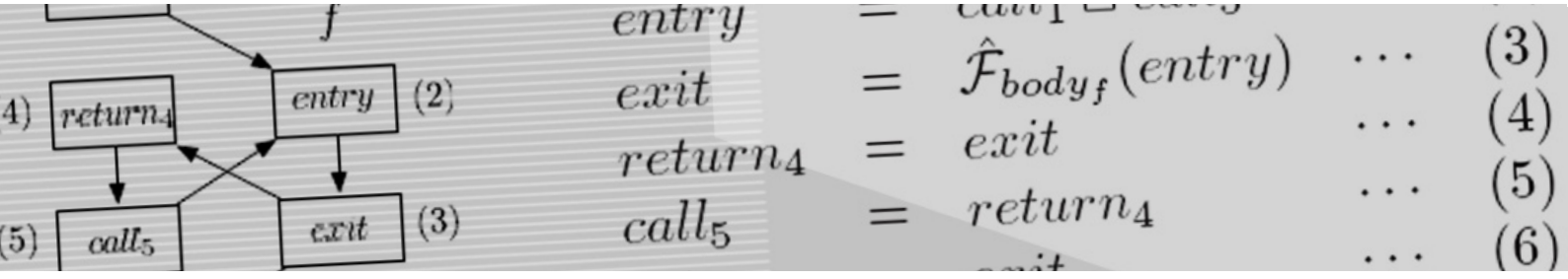
요즘은 스트라센의 $\frac{n}{2} \times \frac{n}{2}$ 행렬곱을 벗어나 $\frac{n}{3} \times \frac{n}{3}$ 행렬을 이용해서 더 좋은 알고리즘이 있는지 찾는 작업을 하고 있다. 10여년 전에 처음 시작하던 때보다 더 난감하고, 벽이 높다. 그렇지만 시간의 문제지 나 죽기 전에는 될 것이다. 지금까지 해온 것들이 정도의 차이가 있을 뿐 주로 그랬으니까... 우리는 그만한 천재가 아니지만 천재를 누를 수 있는 문화적 도구를 갖고 있다. 이런 것이 리처드 도킨스가 말한 '확장된 유전형'이겠지.^^

Scalable Shape Analysis for Systems Code

Hongseok Yang, Oukseh Lee, Josh Berdine, Cristiano Calcagno, Byron Cook, Dino Distefano, and Peter O'Hearn.
Proceedings of the International Conference on Computer Aided Verification (CAV'08). Princeton, NJ, USA. July 2008. pp. 385-398.

방학동안 양홍석 박사가 있는 영국으로 가서 공동 연구한 것이다. 양홍석 박사의 고민이 모양 분석기(shape analysis)의 성능이였다. 잘 알려져 있는 기술들을 활용하여 매일 속도를 반으로 줄였고 결과적으로 몇 백배 빨라진 경우까지 생겼다. 결과는 좋았지만 새로운 기술이 없어 논문 쓰기가 어려웠다. PLDI에 한 번 떨어지고 나서, 마이크로소프트에 있는 Byron Cook을 영입했다. Byron Cook의 현란한 글솜씨와 필요한 것/필요없는 것 골라내는 기술, 감탄스러웠다. 바로 CAV에 게재 승인을 받았다. 글 쓰는 것, 얼마나 중요한 기술인지!

키워드: 모양 분석 (shape analysis), 프로그램 분석 (program analysis), 성능 (performance)



Convex Optimization Algorithms for Active Balancing of Humanoid Robots

Juyong Park, Jaeyoung Haan, and Frank C. Park.
IEEE Transactions on Robotics. 23 (4). 2007. pp. 817.

로봇에게 균형감각을 심어주자

현대인들의 필수품이 된 컴퓨터처럼, 미래에는 인간을 닮은 휴머노이드 로봇이 우리의 삶에 아주 가까이 있을 것이다. 집안에서 집안일을 돕기도 하고, 같이 외출하여 무거운 짐을 들어주기도 하고 말이다. 그런데 이런 휴머노이드 로봇이 균형 감각이 없어 잘 넘어진다면 어떨까? 만약, 버스를 타고 가다가 급출발한 버스 때문에 이 비싼 로봇이 넘어져서 고장나거나 인명 피해라도 낸다면 그 누가 이 로봇과 함께 하겠는가?

이러한 휴머노이드 로봇에게 '균형감각을 심어주자.'라는 목표로 최적화 기반의 로봇 동작 안정화 알고리즘을 연구하기 시작했다. 주변 환경의 변화에도 스스로 넘어지지 않고 균형을 잘 잡아 넘어지지 않을 수 있는 그런 균형감각을 부여해주는 것이다. 일반적으로 이런 휴머노이드 로봇의 안정성에 관련한 제약조건들은 복잡한 비선형 형식으로 표현된다. 문제는 이러한 복잡한 비선형 제약조건들은 가진 최적화 문제는 실시간으로 풀어나가기 힘들뿐더러 전역 최적해를 구해준다는 보장이 없다. 다시 말해, 아무리 좋은 균형 잡기 알고리즘이라 하더라도 아주 고사양의 컴퓨터가 로봇에 탑재되어 있지 않으면 무용지물이라는 것이다.

그러던 어느 날, 이 복잡한 최적화 문제를 볼록 최적화(convex optimization) 문제로 정의 할 수 있다는 사실을 박주용 학생이 알아냈다. 어떤 최적화 문제를 볼록 최적화 문제로 정의 할 수만 있다면, 전역 최적해를 구할 수 있음이 보장될 뿐 아니라, 수치적 계산 효율도 상당히 좋아지게 된다. 이렇게 새로 정의한 균형잡기 볼록 최적화 문제를 시뮬레이션 해 본 결과, 25자유도를 갖는 휴머노이드 로봇에 대해 균형 잡기 알고리즘이 실시간으로 동작한다는 사실을 보였다.

논문을 마무리하면서, 이 균형잡기 알고리즘이 다양한 시나리오에서 성공적인 실시간 균형잡기 성능을 내는지 테스트해 보았다. 우리 태권도 동작의 앞 차기 동작과 마치 버스가 급출발할 때처럼 가속되는 바닥위에서의 균형잡기 동작을 테스트하였다. 그 결과 마치 사람이 넘어지려고 할 때 양팔 및 다른 발을 벌려서 균형잡는 것과 유사한 동작을 취하는 것을 볼 수 있었다. '역시 인간은 본능적으로 많은 것을 터득하고 있구나' 라고 생각해본다.

Design Verification in Model-Based Micro-Controller Development Using an Abstract Component

Yunja Choi and Christian Bunse.
Software and Systems Modeling. accepted.

정형과 비정형의 조화

이 논문은 독일 국제대학의 Bunse 교수와의 다년간의 공동작업의 첫 번째 결실이다. Bunse 교수는 소프트웨어 설계 방법론의 전문가이고 내장형시스템의 모델기반 개발방법론에 특별한 관심을 가지고 있는 사람이기도 하다. 공동작업의 목적은 원대했지만, 과정은 험난했다. 비정형성과 고질적인 습관과 생산 비용과 예외에 대한 고려가 주를 이루는 소프트웨어의 세계와, 논리적 무결성과 명확성이 생명이라고 할 수 있는 정형기법이라는 너무나도 판이해 보이는 두 세계의 실질적 교집합을 찾는 것이다. 이 논문에서는 그 첫 단추로 설계자의 입장에서 이해할 수 있는 정형검증기법의 적용방법론과 기법을 소개하였다.

키워드: 추상컴포넌트, 디자인 검증, 모델기반 개발 방법론

Dependency-Aware Reordering for Parallelizing Query Optimization in Multi-Core CPUs

Wook-Shin Han and Jinsoo Lee.
Proceedings of the 35th SIGMOD International Conference on Management of Data (SIGMOD'09). 2009. pp. 45-58.

VLDB 2008 논문의 확장 연구. 이 연구에서는 dependency-aware reordering이라는 새로운 개념을 사용하여 질의 최적화기를 병렬화하는 범용 병렬화 프레임워크를 처음으로 개발하였다. 개발된 프레임워크는 모든 bottom-up기반의 질의 최적화기를 지원하며, 질의 최적화를 위한 워크로드를 동적으로 분배하는 특징을 가지고 있다. 실험 결과, linear speedup을 보였으며, 기존 연구보다도 좋은 성능을 보였다.

리뷰평의 일부를 발췌하면 아래와 같다.

"The authors have looked at an important class of algorithms (bottom up dynamic programming) and have given a full accounting of how they can be parallelized on a modern multi-core computing system. They have looked not only at the theoretical aspects of scheduling these computations, but the actual software methodology and hardware tuning. (Really nice job!)"

키워드: Dependency-aware reordering, Multi-cores, Query optimization, Parallel databases

---

Query Result Clustering for Object-Level Search

Jongwuk Lee, Seung-won Hwang, Zaiqing Nie, and Ji-Rong Wen.
Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM New York, NY, USA. 2009. pp. 1205-1214.

연구라는 이름의 소통

대표논문 [3], [4], [5]는 센터에 참가하게 된 뒤 센터 사사로 쓰게 된 논문들이다. [1], [2]와 다른 점이 있다면, 학문 후속세대인 연구실 학생들과 다른 연구 분야 연구자의 협동으로 이루어졌다는 점이다-- [3]은 마이크로소프트 아시아의 연구원들과 협력하였고 [4]는 포스텍의 계산기하 연구실과 협력하였고 [5]는 HKUST 소프트웨어 공학 연구실과 협력하였다. 후속세대의 초보 운전 연수를 겸하는 아찔하지만 신선한 연구, 다른 분야 연구자와 소통하면서 낯익은 문제 낯설게 보기, 혹은 낯선 문제 낯익게 보기. 이러한 새로운 도전을 통해, 후속세대와, 다른 분야와 연구로 소통하고 한 단계 성장하게 된 성장하게 된 느낌을 경험하였다. 이러한 소통의 열매인 [3]은 KDD의 유일한 한국 기관 소속 발표 논문이고, [4]는 최우수 논문상을 수상하였고, [5]는 생애최초의 소프트웨어 공학 논문이다.

[1] Seung-won Hwang, Kevin Chen-Chuan Chang:Optimizing top-k queries for middleware access:A unified cost-based approach. ACM Trans. Database Syst. 32(1): 5 (2007)
[2] Seung-won Hwang, Kevin Chen-Chuan Chang: Probe Minimization by Schedule Optimization: Supporting Top-K Queries with Expensive Predicates. IEEE Trans. Knowl. Data Eng. 19(5): 646-662 (2007)
[3] Jongwuk Lee, Seung-won Hwang, Zaiqing Nie, Ji-Rong Wen: Query result clustering for object-level search. KDD 2009: 1205-1214
[4] Wanbin Son, Mu-Woong Lee, Hee-Kap Ahn, Seung-won Hwang: Spatial Skyline Queries: An Efficient Geometric Algorithm. SSTD 2009: 247-264
[5] Adding Examples into Java Documents, Jinhan Kim, Sanghoon Lee, Seung-won Hwang, Sunghun Kim. ASE 2009

A Logical Account of Uncertain Databases Based on Linear Logic

Sungwoo Park and Seung-won Hwang.
Proceedings of the 12th International Conference on Database Theory (ICDT'09). 2009. pp. 141-148.

몇 년 전 Heinz Schweppe라는 독일분이 학과를 방문하셨다. 데이터베이스를 연구하시는 분으로 내 사무실과 가까운 곳에 계셔서 이런 저런 얘기를 나누었는데 "DB and PL are not that apart from each other."라는 인상깊은 말씀을 해 주셨다. 그 뒤 uncertain database에 대해서 설명을 해 주셨는데 linear logic에서와 비슷한 기호를 사용하는 것을 발견했다. 나는 linear logic을 이용하면 uncertain database의 의미를 매우 간결하게 설명할 수 있다고 직감했고 그 뒤 이 아이디어를 논문으로 완성했다. 이 논문의 내용은 깊지 않아서 linear logic의 기초적 지식만 있으면 누구나 이해할 수 있다. 그러나 다른 분야의 사람과 대화를 통해서 새로운 연구 주제를 찾아낸 아주 값진 경험을 나에게 선사한 논문이다.

Unit Testing of Flash Memory Device Driver through a SAT-Based Model Checker

Moonzoo Kim, Yunho Kim, and Hotae Kim.
Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering (ASE'08). Sept. 2008. pp. 198-207.

연구를 하면서 항상 실용성에 대한 고민을 한다. 본 논문은 우리 연구가 실용적임을 보인 시금석이 되는 연구였다. 뿐만 아니라, 다른 연구자들도 산업체 사례 연구에 많이 목말라 하고 있었다는 걸 학회에서 느꼈다.

키워드: formal verification, embedded software, flash memory, SAT-based model checking

Functional Netlists

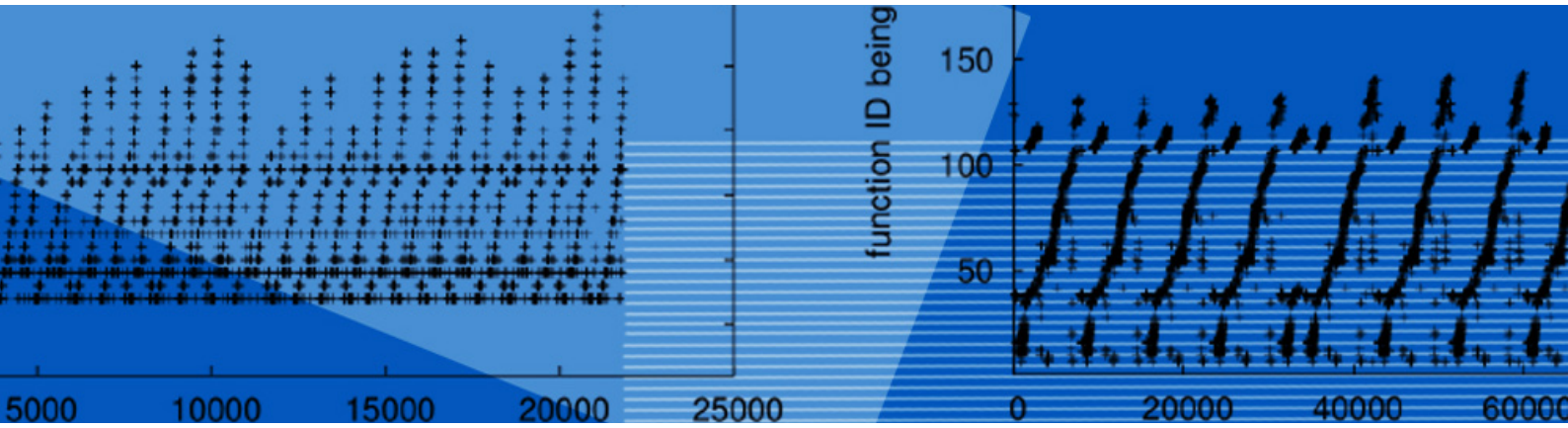
Sungwoo Park, Jinha Kim, and Hyeonseung Im.
Proceedings of the 13th ACM SIGPLAN International Conference on Functional Programming (ICFP'08). 2008. pp. 353-366.

2006년 포스텍에서 처음으로 학부 프로그래밍언어를 강의하는 동안 학생들로부터 과제가 많다는 불평을 줄곧 받았다. 총 8개의 과제가 지나치게 많은 것은 아니라고 생각했기 때문에 자세한 내막을 알아보았다. 학생들의 불만은 전년도까지 프로그래밍언어 과목은 과제가 거의 없는 (널널한) 과목이었는데 갑자기 과제가 많아져서 함께 수강하는 컴퓨터구조 과목의 과제를 할 시간이 없게 되었다는 것이고, 결국 엉뚱하게 애꿎은 나의 과목으로 화살을 돌리고 있었던 것이다.

당시 컴퓨터구조 과목의 과제는 Verilog 프로그래밍이었는데 나는 Verilog로 변환되는 함수형 하드웨어 기술 언어를 학생들에게 배포하면 컴퓨터구조 과목 과제를 빨리 끝낼 수 있을 것이고 그러면 나의 과목 과제를 더 열심히 하게 될 거라는 장난기 섞인 생각을 했다.

관련 연구를 살펴본 나는 이 주제가 사실 좋은 연구 주제임을 발견했고 이 후 본격적으로 문제를 풀기 시작했다.
결국 lambda-calculus를 Verilog로 변환하는 시스템을 설계해서 구현했고 2008년 ICFP에 발표하였다.

순수하게 학부생의 과제를 돕겠다는 연구 동기 때문에 논문의 깊이와는 무관하게 내가 가장 자랑스러워하는 논문이다.



A Probabilistic Language Based on Sampling Functions

Sungwoo Park, Frank Pfenning, and Sebastian Thrun.
ACM Transactions on Programming Languages and Systems. 31 (1). 2008. pp. 1–46.

2006년 초 POPL 2005 프로그램 위원장으로부터 TOPLAS POPL Special Issue에 논문을 제출해 달라는 부탁을 받았다. 초청 논문이어서 POPL 논문을 약간 확장만 하면 게재 승인될 줄 알았는데 4명의 reviewer들로부터 호된 비판을 받았다. 그 이후 3번의 revision을 거쳐서 최종적으로 46쪽의 긴 논문을 완성하게 되었다. Revision summary로 제출한 문서가 총 64쪽으로 논문보다 길었으며, 논문 수정 중에 좋은 박사 학위 연구 주제를 발견하기도 했다. 그런데 모든 reviewer들로부터 OK를 받았을 때는 이미 POPL Special Issue가 출판된 상태였다. 다시 제출 후 3개월 뒤 게재 승인이 났다.

Optimizing Top-k Queries for Middleware Access A Unified Cost-Based Approach

Seung-won Hwang and Kevin C. Chang.
ACM Transactions on Database Systems. 32 (1). 2007. pp. 5.

셋방에서 1가구2주택까지

대표논문 [1]과 [2]는 박사과정 동안 관심을 가졌던 랭킹을 빨리 계산하는 알고리즘을 인덱스 유무 등의 임의의 다양한 환경을 다루도록 확장하여 정리한 저널 원고이다. 나의 연구 결과는 SIGMOD 2002년에 처음 출판되었는데 랭킹 문제가 정보 검색에서 흔히 사용되는 기법이다 보니 SIGMOD에 보내면 SIGIR로 보내라는 리뷰가 오고 SIGIR로 보내면 SIGMOD로 보내라는 리뷰가 오는 설움을 겪었다. 2002년 이런 연구는 데이터베이스 학회에서 "포푸리"나 "질의 처리" 세션에 세 들어 발표되었다. 2009년 현재에는 연구자가 많이 늘어 데이터베이스와 정보검색 학회에서 모두 Ranking I, Ranking II 등의 멀티세션으로 편성 될 만큼 규모가 커졌는데, 나의 연구 커리어와 시작을 같이 하는 이 토픽의 미래 모습에 궁금증과 애착이 크다.

[1] Seung-won Hwang, Kevin Chen-Chuan Chang: Optimizing top-k queries for middleware access: A unified cost-based approach. ACM Trans. Database Syst. 32(1): 5 (2007)
[2] Seung-won Hwang, Kevin Chen-Chuan Chang: Probe Minimization by Schedule Optimization: Supporting Top-K Queries with Expensive Predicates. IEEE Trans. Knowl. Data Eng. 19(5): 646-662 (2007)
[3] Jongwuk Lee, Seung-won Hwang, Zaiqing Nie, Ji-Rong Wen: Query result clustering for object-level search. KDD 2009: 1205-1214
[4] Wanbin Son, Mu-Woong Lee, Hee-Kap Ahn, Seung-won Hwang: Spatial Skyline Queries: An Efficient Geometric Algorithm. SSTD 2009: 247-264
[5] Adding Examples into Java Documents, Jinhan Kim, Sanghoon Lee, Seung-won Hwang, Sunghun Kim. ASE 2009

Kripke Models for Classical Logic

Danko Ilik, Gyesik Lee, and Hugo Herbelin.
Annals of Pure and Applied Logic. accepted.

공저자: Danko Ilik, Hugo Herbelin
키워드: Kripke model, classical logic

미국 태생의 철학자이자 논리학자인 Saul Kripke가 1950년대 후반에서 1960년대 초반 사이에 고안한 Kripke model은 non-classical 논리 시스템에 대한 semantics를 제공한다. 예를 들어 intuitionistic first-order logic과 modal logic에 대해서 sound하며 complete한 semantics를 제공하며 이는 classical first-order logic이 참, 거짓에 기저를 둔 집합론식 semantics를 갖는 것과 대비된다.
Kripke model은 참, 거짓이라는 이분법이 아닌 주어진 명제의 증명가능성을 기본 개념으로 한다. 즉, 어떤 주어진 명제의 참, 거짓 여부가 아니라 그것이 참인지 거짓인지에 대한 증명가능성을 이야기 한다. 예를 들어, 주어진 명제의 참, 거짓 여부가 증명 되었는가 또는 앞으로 증명이 될 것인가에 관심을 둔다.

이와 같이 Kripke model은 intuitionistic logic 또는 modal logic과 깊은 연관을 맺고 있으며 classical logic과는 무관하다고 알려져 왔다. 다만 double-negation을 통해 간접적으로만 정의해서 언급되어졌을 뿐 실질적인 활용성에 있어서는 별다른 연구가 진행되지 않아 왔다. (참고로 double-negation은, 간략하게 설명해서, A라는 명제가 classical logic에서 증명가능하면 ¬¬A는 intuitionistic logic에서 증명가능하다 라는 관계를 보이는 데에 이용된다. 따라서 A가 classical logic에서 forcible 하다는 것을 ¬¬A가 intuitionistic logic에서 forcible 하다라는 것으로 정의하면 classical logic에 대한 sound하며 complete한 Kripke semantics를 얻는다.)

반면에 위 논문은 classical logic에 대한 Kripke semantics를 어떠한 double-negation도 사용하지 않으면서 forcing relation을 정의하는 방식을 소개한다. 이와 같이 classical Kripke semantics를 직접적으로 정의할 경우에만 얻어질 수 있는 중요한 부산물이 있다. 대표적으로 기존의 어떤 증명보다도 쉬운 cut-elimination의 constructive한, 즉 배중율을 사용하지 않는 증명과 그 결과로 얻을 수 있는 classical logic의 무모순성(consistency)의 constructive한 증명이다. 배중율을 사용하지 않는다는 사실은 어떠한 증명으로부터도 cut-rule를 사용하지 않는 증명을 기계적으로 추출할 수 있음을 의미하며 이는 proofs-as-programs을 사용할 경우 주어진 proof term의 normal form을 자동으로 추출하는 프로그램을 작성할 수 있음을 의미한다.

결과적으로 위 논문은 classical logic에 대한 Kripke semantics를 직접적으로 정의하는 방식을 처음으로 제안한 의미 있는 논문이 되었으며 앞으로 새로운 연구방향을 제시하는 역할을 수행할 것이라 기대하고 있다.

4

산학연협력이야기


Invariant Propertie
For the annotated loop
An Invariant I must satisfy the following condi
Observation #1
Observation #2

Mission Impossible!

도경구
교수
한양대학교 컴퓨터공학과

2009년 행정안전부 정보보호정책과가 주관한 전자정부지원사업의 일환으로 “정보시스템 보안강화체계 구축” 사업을 한국인터넷진흥원(KISA)이 수행하였다. 이 사업 중에서 핵심부분인 소스코드 보안취약점 자동진단도구 개발과 진단지원체계 구축 사업에 지티원(주사업자), 파수닷컴, 한국정보보호학회 소프트웨어보안연구회가 컨소시엄으로 참여하여 사업을 성공적으로 수행 완료했다. 개발한 자동진단도구는 최근 개발된 2건의 전자정부 소프트웨어의 보안취약점을 진단하는데 시범적으로 사용되었으며, 앞으로 전자정부에서 발주되는 모든 소프트웨어에 확대 적용될 예정이다. 이 글은 이 도구개발에 관련된 뒷이야기이다.

학교의 연구개발과 산업체의 연구개발에는 괴리가 있다고들 한다. 정부는 산학협력을 활성화하기 위해 엄청난 자금 지원을 해대지만, 쏟아 부은 지원금 대비 실적은 기대한 만큼 좋지는 않아 보인다. 그런데 우연히 제대로 된 산학협력 경험을 한번 해봤다. 정부지원에 이끌려서가 아니라 완전히 자연발생적으로 말이다.

이야기는 2003년 여름으로 돌아간다. 평소에 잘 알고 지내던 소프트웨어관리도구 개발업체 대표와 차 한잔하면서 잡담하다 일이 시작되었다. 이 회사는 엔터프라이즈 소프트웨어를 관리하는 도구를 만든다. 애플리케이션 소스코드에서 데이터베이스 질의문을 추출할 수 있으면, 데이터베이스 스키마 변경에 영향을 받는 애플리케이션을 알아낼 수 있단다. 그러면 수시로 발생하는 데이터베이스의 변경에 따른 소스코드의 변경 관리를 체계적으로 할 수 있을 터이다. 그런데 질의문은 보통 사용자 입력을 포함하여 문자열을 조합하여 만들어진다. 이렇게 만들어질 질의문을 소스코드를 실행해보지 않고 알고 싶다고 했다. 그런데 어떻게 해야 할지 막막하다는 것이다. 내가 즉흥적으로 한 대답은 “허! 그거 문자열 데이터 흐름 분석하면 되는데.”이었다. 아마 프로그램 분석을 알고 있는 사람이면 누구라도 똑같은 대답을 했을 것이다. 바로 연구개발 과제를 하나 받아와서 연구실 학생들에게 풀어놓았다. 그러고 바로 1년 후 문자열분석엔진 버전 1.0이 탄생했다. (이 연구가 시발점이 되어 요약파싱 기술이 탄생하게 되었으며, 지금도 계속 진화중이다.) 이 엔진은 회사의 주력제품에 핵심 부품으로 성공적으로 탑재되어 잘 팔리고 있다. 회사 기술진의 부족한 부분을 메워 주었으니 졸지에 회사로부터 점수를 좀 딴 셈이다.

그러다가 2006년 BK21 2단계 사업을 시작하게 되었다. 정부지원 사업비의 30% 만큼의 산학협력연구사업을 해야 한단다. 모두들 안면 있는 회사들 찾아다니느라 난리들이다. 어쩔 수 없이 이 회사를 찾아갔다. 사정을 이야기했더니 이미 성공적인 협력사례가 있어서인지 호의적이었다. 문제는 뭘 하느냐 이었다. 얼마 후 이 회사 대표 및 기술연구소 담당자들과 미팅을 가졌다. 이 회사 대표는 국내에서도 이미 시판 중이었던 미국산 소프트웨어 보안취약점 검사 도구를 언급하면서, 그 도구와 필적할 만한 소스코드 취약점 분석엔진을 개발해보면 어떨까라고 제안해왔다. 그렇지 않아도 소프트웨어보안에 관심을 가져오던 차였다. 소프트웨어는 설계자가 의도했던 순기능은 차질 없이 잘 수행되고, 의도하지 않았던 역기능은 최대한 억제되도록 구현되어야 한다. 역기능을 완전히 차단하지 못하면 보안취약점이 되고, 해킹 공격은 대부분 이러한 취약점을 악용하여 이루어진다. 보안취약점을 야기하는 역기능이 있는지 개발 시 분석해서 알아낼 수 있으면, 이를 사전에 제거하여 소프트웨어의 보안을 훨씬 강화할 수 있을 터이다. 그래서 시판중이던 도구의 기능을 조사해보고 이 제품의 소스코드 보안취약점 탐지 엔진과 같은 기능을 가진 정적분석엔진을 만들어보기로 결정했다. 바로 동료교수인 이욱세 교수와 같이 연구팀을 만들어 연구개발에 착수했다. 기존 제품을 분석해보았더니 난이도가 있는 깊은 분석은 하지 않는 경량급 분석기로 추측되었다. 따라서 보안취약성을 야기하는 소스코드의 구문패턴과 흐름패턴을 규칙으로 기술할 수 있는 규칙명세언어를 만들어, 취약점 패턴 데이터베이스를 구축한 뒤, 일괄적으로 해당 패턴의 존재유무를 점검하는 규칙기반 보안취약점 정적분석엔진을 개발하였다. 6개월 정도 후 프로토타입이 나왔고, 기존 상용제품과 성능비교평가를 해보았더니 대동소이했다. 회사는 결과에 만족하고, 이 엔진을 코딩 표준 준수 여부를 점검해주는 CodePrism이라는 제품에 탑재하여 상용화하였다. 우리는 좀 더 정밀한 분석의 필요성을 느끼고, 이후 정밀한 분석결과를 얻을 수 있는 깊은 분석을 위한 이론 연구에 주력하였다. 결국 과제 종료 이후에는 서로 다른 길을 가게 된 셈이다. 그게 끝인가 싶었다.

그러던 중 2009년 앞에서 언급했던 행안부 개발과제가 떴다. 이 과제를 반드시 수행하여 전자정부 소프트웨어의 보안 품질을 높여야 한다는 강력한 의지를 가진 몇 분이 강력하게 과제를 밀어붙였다고 한다. 보안개발프로세스의 전도사로 인정받고 있는 고려대 최진영 교수가 이 과제에 상당한 관심을 보이면서 보안취약점 탐지도구는 프로그래밍언어 연구커뮤니티에서 책임지고 해야 하지 않겠냐고 적극적으로 권유해왔다. 바로 3년 전 보안취약점 분석엔진을 만들어 본 경험이 있었고, 정밀한 C 오류분석에 특화된 이광근 교수 연구팀의 Sparrow도 있었던 터라, 우리가 맡아서 할 수 있다고 자신 있게 반응했다. 바로 해당 회사와 교수들이 모여 컨소시엄을 산업체 중심으로 구성하여 도전하였다. 하지만 국내의 기술 능력에 대한 비관적인 시각이 주류를 이루는 듯 보였다. 이미 상품화되어 현장 검증이 된 외산제품이 더 신뢰를 받았는지, 결국 외산제품을 등에 업은 컨소시엄이 우선협상대상자로 선정되었다. 실망이었다. 그러나 몇 달 후 낭보가 들어왔다. 외산제품을 제공하는 본사가 사업의 주된 요구사항인 소스코드 제출을 거부하여 우선협상대상에서 제외되었다는 것이다. 결국 천신만고 끝에 우리 컨소시엄이 과제에 참여하는 기회를 얻게 되었다. 그런데 문제는 과제 수행대상자를 선정하는 과정에서 시간이 너무 흘러버렸다는 것이다. 과제를 수행하는데 3개월 밖에 주어지지 않았다. 그야말로 Mission Impossible 이었다. 그러나 다행히 우리는 몇 년 전 만들어 놓은 보안취약점 분석엔진 기술을 확보하고 있었고, 완제품 Sparrow도 있었다. 전투 준비 태세를 완전히 갖춘 전사였던 것이다. 취약점 분석 및 데이터베이스 구축은 교수들이 담당하고, 취약점 검수도구 및 진단지원체계 구축은 산학협동으로 수행하였다. 밤낮을 가지지 않고 100% 총력을 기울여 시간 내에 완성할 수 있었다. 미리 준비해두지 않았다면 절대적으로 불가능한 작업이었다. 주위의 대부분의 관계자들이 반신반의하며 걱정스런 시각으로 바라봤지만, 이제 그 시각이 희망적이며 긍정적으로 바뀌었다. 이것만으로도 대단한 수확이다.

이제 우리 소프트웨어 기술도 국제적으로 경쟁할 수 있는 지위에 있다는 점을 확인하고 자신감을 가지는 계기가 되었다고 생각한다. 이번에 개발한 도구도 아직 개선의 여지가 많이 남아있다. 좀 더 안심하고 사용할 수 있는 사이버세계를 만드는데 일조한다는 자부심으로 열심히 즐겁게 연구해보자고 다짐해본다. 준비된 자에게는 언젠가는 기회가 온다는 기대를 가지고 말이다. —

5

세미나와 워크샵


교육과학기술부 · 한국연구재단 지정 우수연구센터
소프트웨어무결점연구센터 Workshop
장소 : 제주 휘닉스 아일랜드 벨라테라스 오렌지동 2층

2010-03-15 Completeness of Pointer Program Verification by Separation Logic
Makoto Tatsuta , National Institute of Informatics

2010-01-07 The 3rd ROSAEC Center Workshop
ROSAEC members , ROSAEC Center

2010-01-05 The Linear Temporal Logic of Rewriting Model Checker in Maude
Kyungmin Bae , Univ. of Illinois at Urbana-Champaign

2009-12-23 Constrained Environment Inference for Verification of Multi-Threaded Programs
Corneliu Popeea , Max Plank Institute for Software Systems

2009-12-22 Local Reasoning for Read-Copy-Update
Hongseok Yang , Queen Mary, Univ. of London

2009-12-16 Types and Recursion Schemes for Higher-Order Program Verification
Naoki Kobayashi , Tohoku University

2009-12-15 The Twilight Zone: from testing to formal specifications and back again
Koen Claessen , Chalmers University of Technology, Gothenburg, Sweden

2009-12-14 The Sketching Approach to Program Synthesis
Armando Solar-Lezama , Massachusetts Institute of Technology

2009-12-13 Parallel Programming in Fortress
Sukyoung Ryu , Computer Science, KAIST

2009-12-13 Separation Logic from the Perspective of Program Analysis
Hongseok Yang , Queen Mary, University of London

2009-11-06 Hoare Logic for the Coinductive Trace-Based Big-Step Semantics of While
Keiko Nakata , Tallinn University of Technology

2009-10-29 An Evolution-Centric Perspective on Software Testing
Gregg Rothermel , University of Nebraska-Lincoln

2009-10-29 Automated Verification via Separation Logic
Cristina David , National University of Singapore

2009-10-14 Logical Relations and Compositional Compiler Correctness
Chung-Kil Hur , University of Cambridge

2009-09-08 Type Checking Program Generators Using the Record Calculus
Tankut Baris Aktemur , Ozyegin University, Turkey

2009-09-05 Deriving Invariants by Algorithmic Learning, Decision Procedures, and Predicate Abstraction
Bow-Yaw Wang , Institute of Information Science, Academia Sinica

2009-09-05 Comparative Study on Software Model Checkers as Unit Testing Tools based on an Industrial Case Study
Moonzoo Kim , Computer Science, KAIST

2009-08-31 Survey Workshop on Corpus-based Shape Analysis
ROPAS Members, Research on Corpus-based Shape Analysis , SNU

2009-08-31 Components and Static analysis
Bruno Oliveira , ROSAEC Center [talk slide]

2009-07-30 Approximate Inference: Decomposition Methods with Applications to Computer Vision
Kyomin Jung , Computer Science, KAIST

2009-07-09 The 2nd ROSAEC Center Workshop
ROSAEC members , ROSAEC Center

2009-07-08 A Brief Overview Assume-Guarantee Reasoning via Learning
Bow-Yaw Wang , Institute of Information Science, Academia Sinica

2009-06-19 Interactive proving and programming in Coq
Hugo Herbelin , INRIA

2009-06-02 WYSINWYX: What You See Is Not What You eXecute
Gogul Balakrishnan , NEC Laboratories America, Inc.

2009-05-12 Bi-abductive inference for reasoning about heaps
Peter O'Hearn , Queen Mary, University of London

2009-05-11 From Separation Logic to Systems Code
Peter O'Hearn , Queen Mary, University of London

2009-04-30 Symbolic Model Checking Property Specification Language
Ji Wang , National University of Defense Technology

2009-04-17 소형무인헬기용 임베디드 소프트웨어의 구조와 적용
김두현 , 건국대학교 인터넷미디어공학부

2009-04-10 Relationship between Curry-Howard-correspondence and semantics
Gyesik Lee , ROSAEC Center

2009-02-13 Survey Workshop on Static Analysis of Binary Code
MES Lab. Members , SNU

2009-02-04 Survey Workshop on Static Analysis for Malware Detection
SE Lab. Members , SNU

2009-01-21 Domain-Specific Abstract Interpretations : Experience Reports
Jerome Feret , INRIA/ENS/CNRS, France

2009-01-17 Survey Workshop on Class Analysis
ROPAS Members, Research On Program Analysis System , SNU

2009-01-14 P2P Botnet Enumeration and Banking Trojan Crimwares
Byung-Hoon Kang , University of North Carolina at Charlotte


2008-12-30 Randomized Active Testing for Concurrent Programs
Chang-Seo Park , UC Berkeley

2008-12-24 Shared Subtypes : Subtyping recursive parameterized algebraic data types
Ki-Yung Ahn , Portland State University

2008-12-12 Alleviating False Alarm Problem of Static Buffer Overflow Analysis
Youil Kim , KAIST

2008-11-21 ROSAEC Center Kick-off Workshop
ROSAEC members , ROSAEC Center

2008-11-19 Destabilization of Adversarial Organizations with Strategic Interventions: Computational Organization Theory Approach
Il-Chul Moon , KAIST

2008-11-17 Analyzing and Inferring the Structure of Code Changes
Miryung Kim , University of Texas, Austin

2008-10-01 Statistical Machine Translation
Changbum Park , Seoul National University

2008-09-30 Abstract Interpretation and Application to the Static Analysis of Safety-Critical Embedded Computer Software
Patrick Cousot , École Normale Supérieure, Paris

6
방문기

Stata Center, Research Atmosphere, Seminar

글쓴이 : 정영범
박사과정, 서울대학교 컴퓨터공학부 프로그래밍 연구실
방문지 : MIT, Cambridge, Massachusetts, USA
방문기간 : 01/15/2009 ~ 05/31/2009

Stata Center

Stata Center는 겉보기에도 재미난 곳이지만 내부에도 흥미로운 점들이 아주 많다. 1층 로비에 있는 커다란 칠판은 많은 학생들이 공고를 하거나, 토론을 할 때 유용하고, 때로는 아이들의 낙서 공간이 되기도 한다. 건물 내부 구조도 기괴하여 처음에는 상당히 헤매기 십상이다. 계단 5칸짜리 정도 되는 공간만을 왕복하기 위한 엘리베이터도 있다. 정말 쓸데 없어 보이는 공간이다. 장애우를 위한 것이라고 하기에도 너무 쓸데 없어 보인다. 손이 닿을 수 없는 곳곳에 배치된 소의 형상, 경찰차 등등... 이 모든 것이 조금이라도 기발하고 창의적인 생각을 이끌어내려는 노력이리라.

Stata Center의 가장 큰 장점을 꼽는다면 바로 1층에 체육관이 있다는 것이다. 일과 시간 중에 언제라도 잠시 내려 가서 운동을 하고 다시 올라 올 수 있다. 서울대에서는 302동에서 체육관을 가려면 차를 타고 내려가야 한다. 물론 공대에도 체육관이 있기는 하지만 내려 갔다가 올라오는데 삼사십분은 족히 걸린다. 햇살이 좋은 날 MIT나 BU(Boston University) 캠퍼스를 돌아다니면 좋은 몸매를 드러낸 학생들을 자주 볼 수 있다. 운동에 접근하기도 용이하고, 다양한 운동들을 체계적으로 즐길 수 있다. 요트나 카약 같이 한국의 대학교에서 쉽게 접할 수 없는 스포츠들도 있다. MIT에서는 온라인 상에 리그를 만들어 학생들이 좋아하는 종목을 보다 적극적으로 즐길 수 있게 한다. 자신의 레벨에 맞는 리그에 선수로 등록하고, 팀에 합류해서 리그 경기에 나갈 수 있다. 나도 indoor soccer B level league에서 MIT EECS 팀 소속으로 경기를 했다. 2승2무1패로 리그2위로 playoff에 진출했지만 아쉽게도 1차전 탈락했다.

이 곳에는 1층에 어린이집까지 있다. 우리나라에서 상상도 못 해본 일이다. 학교안에 어린이 집이 있다니! 자그마한 야외 놀이터도 있다. 어린 아이들을 가진 연구원들은 아이들을 데리고 왔다가 보고 싶을 때 언제든지 가서 보고, 나중에 집에 데려갈 수 있다. 아이를 가진 연구원들의 천국이다.

Research Atmosphere

MIT의 연구 분위기는 서울대와는 사뭇 다르다. MIT CSAIL 연구실 교수진만 96명이다. 거기에 research scientist와 학생들까지 합하면 엄청난 규모의 연구소이다. 그럼에도 교류가 활발하다. 이곳에서는 지도교수와 학생간의 관계가 서울대처럼 일대일이 아니다. 많은 학생들이 여러 교수들과 연구를 동시에 진행하고, 물리적인 연구실의 개념도 모호해서, 학생이 앉을 자리가 없으면 흔쾌히 남는 다른 연구실 자리를 쓰곤한다. 내가 있는 7층의 라운지로 나가면 학생들이 혹은 교수님들이 이야기를 나누는 모습을 쉽게 볼 수 있다. 각 연구실들도 우리 서울대처럼 문을 꽁꽁 닫아 두지 않는다. 수시로 학생들이 드나들며 다른 연구실 학생들과 교류한다. 비서들이 라운지에 가끔가다 먹을 거리를 장만해놓아 여럿이 모여 담소를 나누며 즐기기도 한다. 그리고, 모두가 같이 쓰는 대형냉장고, 차나 커피, 토스트기, 전자레인지등은 서로를 자주 만나게 하고, 친숙하게 하는 의도된 장치들이다. 학생들이 연구에만 집중할 수 있게 많은 사람들이 도와준다. 청소나 서류 작업 같은 일들은 모두 CSAIL 소속의 직원들이 다 해준다. 모두들 하나 같이 친절하다.

“지금 하고 있는 연구를 세상해서 가장 잘 아는 사람은 바로 그 연구를 지금하고 있는 사람이다.” 이 곳에서 나를 지도해준 Martin Rinard 교수님께서 나에게 제일 처음 해주신 말씀이다. 연구는 자신감 있게 하면 된다. 즐기지 않으면 연구가 진행될 수가 없다고 했다. 맞는 말이다. 원래 연구라는 것이 할일없는 사람들의 유흥으로 시작하지 않았나? 새로운 것을 알아가고, 유용한 일들을 해내는 성취감은 연구에 이미 내제되어 있는 즐거움이다. 어떤 요인들이 그 즐거운 연구를 해야하는 일로 만드는 지 다시 한번 생각해 봐야 한다.

Seminar

이 곳에서는 훌륭한 세미나들이 자주 열린다. 제일 위에 있는 사진은 한 강의실에서 Patrick Henry Winston이 How to Speak라는 주제로 talk을 했을 때의 사진이다. 우연히 내 모습도 찍혔다. 이 talk은 어떻게 효과적으로 presentation을 하는지에 대한 것이었다. 내용도 훌륭했지만 이 talk자체가 presentation을 어떻게 해야 하는지를 아주 잘 보여주었다. 강의실의 크기나 조명, 시간, 소품 같은 평소에 내가 생각하지 않았던 부분들에 대해서도 신경쓰라고 조언해 주었다. 또, 발표 처음에 여러분들이 어떤 것을 얻어가게 될 것인지 약속하라고 했다. 감사의 인사로 마무리하는 것이 아니라, 나의 약속을 이 발표가 지켰는지를 확인하는 것으로 발표를 맺으라고 했다. 자신감이 넘치는 발표는 그렇게 만들어진다. 내가 청중들에게 원하는 것을 주었으니 내가 감사할 필요가 있는 것이 아니라는 생각. 물론 적당한 감사는 꼭 필요하다고 이야기 하지만, 발표가 훌륭하지 못했다면 그것은 당당한 감사가 아니라 오히려 사과에 가까운 것이 된다.

이 talk의 분위기를 좋아했다. 작은 강의실에 사람이 꽉 찼었고, 그래서 저렇게 발표자 바로 옆에 앉아서 발표를 들어야 했다. 열의와 총기가 가득한 많은 눈동자들과 그들을 하나하나 집중하게 만드는 발표자의 행동이나 말들... 정말 많은 것을 배웠다. 이 곳의 세미나들을 더욱 훌륭하게 만드는 것은 refreshment 에 있다. CSAIL의 모든 seminar들은 시작전에 간단한 다과가 준비되어 있다. 이것들은 비싼 점심값에 굶주리고(?), 비타민을 충분히 섭취할 수 없는 가난한 유학생들에게 굉장한 도움이 된다. Tutorial seminar 같은 경우는 점심으로 피자까지 제공했다. 음식은 사람의 마음을 누그러뜨리는 효과가 있다. 발표자와 청중들 모두를 부드럽게 하는 효과가 있다. 게다가 많은 학생들의 참석을 유도한다. 서울대에서 강제로 세미나를 듣게 하기 위해 세미나 수업을 개설할 것이 아니라 refreshment를 이용하면 어떨지 생각해본다. 물론 발표 내용의 충실함이 전제가 되어야 겠지만...

Class

두 가지 수업을 청강했었다. 1학년 학부생을 위한 Mathematics for Computer Science와 우리의 컴파일러 수업이라고 볼 수 있는 Computer Language Engineering 수업이다. 전자는 서울대 교육 과정 중에 겪어 보지 못한 나에겐 너무나 신선한 방식으로 수업이 진행된다. 교실에 들어서면 호텔 만찬장과 같은 모습의 원형탁자가 점점이 흩어져 있다. 각 원형탁자 옆에는 화이트 보드가 하나씩 세워져 있다. 4면의 벽 각각에는 2개의 스크린이 있어서 강의 자료를 보여주고 있다. 학생들은 무작위로 앉고 싶은 탁자에 앉게 된다. 수업은 30분 정도의 강의와 그 후에 이어지는 토론을 통한 문제 풀이로 이루어진다. 강의는 학생들에게 오늘 배울 내용을 설명하고, 간단한 문제를 풀며 학생들에게 학습할 준비를 시키는 정도로 진행한다. 노교수의 강의가 끝나면 문제가 출제 되고, 각 탁자에 앉은 학생들끼리 토의를 하면서 같이 문제를 해결한다. 해결한 문제는 그때그때 화이트 보드에 적고, 조교들과 교수는 하나 하나 돌아가면서 학생들이 제시한 해답을 살펴본다. 문제마다 무작위로 선출된 학생 한명이 화이트 보드를 보면서 해답을 설명해야 한다. 행여 문제를 틀렸다고 해서 감점이 되거나 하지는 않는다. 혹시 재밌는 접근 방법이 나오면 교수가 모든 학생들이 알 수

있도록 공개하기도 한다. 토론하는 과정도 재미있고, 문제도 수준별로 다양하게 출제되어 흥미를 잃는 학생이 없도록 하는등 정말 훌륭한 수업이었다. 컴파일러 수업은 MIT 나 서울대나 비슷한 내용이었다. 역시나 숙제가 중요한 수업인 것도 마찬가지 였다. 재밌는 점은 두 명의 교수가 번갈아가면서 수업을 한다는 것. 각자의 연구 분야에 맞게 깊이 있는 수업이 가능했다.

English

가장 큰 수확은 내가 영어를 못하는 이유를 잘 알았다는 것. 나에게 있어 영어는 수학이나 과학보다 내 능력이 떨어지는 분야였다. 지금도 마찬가지지만... 근데 영어랑 수학은 전혀 다른 분야이다. 내 능력이 떨어지는 게 문제가 아니다. 영어는 머리가 아닌 눈, 귀, 입, 손으로 해야 한다. 영어를 머리로 하려고 애를 쓰다가 잘 되지 않으면 좌절하고, 난 영어 체질이 아닌가 보다라고 생각하고 포기하면 절대 늘지 않는다. 애초에 많은 시간과 노력을 들이지 않으면 할 수 없는게 영어이다. 되든 안되든 영어로 말을 하고 듣는 생활을 꾸준히 해야한다. 노력은 결코 배반하지 않는다. 반면 내 경험상 수학 분야에서는 노력이 나를 배반하기도 한다.

영어로 말을 잘 하려면 일단 영어로 말을 해야하는데 영어를 잘 못하면 말하고 싶지가 않다. 결국 영어로 말을 쭈욱 못하게 된다. 영어 회화에 있어서 가장 중요한 것은 자신감과 뻔뻔함이다. 그럼 자신감은 어디서 나오나? 문법, 단어? No! 발음이 가장 중요하다. 아무리 문법이 정확하고, 올바른 단어를 사용했다 하더라도 발음이 틀리면 상대방이 절대로 못 알아 먹는다. 이런 일이 계속되면 자신감을 잃게 된다. 하지만, 문법 다 틀리더라도 중요한 단어 하나만 발음을 정확하게 하면 상대가 잘 알아듣는 경우가 많다. 나에게 있어서 가장 큰 문제가 발음이었다. 고등학교때 ceremony를 씨리모니로 발음을 했다가 모든 급우들에게 놀림을 받은 아픈 기억 후로 나에게 영어발음은 민망함일 뿐이었다. 책을 읽을때도 나만의 가상 발음을 만들어 놓고, 읽어 왔다. 그 발음은 영어권 사람들은 전혀 쓰지않는 새로운 것이다. 그 당시에 영어 선생님들이 발음을 단 한번만이라도 내게 제대로 가르쳐 주었다면 하는 점이 지금도 가장 원망스럽다. 난 작년에 American Accent Training이라는 책을 읽기 전에는 f, t, v, th 발음을 어떻게 해야 하는지 조차도 모르고 있었다. 비슷하게 발음하는 것은 차치하고 라도 최소한 어떻게 해야하는 지는 알고 있었어야지. 만약 이 글을 읽는 사람 중에 모르는 사람이 있다면 유투브에서 english pronunciation으로 검색해서 딱 한 시간만이라도 보기 바란다. 그 한시간의 투자가 10년전에만 이뤄졌더라면 현재의 내 영어 실력은 확실히 달라졌을 것이다. 그 수많은 시간동안 나만의 가상 발음으로 읽어 왔던 글들을 살아있는 영어에 비슷한 발음으로 연습을 했을테니... 영어로 말을 하려면 뻔뻔함이 중요하다고 이야기 했다. 뻔뻔함은 내가 외국어를 하고 있다는 사실에서 나온다. 내가 영어를 못한다고 무시하는 사람은 자기의 언어밖에 못하는 경우가 많을 것이다. 그런 사람들은 외국어를 하는 심정을 이해하지 못한다. 반면에 MIT 에 있는 영어권이 아닌 나라에서 온 학생들은 내가 발음, 문법, 단어가 틀리더라도 이해하고 전혀 나를 불편하게 만들지 않는다. 그들은 외국어를 하다 틀리는 것을 전혀 창피하게 생각하지 않는다. 영어는 일단 발음만 공부한 후에 계속 연습해서 막 말하기를 바란다. —

OPLSS, MLPA, CADE

글쓴이 : 임현승
박사과정, 포항공과대학교 컴퓨터공학과 프로그래밍언어 연구실
방문지 : Eugene, Oregon, US & Montreal, Quebec, Canada
방문기간 : 07/22/2009 ~ 08/10/2009

1. Summer School on Theory and Practice of Language Implementation

July 23–31, 2009, University of Oregon, Eugene, Oregon, US

Oregon Programming Language Summer School (OPLSS) 은 올해로 8번째 개최된 깊이 있고 체계화된 교육 프로그램입니다. 올해는 Theory and Practice of Language Implementation이라는 주제로 흥미진진강좌가 많이 개설되었는데요, 본 글에서는 그 중에 인상 깊었던 강좌 몇 개를 소개하고자 합니다. OPLSS-09 홈페이지에서 강좌 소개뿐만 아니라 강의 자료 및 동영상도 제공하고 있으니 자세한 내용은 홈페이지를 참고하시길 바랍니다.
우리에게 여름학교 기간 내내 맛있고 영양가 있는 다양한 음식을 단돈 $5에 제공해준 학생식당 앞에서 포항공대를 비롯하여 우리나라 대학 대부분에서는 프로그래밍 언어를 공부하시는 교수님들이 없거나 한명뿐이어서 관련 수업을 다양하게 수강하기가 참 어렵습니다. 따라서 OPLSS 같은 여름학교를 통해서 각 분야의 문가들에게 직접 가르침을 받는 것은 정말 좋은 기회라고 생각합니다. 이번 여름학교에는 한국에서는 저와 함께 역시 포항공대에 재학 중인 박종현군과 카이스트의 김철주씨, 이철우씨, 그리고 창병모 숙명여대 교수님도 참여하셨습니다.

사진 2 박종현군(왼쪽)과 필자(오른쪽),
우리에게 여름학교 기간 내내 맛있고 영양가 있는 다양한 음식을
단돈 $5에 제공해준 학생식당 앞에서

이번 여름학교에서는 총 8개의 주제로 32개의 80분 강좌가 진행되었는데요, 아침 9시부터 오후 5시까지 매일 4개의 강의를 들어야 하는 강행군이 계속되었습니다. 이러한 강행군 덕에 오른쪽 사진에 보이는 바와 같이 수업 사이사이 쉬는 시간에 단잠에 취하시는 분들이 꽤 보였습니다. ㅎㅎ 물론 강의내용을 숙지하고자 쉬는 시간에도 열공하시는 학구파 분들도 꽤 있었습니다. ㄷㄷㄷ 자, 그럼 이제부터 각 강의 주제에 대해 간략하게 소개의 한 말씀 올리도록 하겠습니다. ^^

A. Algorithmic Program Synthesis

Ras Bodik University of California – Berkeley
서울대 공순호씨의 trip report를 보니 Bodik 교수님이 SAS-09에서 초청강연을 했더군요. 강의내용은 초청강연의 240분 확장판이라고 생각하시면 될 것 같습니다. 공순호씨가 잘 정리해주셔서 이에 대한 내용은 공순호씨의 trip report를 참조하시길 바랍니다. 쿨럭…

B. Continuations to Go

Olivier Danvy Aarhus University
Danvy 교수님은 Continuations에 관한 한 최고의 지성이죠. 또한 굉장히 열정적입니다. 번외편으로 거의 4~5시간 동안 Practi

[1] 어디까지나 개인적인 추정으로 확실하지는 않습니다. 2002년부터 올해까지 OPLSS 홈페이지는 다음 사이트에서 찾아볼 수 있습니다
http://ix.cs.uoregon.edu/~jallen/

cal PhD Requirements라는 주제로 총합 242페이지에 달하는 방대한 양의 조언을 해주셨습니다. (주옥 같은 조언을 많이 해주셔서 좋기도 했지만 참 힘들었습니다. (>.<))

이 강의는 이론과 실재를 겸비한 참 재미있는 강좌였습니다. Continuations에 대해서 강의해주시고 예제 코드를 각자 continuation-passing style (CPS) 바꿔보는 연습을 할 수 있었습니다. 총 3개의 연습문제가 있었는데요, 심심하신 분들을 위해 문제를 소개하자면 (Objective Caml 문법으로 문제를 소개하겠습니다.) 첫 번째로 다음 두 함수를 어떤 list comprehensions (map, reverse, foldr 등)도 사용하지 않고 순순하게 CPS 함수로 작성하는 것입니다.

ㄱ. val list_of_suffixes: ' a list -> ' a list
e.g., list_of_suffixes [1;2;3;4;5] -> [[1;2;3;4;5]; [2;3;4;5]; … ; [5]; []]
ㄴ. val list_of_prefixes: ' a list -> ' a list
e.g., list_of_prefixes [1;2;3;4;5] -> [[]; [1]; … ; [1;2;3;4]; [1;2;3;4;5]]

참고로 CPS transformation의 기본 원칙은 다음과 같습니다.
1. Names intermediate results.
2. Sequentializes their computations.
3. Introduces continuations (first-class functions).

두 번째 문제는 주어진 regular expression과 string list가 일치하는지 검사하는 함수를 작성하는 것입니다.
아래의 예제를 만족하는 함수를 CPS 함수로 작성하시면 됩니다. 힌트는 tail-recursive 함수입니다.

type regexp = Elem of string | Seq of regexp * regexp
e.g., Seq (Seq (Elem "임", Elem "현"), Elem "승")
= Seq (Elem "임", Seq (Elem "현", Elem "승"))
= ["임";"현";"승"] != ["임";"현승"]

세 번째 문제는 두 개의 lists [x1; … ; xn], [y1; … ; yn]이 주어졌을 때, [(x1,yn); … ; (xn,y1)]을 계산하는 CPS 함수를 작성하는 건데요, 여기에서 주목할 점은 n 값을 모른다는 것입니다.

앉아서 수동적으로만 이론 강의를 듣는 것이 아니라 적극적으로 continuation에 대해 생각해보고 참여하는 멋진 수업이었습니다. 여담으로 대학원에 처음 들어와서 Selective CPS transformation에 대해 공부했던 기억이 새록새록 떠오르더군요. ㅎ

사진 2 매일 저녁 1~3시간 가량 학교 주변을 산책하고 중간에 근처 아이스크림 가게, 커피숍, 술집 등을 들리더군요. 처음 두세 번은 함께 했는데 나중에는 지쳐서 그냥 숙소에서 쉬게 되었습니다. ^^;

C. Control-flow Analysis of Higher-Order Languages

Matt Might University of Utah

이번 강좌에서는 함수형 언어에서 많이(?) 사용하는 정적분석 기법 중 하나인 Control-flow analysis (CFA) 에 대해 자세하고 깊이 있게 다뤘습니다. 간단히 말해 CFA는 함수호출 시에 어떤 함수가 호출될지를 보수적으로 추정하는 기법인데요, 예를 들어 코드에 f (x) 라는 함수호출이 있을 때 f가 어떤 함수인지 정적으로 어림잡는 것입니다. 0CFA, 1CFA, k-CFA, … , Environment analysis 등 종류도 다양한데요, Matt 교수님이 어찌나 빨리 발표를 하든지 처음에는 좀 따라가다가 엎친 데 덮친 격으로 몇 가지 배경지식까지 가정하고 진행하다 보니 나중에는 정신을 놓게 되더군요… (>.<) 다행히(?) 다른 많은 친구들도 정신줄을 놓고 있었다는 것을 나중에 알았습니다. ㅎㅎ

D. Managed Runtime Environments: Implementations and Opportunities

Chandra Krintz University of California – Santa Barbara

Krintz 교수님은 Java, C#, Python과 같은 언어에서 사용되는 bytecode 및 runtime environments에 대해서 소개해주셨습니다. 또한 Web service와 business applications 에서처럼 여러 언어로 작성된 응용 프로그램을 managed runtime environ-

ments를 이용해서 어떻게 support 할수 있는지에 대해서도 다루시고 끝으로 현재 진행중인 cloud computing 연구에 대해서도 소개해 주시더군요. 굉장히 자신감이 넘치시는 분이었는데, 강의 내용이 워낙 방대해서 기억에 남는 것은 별로 ... ^^;

E. Multi-Threaded Programming and Transactional Memory

Yannis Smaragdakis University of Massachusetts, Amherst

병렬 프로그래밍이 앞으로 대세가 될 것이라는 점에는 누구나 공감하실 거라 생각합니다. Smaragdakis 교수님은 병렬 프로그래밍의 한 유형인 multi-threaded programming에 대해서 소개하였습니다. Multi-threaded programming 방법으로 먼저 현재 가장 많이 사용되고 있는 monitor style programming을 설명하고 현재 활발하게 연구되고 있는 transactional memory에 대해 소개했습니다. 병렬 프로그래밍이라고는 박성우 교수님의 Parallel Programming이라는 대학원 과목을 수강하며 OpenMP와 MPI를 이용하여 C programming을 해본 게 전부인 저에게는 상당히 유익하고 재미있는 강의였습니다.

Monitor style programming에 대해서 잘 모르시는 분들을 위해 간략히 소개하면 mutexes (lock) 와 condition variables (wait, signal, broadcast) 을 이용해서 프로그래밍 하는 방법으로 Java에서는 모든 Object가 mutex 이고 condition variable 입니다. lock은 synchronized로, wait는 Object.wait로, signal은 Object.notify로, broadcast는 Object.notifyAll로 구현되어 있고, C와 C++에서는 PThreads를 이용해서 monitor style programming이 가능합니다. 자세한 내용 및 예제는 OPLSS-09 홈페이지에서 제공하고 있는 강의자료를 참고하시기 바랍니다.

Transactional memory는 monitor style programming의 단점을 (no modularity, lock is a global property, lack of composability) 보완하기 위한 대안으로서 database에 있던 개념을 차용 것입니다. Monitor style programming의 단점을 잘 시사하는 가장 유명한 예는 여러분들도 다들 아실 bank account 문제입니다.

```
class Account {
        int balance = 0;
        public synchronized int withdraw(int amt) { ··· }
        public synchronized void deposit(int i) { ··· }
}
class Client1 {
        public synchronized void move (Account a1, Account a2) {
        a2.deposit(a1.withdraw(10));
        }
}
```

위 Java code에서 withdraw 함수와 deposit 함수를 아무리 올바르게 구현했다고 하더라도 두 함수를 이용하여 구현된 move 함수는 올바르지 않을 수 있습니다. Transactional memory는 atomic code sections을 (all-or-nothing) 도입함으로써 위와 같은 문제를 우아하게 해결하는데요, 현재는 성능이 별로 안 좋다는 큰 단점이 있습니다. Software Transactional Memory: Why Is It Only a Research Toy 라는 제목으로 ACMQUEUE에 개제된 article은 이와 같은 문제점을 강력히 시사하고 있는데요, Smaragdakis 교수님의 말을 빌리면 현재의 TM은 80~90년대의 garbage collector가 처한상황과 유사하기에 앞으로 더많은 연구가 진행되고 컴퓨팅 파워가 증가하면 transactional memory가 병렬 프로그래밍을 쉽고 안전하게 할 수 있는 가장 좋은 방법 중에 하나가 될 것이라 더군요. 앞으로 병렬 프로그래밍이 엄청 중요해질 것이라는 점은 자명하니 TM의 성능이 더 좋아지기를 기대해봐도 좋을 것 같네요. 관심 있는 분들을 위해 참고로 현재 Haskell에서 STM을 사용해볼 수 있답니다.

여담으로 Smaragdakis 교수님도 PhD rants and raves 라는 주제로 번외강의를 해주셨는데요, 상당히 흥미롭더군요. 이 분 자료는 꽤 유명한 것 같던데, 관심 있으신 분들은 링크를 따라가서 한번 읽어보시길 추천합니다.

F. Program Analysis for Computing Symbolic Complexity Bounds

Sumit Gulwani Microsoft Research

이 강좌에서는 특정 procedure에 특정 input이 주어졌을 때, procedure 안에 선언된 특정 control-location이 (loop의 conditional expression 등) 얼마나 많이 방문되는지를 계산하는 방법에 대해 다루었습니다. 물론 upper bound를 계산하는데요, 이것을 symbolic bound라고 한다고 하더군요. Symbolic bound를 계산하기 위해 먼저 procedure의 invariants를 찾아야 되는데요, colorful logic, fixpoint brush, program transformations 등과 같은 방법을 이용해서 invariants를 찾아낼 수 있다고 합니다.

이러한 방법을 통해 invariants를 찾고 나면 symbolic bound를 계산할 수 있다더군요. 주제 자체는 굉장히 흥미로웠습니다.

G. Pointer analysis

Ondrej Lhotak University of Waterloo

Lhotak 교수님은 Java와 같은 object-oriented languages를 위한 포인터분석기법에 대해서 소개해주셨습니다. 포인터분석은 주로 C 언어에서 많이 하는 줄로만 알았는데 Java에서도 많이 하고 있나 보더군요. 상속이나 캐스팅, 오버로딩 등 때문에 포인터분석이 어렵고 또한 필요하다네요. 한가지 흥미로웠던 점은 서로 별 연관이 없어 보이던 control flow analysis와 pointer analysis 사이에 유사점이 꽤 된다는 것이었습니다.

H. Garbage Collection and the Metronome GC

David Bacon IBM Research

이번 강좌에서는 IBM에서 개발했던 garbage collector 중에 하나인 Metronome Garbage Collector에 대해 소개했습니다. Metronome은 현재 IBM에서 개발한 WebSphere Real-time Java Virtual Machine에서 사용되고 있습니다. 강의는 기술적인 세부사항을 다루기 전까지는 정말 흥미로웠고 재미있었습니다. Garbage collection의 방법으로는 크게 네 가지가 있습니다: “Stop the world”, parallel, concurrent, and incremental garbage collection. Metronome은 그림 5에서 볼 수 있듯이, parallel, incremental, and concurrent garbage collection을 실시간으로 지원합니다. 흥미로운 데모로 실시간 피아노연주를 일반 Java garbage collector를 이용하여 처리한 경우와 Metronome을 이용해서 처리한 경우를 비교해보시기 바랍니다. 자세한 Metronome 설계 및 구현 방법에 대해서는 OPLSS-09 홈페이지에서 제공하고 있는 강의자료를 참고하시기 바랍니다.

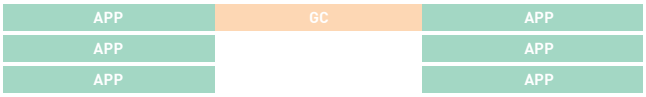


그림 1 “Stop the world” garbage collection

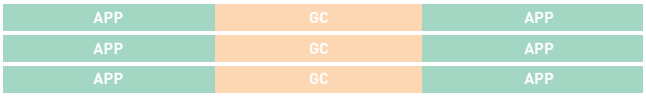


그림 2 Parallel garbage collection

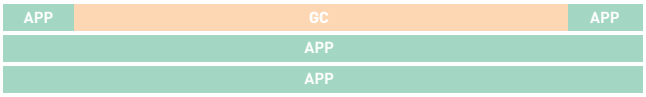


그림 3 Concurrent garbage collection

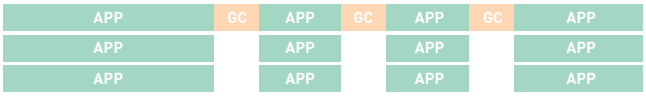


그림 4 Incremental Garbage collection

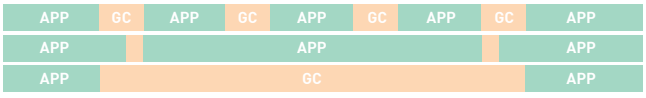


그림 5 Parallel, incremental, and concurrent garbage collection

I. Abstract Interpretation

Patrick Cousot École Normale Supérieure / New York University

Cousot 교수님이 abstract interpretation에 대해서 세 번에 걸쳐서 강의를 해주셨는데요, 강의가 끝날 때마다 매번 ASTRÉE Static Analyzer 광고를 잊지 않으시더군요. ㅎㅎ ASTRÉE는 Airbus 비행제어소프트웨어에 런타임에러가 없다는 걸 증명하는데 사용되었다고 하더군요.

총평 : 각 분야의 최고 전문가들에게 직접 강의를 들을 수 있다는 것은 참 즐거운 일인 것 같습니다. 강의 스케줄이 빽빽하여 다소 힘들기는 했지만 강사들의 친절하고 자세한 설명 덕분에 만은 것을 배울 수 있었던 귀중한 경험이었습니다. 기회가 되신다면 적극 추천하고 싶네요. 위의 왼쪽 사진은 저녁 피크닉 때 (피크닉도 두세 번은 했던 것 같네요.) 찍었던 사진이고 오른쪽 사진은 저녁식사 후 교정을 거닐다가 한 컷 찍어봤습니다.

SAS 2009 , PCC 2009

글쓴이 : 최원태
석사과정, 서울대학교 컴퓨터공학부 프로그래밍 연구실
방문지 : UCLA, USA
방문기간 : 08/07/2009 ~ 08/17/2009

순호형과 UCLA에서 열림 SAS 2009 학회와 PCC 워크샵에 참석하였다. 올해에는 SAS와 LICS 두개의 학회와 4개의 부속 워크샵이 함께 진행되었는데 행사가 SAS로 시작하여 PCC로 끝난 덕분에 우리는 행사기간 내내 UCLA에 머물렀다.

학회 참가자 기숙사 건물에 묵었다. 기숙사 건물은 ㅁ자로 생겨서 건물 밖에서는 일단 중앙의 공터로 들어온 다음에 중앙 공터에서 건물 내부로 들어가게 되어있었다. 공터에는 나무와 벤치가 있는데 외부의 시선에서 차단된 하늘과 트여있는 공간이란 느낌이 좋았다. 학회는 기숙사 지역 중앙의 학생회관 같은 건물(Covel Commons)에서 진행되었다. 학교 곳곳에 Commons란 이름이 붙은 건물이 있는데 대체로 1층에 카페테리아나 식당 있고 2,3층으로 세미나실이 있었다. 그런 건물들을 Commons라고 부르나보다.

SAS 2009

9일부터 11일까지 3일간은 SAS Conference가 진행되었다.
Rastisloav Bodik의 Invited Talk과 한양대팀, Songtao Xia의 발표가 흥미로웠고 Patrick Cousot 교수님이 ROSAEC 워크샵에서의 최광무교수님처럼 모든 발표에 코멘트를 하는 것이 인상적이었다.

Rastiv Bodik,
Algorithmic Program Synthesis with Partial Programs and Decision Procedures

프로그래밍을 하다보면 세부사항을 컴퓨터가 자동으로 채워주었으면 하는 경우가 있다. 이 세션에서는 그 중에서도 “올바르지만 느린 구현”과 “새로운 구현의 뼈대”를 바탕으로 컴퓨터가 “새로운 구현”을 도출하게 하는 연구에 대해 발표하였다. 뼈대에 채워질 코드가 돌아다니는 문제공간이 유한하다면, Model Checker로 원래의 프로그램과 동일한 결과를 보장하는 코드를 찾아낼 수 있을 것이다. 컴파일시간에 빈칸을 채우는 기법(Programming by Sketch)와, 동적 시간에 빈칸을 채우는(Angelic Operator) 두 가지 기법을 소개하였다.

최초의 구현이 올바르다는 것을 검증하면 그 위에서 안전하면서도 효율적인 프로그램을 기계를 이용하여 유도할 수 있는듯 해 매력적이었다. Theorem Proving 커뮤니티에서는 증명에서부터 안전한 구현을 도출하는 연구를 하고 있다고 들었는데, 증명에서 비효율적이지만 안전한 구현을 도출하고 거기서부터 Programming by Sketch를 이용하여 점진적으로 개선해 나갈 수 있으면 어떨지.

Songtao Xia,
Inferring Dataflow Properties of User Defined Table Processor

DB 질의가 포함된 프로그램을 분석 할때, 프로그램의 실행 흐름과 질의문의 의미를 같이 고려하여 “사용되지 않는 Table Entry”등을 찾아내는 분석기법을 소개하였다. 이 작업은 C#과 연동되는 DB 질의실행기를 최적화하는데 사용된다고 하였고 그래서인지 C#으로 작성된 프로그램의 Control Flow를 구하는 분석을 소개하는데 절반에 가까운 시간을 할애하였다. DB 질의문이나 Script언어를 고려한 분석은 큰 규모의 프로그램을 검증하고 최적화하려면 넘어야하는 산이라고 생각하던 터라 더욱 관심있게 들었다. 이론적으로 새로울 것이 없고 노력은 많이 필요하여 학계에서 다루기에는 적합하지 않을지 몰라도 분석 커뮤니티가 더 커진다면 산업체를 중심으로 연구가 본격화되지 않을까 싶다.

Clement Hurlin,
Automatic Parallelization and Optimization of Programs by Proof Rewriting

이 세션에서는 주어진 프로그램에 대한 Seperation Logic기반의 증명이 존재할 때 Framerule을 이용해 프로그램을 (Observationaly) 동일한 다른 프로그램으로 다시 쓸 수 있다는 것을 보이고, 이것을 이용한 최적화를 제안하였다. Seperation Logic에 관심 있는 사람이라면 읽어볼 만한 (어렵지 않고 흥미로운) 논문이다. 하지만, 이론적으로 흥미로운데 반해 실험결과가 없어 아쉽기도 하였다. 아마도 Seperation Logic으로 산업현장의 소스코드에 대한 만족할만한 증명을 제공할 수 있는 분석기가 아직 없기 때문이 아닐까 추측해본다.

David Schumit and KyungKu Doh,
Abstract Parsing : Static Analysis of Dynamically Generated String Output Using LR-Parsing Technology
프로그램 실행중에 만들어질 수 있는 문자열을 직접 모으는 대신 문자열이 파싱된 결과를 모음으로서 빠르게 문자열이 올바른 모양을 지니고 있는지 검증하는 기법을 소개하였다. 발표는 David Schumit교수님이 내용을, 도경구교수님이 실험부분을 담당하여 진행했다. 내용을 익히 알고있던 터라, 내용 자체보다는 어떻게 발표하는지 살펴보는데 집중하였다. 어떤 논문이 되었던지 질문시간을 포함해서 30분 남짓 주어지는 발표시간동안 모든 디테일을 다루기는 불가능하다. 때문에 발표는 무엇을 포기하고 무엇을 남겨야 하는지 잘 선택하는 문제가 된다. Schumit 교수님은 세부사항을 최대한 감추면서 옛날이야기를 하듯 부드럽게 직관을 전달하였다. 세부사항을 언급해야할 필요가 있으면 Formal한 정의를 살짝 보여준 뒤 곧바로 딱 하나의 예제를 통해 직관이 실제 어떻게 적용되는지를 보여주었다. SAS 셋째날 있었던 Schumit 교수님의 다른 발표도 비슷한 느낌으로 진행된 것으로 미루어, 고유한 발표방식인것 같다. SAS와 PCC를 합쳐서 20명이 넘는 발표자의 발표를 보니 노련한 발표자일 수록 세부사항을 전달하는데 시간을 많이 할애하지 않았다. 직관을 전달하되 꼭 필요한 디테일을 선별해서 첨가하는데 학회 발표의 묘가 있다.

E.D.Clark,
27 Years of Model Checking, E.D.Clark
Model Checking이 지나온 길에 대해 개괄적으로 설명하였다. 아마도 2007년에 우리 연구실을 방문하여 발표했던 내용과 크게 다르지 않을 것이다. Model Checking에 대해 이미 알고있는 사람은 무용담을 듣는 가벼운 기분으로, 잘 모르는 사람은 “Sound하지만 Termination이 보장되지 않는 검증기법” 이라는 정도의 이해를 얻어서 돌아갈 수 있지 않았나 한다. 개인적으로는 Predicate Abstraction에 대한 Model Checking 대부의 통찰을 듣고 싶었는데 들을 수 없었다. Talk의 주제가 포괄적이어서 Predicate Abstraction만 자세히 다루기에는 시간이 부족했을 것이다. Model Checking 전체를 아우르는 입장에서는 Software Model Checking이란 분과에서 사용하는 하나의 State Abstraction 기법에 불과할지도 모르겠다.

PCC 2009

15일에는 PCC Workshop이 있었다.
두개의 invited talk이 모두 영양가 있었다. Andrew Appel의 발표는 내용이 익숙하지 않아 절반정도 밖에 따라가지 못해 아쉬움이 남는다. 나머지 발표중에는 Juan Chen과 David Pichardie의 발표가 들을만 했다.

Kelly Hayhurst,
Mdeing the Gap, An effort to aid the transfer of formal methods technology
산업 프로젝트에 “Formal Method”를 적용하는 어려움에 대해 이야기하였다. NASA의 비행기 개발 부문(우주 개발이 아닌) 에서 “표준 개발 방법론”을 개발하고 문서화하는 일을 담당하는 분이 Speaker였다. 프로그램 분석에 대해 잘 모르는 분이었고, 그래서 더 가치가 있는 세션이었다고 생각된다. 산업체에서는 치밀하게 설계된 수천페이지에 달하는 공정 메뉴얼을 따라 항공기를 설계하고 제작한다고 한다. 지금은 소프트웨어 검증 비용이 항공기 전체 개발 비용의 절반에 육박하기 때문에 “Formal Method”를 도입해야할 필요성은 절감하지만 어느 산업체에서도 공정에 포함시키지 못하고 있다. (Airbus 360은 유일한 시범 케이스다) 도입하지 못하는 이유는 “Formal Method” 가 무엇을 검증해주는지 “품질관리” 측면에서 설명할 수 없기 때문이라고 한다.

David Pichardie
Towards a Certified Lightweight Array Bound Checker for Java Byte Code
정적 분석을 통해 Java byte code에서 array bounday check를 가능한 없애서 실행속도를 높이는 논문이 있는데, 그것에 PCC를 적용하여 실행 전에 정말 없애도 되는 검사들만 사라졌는지 실행전에 한번 확인하는 기술을 소개하였다. PCC의 한 갈래인 abstract carrying code에서는 trusted base의 크기를 줄이기가 매우 어렵다. abstraction이 올바른지 확인하기 위해서는 abstraction을 바탕으로 다시 한번 고정점 계산을 수행해야하고, 이를 위해서는 분석기 전체가 필요하기 때문이다. 저자는 이런 문제를 fix-point 확인에 최적화된 Semantics를 디자인하여 해결하였다. Domain과 언어에 따라 최적화 Semantics는 천차만별일 것이므로 이 작업이 직접적인 도움이 되는 일은 없겠지만, abstract carrying code를 사용할 일이 있다면 한번쯤 참고할만할 것 같다.

Our Talk
우리 차례는 오전 11시 30분 부터 12시 까지 30분이었다. 순호형이 발표하였고 노트북에서 프로젝터를 잘 인식하지 못해 시작이 약간 지연되었지만, 발표 자체는 연습할때에 비해서 부드럽게 잘 진행되었다. 13,14일 이틀간 준비하면서 어떻게 하면 디테일을 배제하면서도 뜬구름잡는 느낌이 들지 않게 발표할지를 주로 고민하였다. 분석예제를 거의 에니메이션 수준으로 보여주면 많은 말을 하지 않아도 직관적으로 이해할 수 있겠다 싶어서 50컷에 달하는 분석 예제 에니메이션을 만들어 보여준 것이 주효한듯 하다.

질문시간에 Naoki Kobayashi 교수가 Syntax Check 이상의 것을 할 수 있냐는 질문을 하였고 점심시간에도 두 사람이 (Ewen Denney, Sagar Chaki) 같은 질문을 하였다. Syntax만 가지고 얼마나 대단한 성질을 체크할 수 있겠느냐는 뉘앙스였다. 오후세션에 Syntax Check를 넘어설 궁리를 하다가 의외로 쉽게 방법을 찾았다. 우리는 Syntax만으로도 훌륭하다고 생각하여 다른 시도를 해볼 생각을 못했는데, 잘 모르는 사람의 눈에는 이상하게 보였던 모양이다. 다양한 관점을 접하고 공개적으로 평가받는 것이 엎마나 중요한지 새삼 느꼈다.

학회를 알차게 보내려면

처음 가보는 학회라서 제대로 즐기지(?) 못하는 기분이 들 때가 여러번 있었다.
어떤 준비가 필요한지 적어둔 것이 있어서 옮겨본다.

대화!

발표를 듣지 않아도 논문을 읽으면 무슨 연구를 했는지는 알 수 있다. 보다 중요한 것은 대화를 통해 다른 사람들의 통찰과 조언을 얻는 일인 것 같다. 학회에 오는 사람들의 관심분야와 얼굴(!)을 알아두고 발표할때 질문거리들을 만들었다가 쉬는시간에 이것저것 대화를 해 보면 좋을것 같다. 영어를 못하면 정보라도 있어야 이야기를 풀어나가기 쉬울테니까.

식사시간

점심시간이 2시간이 나 되기 때문에 식사시간을 다른 참석자들과 같이 보내는 것이 중요하다. 쉬는시간에는 친한 사람끼리 2-3명씩 무리지어 이야기를 나누기 때문에 처음 참석하는 사람들은 대화상대를 찾기가 쉽지 않지만, 식사시간에는 작게는 4명, 많게는 10명까지 같이 앉을 수 있다. 일단 같이 앉으면 통성명도 하고 대화도 하는 것이 인지상정이다. 이것을 마지막날 깨닳아 아쉽다.

영어!

말하는 능력보다는 듣는 능력이 훨씬 중요하게 느껴졌다. 상대방의 말을 따라가는 동안은 한단어 두단어로 응수해도 대화가 이어졌지만, 상대의 말을 제대로 듣지 못하고 “Pardon?”을 외치는 순간 대화의 흐름이 뚝 끊기는 것을 느꼈다.

발표 직전 노트북 확인

노트북에서 프로젝터를 인식하지 못하는 일이 발표 시작전에 종종 벌어졌다. 프로젝터를 인식하는 동안 사라진 시간 자체는 유동적으로 조절할 수 있지만, 뜻하지 않은 사건을 만났을때의 긴장을 피하기 위해서라도 미리 작동하는지 확인해두는게 좋겠다.

UCLA

첫날 만찬이 끝난뒤 소화도 시키고 학교지리도 알아볼겸 UCLA를 돌아보았다. 넓직한 평지에 여유있게 자리잡은 건물과 학교 곳곳에 가득한 수목이 아름다웠다. 서울대도 산속에 위치해있어 자연과 가까운 것으로는 빠지지 않지만, 건물들이 있는 구역과 그렇지 않은 곳이 완전히 다른 분위기인데 비해 UCLA캠퍼스는 둘이 자연스럽게 섞여있었다. 배우고 싶은 부분이다.

둘째 날 점심시간에 학생회관을 찾아가 25$에 자전거를 빌렸다. 기숙사 키카드와 신용카드를 제시하니 별로 어렵지 않게 빌릴 수 있었다. 알고보니 기숙사 키카드가 UCLA 전체에 통용되는 ID 카드였다. 구내 식당부터 시작해서 도서관과 스포츠센터까지 거의 모든 시설을 이용할 수 있다. 우리는 식당과 자전거대여에 밖에 사용하지 않았지만...

자전거를 빌린 덕에 월요일부터는 행동반경을 UCLA 학교 밖으로 넓혔다. 덕분에 인근 마트에서 먹거리를 사와 아침,점심을 저렴하게 해결하고 남는 돈으로 학교 밖에 있는 제법 그럴듯한 식당에서 저녁을 즐길 수 있었다. UCLA 인근이 제법 잘사는 동네라고 들었는데 거리도 식당도 깨끗했다. 그만큼 비쌌지만.

우리가 방문해있던 기간은 여름방학이라 미국학생들은 집에 가있고 계절학기를 방문한 외국인(중국,일본) 학생들과 연습때문에 학교를 떠날 수 없는 운동부 소속 학생들이 대부분이었다. 그래서 장소에 따라 캠퍼스가 아주 다른 느낌을 주었다. 계절학기가 진행중인 기숙사 인근은 중국인이 바글바글하고 기숙사만 벗어나면 운동부 학생들이 여기저기서 뛰고 체조하고 있었다. —

7

블로그 : Rosaec Minutes

papers & stories

Tuesday, November 17th, 2009

교수님들께,
교수님 랩의 대표 논문들을 모집하겠습니다.
세부 사항은 행정원이 공지드리겠습니다.
센터 웹과 뉴스레터에 실리게 됩니다.
뉴스레터는 종이의 장점을 살린 인쇄물로 만들겠습니다.
대표 논문을 보내실 때, 한가지 부탁드리겠습니다.
대표 논문마다 작은 “이야기”를 보내주십시요. 웹과 뉴스레터에 논문마다 한 단락으로 붙이겠습니다.
100자 이내로, 연구에 대해서 공유하고픈 에피소드, 키워드, 리뷰 구절 등을 써 주십시요.

이유가 있습니다.
* 우리가 즐겨야겠습니다. 재미없는, 에피소드없는 연구는 피하자, 가 구호입니다. (억지로라도 만드셔서 같이 즐길수있게)
* 마음을 움직이면 그 분야가 발전합니다, 가 가설입니다. 연구에 대한 지극히 개인적인 “이야기”들,
  우리의 웹/뉴스레터에 그런 것들이 드러나도 좋다고 봅니다.
* 센터 평가에도 도움될 것입니다.

한 예를 첨부합니다. 교수님들의 개성을 발휘해주십시요.
예) “논문제목”
3개월간 방문했던 XXX교수 와의 공동연구 결과. 좋은 연구를 짧은 기간에 같이하게 된 아주 즐거운 경우였다.
연구중에 이거다 싶은 느낌의 연구였다. YYY이라는 분야를 프로그램 검증에 사용할 수 있는 시초를 만든 연구였다.
논문리뷰중에 “The paper is NNNN…”라는 평을 받았다.
키워드: AAA, BBB, CCC

이광근 드림



visitors & updates

Saturday, October 3rd, 2009

추석 잘 보내고 계시는지요. 저는 미국출장 중입니다. 이곳 달은 꽉차보이지는 않았습니다.
시차때문인가봅니다. 비행기에서 "엄마를 부탁해"를 읽었습니다.
좋은 소설을 읽고나면 몸 속이 달라지는 느낌이잖씁니까.
심장이나 허파나 간 어딘가가요.
우리가 꾸미는 논문이나 sw들은 사람들의 어디를 만져주는 걸까요?
소설같지는 않겠지요. 문학이나 노래가 위대합니다.

센터에는 외국에서 오는 다양한 레벨의 방문 연구진들이 있습니다.
흥분되는 결과가 나오기도 하고, 당장은 그저그런 경우도 있습니다.

교수급
지난 3개월간 방문했던 Bow-Yaw Wang 교수(Academia Sinica, Taiwan)와의 공동연구 이야기입니다.
좋은 연구를 짧은 기간에 같이하게 된 아주 즐거운 경우였습니다.

Algorithmic Learning 이라는 분야를 프로그램 검증에 사용할 수 있는 시초를 만든 연구를 같이 하고 갔습니다.
키워드는 algorithmic learning, randomness, invariant inference입니다. 정영범, 공순호 학생과의 팀웍이 완벽했습니다.

연구를 하면서 "이거다"싶은 느낌이 가끔 있습니다. 그 경우였습니다.
내년 1월 VMCAI(Verification, Model Checking, Abstract Interpretation) 학회에서 발표합니다.

논문리뷰중에 "The paper presents a fresh/novel perspective to invariant inference and
has a potential to lead to a new line of subsequent work in this direction." 라는 평을 받았습니다.
드문 리뷰 기분 좋았습니다.
논문 마지막 손보기위해 모여 앉았던 이른 아침의 캠퍼스 벤치, 그 추억이 소중합니다.
센터 "tech memo" 페이지에 논문이 있습니다. "seminar" 페이지에는 논문 세미나 슬라이드가 있습니다.
Wang교수는 현재 베이징 칭화대에 2년간 방문하고 있습니다. 계속 확장연구 함께하고 있습니다.

고참 박사과정
싱가폴 국립대(NUS)의 Cristina David가 3개월간 방문 목적으로 10/1일 합류했습니다.
NUS의 Wei Ngan Chin교수의 박사과정 학생입니다. 거의 학위를 마친 학생입니다.
그동안 좋은 논문을 꾸준히 냈던 친구입니다.
루마니아 출신 여학생입니다. 같이 뭔가가 나올거라고 봅니다.

신참 석사과정
독일 Aachen 공대(RWTH)에서 Lucas Brutschy가 6개월 방문 목적으로 9/27일 합류했습니다.
지난 6월 Aachen에서 열린 SSV(System SW Verification)에서 가까워진 Bastian Schlich 박사의 제안이었습니다.
Sparrow관련 정적분석기술 익히고 싶다고 합니다.

방문 연구진들은 다양한 계기로 오게 됩니다. 아는 교수들에게 제가 먼저 제안하는 경우도 있고,
아는 교수가 먼저 제안을 하는 경우도 있습니다.

방문 연구진들은 경우마다 다른 지원을 합니다. Bow-Yaw Wang교수의 경우는 왕복 비행요금과 원룸 비용을 지원했습니다.

Cristina David는 왕복 비행요금과 원룸 비용 + 월 60만원 지원합니다.
Lucas Brutschy는 편도 비행요금과 원룸 비용 + 월 60만원 지원합니다.

방문연구진과 함께 하는 연구는 뭐랄까요 소풍가는 기분입니다. 작은 여행을 같이가는 것 같은.

이광근 드림 __

center visitors

Wednesday, April 29th, 2009

올해는 황사가 없는대신 기온이 쌀쌀한 봄입니다. 5-6월에 센터의 방문객들이 있습니다.

5/11- 5/12: Prof. Peter O'Hearn (U of London) http://www.dcs.qmw.ac.uk/~ohearn
"Separation Logic"의 태두. 힙 메모리를 다루는 프로그램의 검증에 유용한 방식으로 확인되고 있는 Separation Logic.
아내와 함께 한국온 길에 들른다고 합니다. 저는 개인적인
느낌으로 이 사람이 장차 John Reynolds와 Hongseok Yang과 함께
큰 상을 받을것 같더군요.(Turing Award?…)
5/11에 서울대에서 Distinguished Lecture Series를 합니다.(for generalaudience) 참석부탁드립니다.
5/12에는 specialist seminar를 합니다.

6/1-6/5: Dr. Gogul Balakrishnan (NEC Research) http://pages.cs.wisc.edu/~bgogul
binary code(assembly code) static analysis의 선두주자입니다. 저희가 586코드 분석할 일이 있을게고,
미리해 본 사람의 이야기를 들을 수 있는 일주일이 될 것 같습니다.

미확정이지만, 여름중에 Prof. Robert Harper(CMU) http://www.cs.cmu.edu/~rwh
type-based programming language research의 대가, 라는 이야기 필요없을듯 합니다.
박성우교수님 초청으로 오시는 것으로 압니다. 센타 세미나도 하시고 좋은 중요한 시간이 될 것 같습니다.

학생들도 방문합니다.
5월-8월: Will Klieber (phd student, CMU)
9월-12월: Cristina David (phd student, NUS)
* 뉴스레터는 wiki방식으로 만드는 것을 해 볼까 합니다. 실험.
* 관련분야 계간 무크지(whatever that means)를 꾸밀계획도 하고있습니다.
(opinion channel, with light touch, deep resonance)

이광근 드림 __



5 Flagship Projects(5FP) minutes

Wednesday, April 29th, 2009

저희 센터의 5개 선도 프로젝트의 성공이 센터 성공의 주요 축입니다.
5개 선도 프로젝트들의 진행에 대해서 소개 하겠습니다.

인공위성 SW :
이욱세 교수님과 한양대/서울대 학생 4명이 팀이 되어 KAIST인공위성연구센터에서 개발한
과학위성 탑제 소프트웨어에 특화된 오류검증기 개발이 진행중입니다. 위성 소프트웨어에만 있는
특별히 분석할 오류들이 몇개 있습니다.
그 중에서 우선, 위성에 탑제되는 소프트웨어가 항상 주어진 일을유한시간내에 끝내는지(termination analysis)를
확인해 주는 분석기를 만드는 길을 떠났습니다. C 소스 입니다.

무인 항공기 SW :
김유단 교수님의 도움으로 구체적인 대상 SW와 대상 오류 문제를 찾기위한 미팅을 계획하고 있습니다.
무인항공기 기술팀과 소프트웨어 오류검증기술 팀이 편하게 모이는 시작입니다. 4월말/5월초로 잡고 있습니다.
무인항공기 관련 학계와 연구소 분들과 소프트웨어 검증기술자가 같이 모여서 실망하고 희망하는 시간이
될 것을 상상하고 있습니다.

금융 SW1 :
최근 금융sw 에서 놀라운 일이 일어났습니다.
지난 금요일(2/26) 밤입니다. 우리에게는 거의 완벽한 기회가 열렸습니다.
JP Morgan에서 지난 수십년간 개발해서 전세계 90%이상의 credit trader 들이 사용하는
“standard CDS model”(뭔지 저도 모릅니다)
코드를 공개했습니다. C 입니다. 상당히 영향력있고 중요한 sw라고 합니다. 있을 오류를 찾아달라고 전세계에 공개한 것입니다.
팀을 구성해서 길을 나서려고 합니다. 뉴욕 Deutch Bank의 Jeff Polakow박사가 도와주고 있습니다.
참고 : http://zerohedge.blogspot.com/2009/02/isda-open-sources-cds-model-challenge.html

금융 SW2 :
Deutch Bank에서 Haskell로 금융 분석 시스템 sw를 완성했다고 합니다.
개발중에 골치아픈 오류가 있었다고 합니다: unit conversion과 rate consistency라는 겁니다.
그 Haskell코드와 위의 오류에 특화된 검증분석기를 만드는 일도 진행하려고 합니다.
Haskell코드를 우리가 받아야 하는데, 관련 logistics문제를 어프로치하고 있습니다.

산업체 SW 1:
국내 모 회사에서 제품에 들어가는 100KLoC정도 되는 C 병렬 코드의 특정 오류를
허위경보율 5% 이내이고 안전하게자동 검증해 주는 분석기 개발이 곧 시작될 듯 합니다. 서울대 팀이 맡게 됩니다.

산업체 SW 2 :
신승철 교수님이 공장 자동화에 쓰이는 코드들을 검증하는 도구개발에 관한 위탁과제를 제안하셨습니다.
진행 부탁드렸습니다.

mal-JavaScript나 mal-binary를 감지하는 분석기를 개발하는 연구가 시작되었습니다. 서울대/KAIST 팀이 나섰습니다.
mal binary static detector의 경우, 바이러스 백신 스캐너의 수준을 한단계 올릴 수 있다고 생각합니다.

이광근 드림

8

참여 연구실 소개 & 방문 연구원 소개

01 서울대 박종우 교수님 _ 로봇자동화 실험실

서울대학교 로봇 자동화 연구실은 1995년에 설립되어 현재까지 45명의 박사 및 석사를 배출하였으며, 현재 석박사 통합과정 8명, 박사과정 1명, 석사과정 6명으로 구성되어 있다. 본 연구실은 로봇에게 인간과 유사한 작업 능력 및 동작 생성 능력, 동작 제어 능력을 부여하기 위하여 move ment coordination and learning, mobile manipulation systems, physical-based simulation 및 vision에 관한 연구를 진행해 왔다. 그 중 몇 가지를 소개하면 다음과 같다.

Mobile Manipulator Motion Planning

본 연구는 mobile manipulation system을 위한 motion planning 기법을 개발하는 것을 목표로 한다. Korea Institute of Science and Technology(KIST)에서 제공하는 CIROS를 대상으로 하고 있는데, CIROS는 각각 7자유도인 양 팔과 nonholonomic base로 구성된 하드웨어 플랫폼이다. 가정이나 실버타운에서 서비스하는 것을 목적으로 개발되는 로봇이기 때문에 인간의 환경 속에서 물건 잡기, 문 열기, 엘리베이터 타기 등의 다양한 작업을 하기 위하여 실시간으로 충돌을 회피하고, 필요한 동작을 계획 및 생성하고 제어한다.

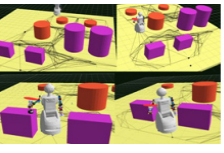
Open Robot Dynamics Library

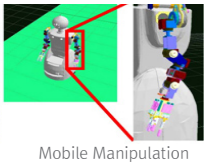
동역학 시뮬레이션 소프트웨어는 하드웨어의 개발 시간을 단축시키고 비용 절감의 효과가 있으며, 정량적인 물리 데이터를 얻어낼 수 있다. 또한 하드웨어에 직접 연구결과를 적용해 봄에 있어서 따를 수 있는 위험을 제거하고 한정된 하드웨어 자원을 대체할 수 있는 장점이 있다. 본 연구실에서 개발한 동역학 라이브러리는 3D환경에서 로봇을 쉽게 모델링하고 시뮬레이션 해 볼 수 있어 로봇을 개발하는데 있어 디자인에서부터 동작을 계획하고 제어하는데 까지 로봇을 개발하는데 널리 사용될 수 있다.

Bio-Inspired Robotics

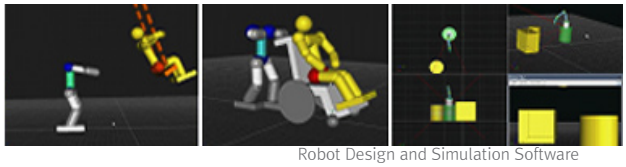
자연계의 생물체는 오랜 시간 진화하여 환경에 최적화 된 상태이기 때문에 그러한 생물체를 모방하여 필요한 기능을 구현하는 것은 그 한계를 극복하여 현재의 기술을 한 단계 뛰어 넘을 수 있도록 하는데 큰 힘이 된다. 본 연구에서는 생체의 근육을 모방한 스프링을 모델링하고 최적화 알고리즘을 사용하여 점프하는 동작을 생성함으로써 기존보다 약 20%가량 성능을 향상시켰다.

위에서 언급된 연구를 통해 개발된 기법들은 인간형 로봇의 모션 control 엔진으로 활용되어 지능형 로봇 개발에 기여할 수 있다. 로봇에게 인간에 좀 더 가까운 수준의 작업능력을 갖게 하고 자연스러운 모션을 생성할 수 있게 하므로, 인간과 함께 생활하면서 인간의 일을 대신하게 될 미래형 로봇의 원천 기술을 제공할 것이다. 하지만 파급 효과는 로봇 분야에 국한되지는 않는다. 컴퓨터 그래픽스에서 애니메이션과 같은 디지털 컨텐츠 산업에도 많은 영향을 끼칠 것이다. 기존방식으로 애니메이션을 제작했을 때와 달리 제작 시간과 소요 노동력을 줄일 수 있고, 훨씬 자연스러운 움직임을 보여줄 수 있을 것이다. 더 나아가 HCI (Human-robot interaction) 분야에서의 동작 인터페이스 개발에도 적합한 기술을 제공할 수 있다. 뿐만 아니라, 본 연구는 제어 알고리즘, 최적화 기법, 기구학과 동역학 해석, learning 알고리즘 등 다양한 수학적, 공학적 이론이 적용되므로 이론적 연구 발전에도 중요하다. —

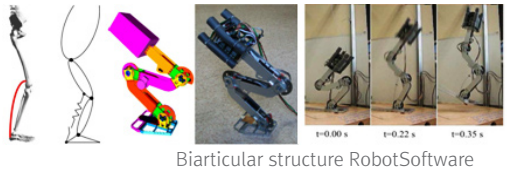




Mobile Manipulation



Robot Design and Simulation Software



Biarticular structure RobotSoftware

02 포항공과대학교 박성우 교수님 _ 프로그래밍 언어 연구실

포항공과대학교 프로그래밍 언어 연구실은 박성우 교수의 지도하에 통합과정 학생 3명, 석사과정 학생 1명으로 구성된 연구실입니다. 시작한지 4년이 지난 저희 연구실은 프로그래밍 언어 이론과 전산논리학을 바탕으로 새롭고 도전적인 연구 주제를 찾아서 성장하고 있습니다.

저희 연구실의 장기적 비전은 멀티코어 컴퓨팅 환경을 위한 함수형 병렬 프로그래밍 언어를 개발하여 교육과 연구에 이용하는 것입니다. 2000년대 중반 이후 마이크로프로세서는 멀티코어 구조로 전환되었으며 이런 추세는 메니코어 구조가 대세가 되기 전까지 적어도 10년 이상 유지될 것으로 전망됩니다. 그러나 멀티코어 컴퓨팅 환경을 주 대상으로 하는 함수형 언어는 최근에야 본격적 개발이 시작된 상황입니다.

멀티코어 컴퓨팅 환경을 위한 함수형 병렬 프로그래밍 언어를 개발한다는 장기적 목표 아래 저희 연구실에서는 관련 연구를 수행하고 있습니다. 언어 정의 부분에서는 병렬 프로그래밍에서 데이터 및 코드 전송을 안전하게 지원하는 타입 시스템을 이미 완성하였습니다. 언어의 모든 단계를 직접 개발하려는 노력 과정에서 처음에는 계획하지 않았던 파생 연구도 시작되었습니다. 예로서 파서 생성기-생성기가 있습니다. 언어 개발시 제작해야 할 중요한 도구 중 Yacc과 같은 파서 생성기가 있습니다. 보통 새로운 언어를 개발할 때마다 해당 언어용 파서 생성기를 별도로 제작하는데, 파서 생성기-생성기를 이용하면 언어 명세로부터 해당 언어용 파서 생성기를 만들 수 있습니다. 또 다른 예로서 모든 언어에 이용할 수 있는 모듈 시스템이 있습니다. 원래 목표는 우리 언어에 이용할 모듈 시스템을 설계하는 것이었는데 모든 언어에 이용할 수 있도록 목표를 확장하여 별도의 연구로 진행하고 있습니다.

저희 연구실에서 최근 시작한 또 다른 장기적 연구 주제는 자동정리증명기를 이용한 연역적 프로그램 검증입니다. 프로그램이 주어진 명세를 만족하는지를 자동정리증명기를 이용하여 연역적으로 검증하는 기술은 테스팅, 모델체킹, 정적분석 등과 같은 기존의 검증 기술의 한계를 극복하며, 최근 임베디드 소프트웨어, 시스템 소프트웨어, 운영체제 검증 등에 본격적으로 적용되고 있습니다. 저의 연구실에서는 분리논리(separation logic)에 기초한 연역적 프로그램 검증 시스템 개발을 목표로 하고 있으며, 현재 논리체계 설계와 같은 이론적 연구를 수행하고 있습니다.

이러한 장기적 연구 주제 외에도 저희 연구실에서는 프로그래밍 언어와 전산논리에 관련된 다양한 연구를 진행하고 있습니다. VHDL이나 Verilog를 대체할 수 있는 함수형 하드웨어 기술언어의 설계를 완료하고 구현을 준비하고 있으며, 양상논리(modal logic)와 관련된 이론적 논리학 연구도 진행하고 있습니다. 최근에는 데이터베이스 분야로 연구 영역 확장을 시도하고 있으며, 프로그래밍 언어 이론과 전산논리를 데이터베이스 분야와 연계시키는 연구 주제를 찾고 있습니다.

03 항공대 안준선 교수님 _ 시스템소프트웨어 연구실

컴퓨터 기술의 양적 진보에 힘입어 대상이 되는 정보의 범위가 팽창함에 따라 컴퓨터 시스템은 일상생활 곳곳에 점점 깊숙이 스며들고 있고 의식하지 않아도 우리는 점점 더 많은 프로그램을 실행시키며 살고 있다. 그러나 소프트웨어 기술에 대한 질적 성장은 이러한 양적 팽창을 따라가지 못하고 있어 소프트웨어의 안전성과 소프트웨어 시스템의 지속 가능한 발전에 대한 의심이 커져가고 있으며, 소프트웨어 기술 자체에 대한 인식도 낮아지고 있는 상황이다.
한국항공대학교 정보통신전공 시스템소프트웨어 연구실에서는 프로그래밍언어와 프로그램 분석에 기술을 기반으로 안전한 컴퓨팅 환경의 구축에 도움이 되는 기술을 개발하고자 설립되었다. 2008년 첫 석사과정 졸업생을 배출한 이래 4명의 석사과정을 배출하였고, 현재 안준선 교수와 2명의 석사과정 그리고 다수의 학부 연구생들이 함께 연구를 수행하고 있다. 본 연구실에서 현재 수행하고 있는 주요 연구 주제를 소개하면 다음과 같다.

– 유비쿼터스 컴퓨팅을 위한 프로그래밍 환경 구축
본 연구는 안전한 유비쿼터스 컴퓨팅 환경을 효과적으로 구축하기 위한 프로그래밍 환경을 개발하는 연구이다. 한국과학재단 기초과학연구사업(2006.3~2009.2)의 지원으로 숙명여대 및 한양대학교와 공동으로 연구를 수행하였으며, 유비쿼터스 환경을 위한 선언적 언어인 PDL(Policy Description Language), 상황 적응 실행 시스템 및 관련 프로그래밍 지원도구 등에 대한 연구를 수행하였다. PDL은 유비쿼터스 환경을 개체(entity)들과 개체들 간의 관계로 묘사하고 상황 변화에 따른 적응 동작을 기술할 수 있는 구조를 가지며, 아울러 개체들에 대한 접근 권한을 동적 상황에 맞게 명세할 수 있는 상황인식 접근제어(context aware access control) 모델 기반의 접근제어 규칙을 기술할 수 있다. 실행 시스템은 PDL로 기술된 유비쿼터스 환경을 입력으로 받아 유비쿼터스 환경을 이루는 개체들의 상태를 관리하고 상황의 변화에 따른 적응 동작을 자동으로 수행시키는 기능을 가진다. 관련 분석도구로서 상황 충돌 분석기는 PDL 명세에 대하여 상호 충돌되는 적응 동작이 수행될 수 있는지를 미리 검사해준다. 이를 통하여 실행시간에 동적인 상황 변화에 의하여 충돌하는 동작이 작동하는 것을 미리 방지할 수 있다.

– 웹 응용프로그램 보안 취약성 분석
웹 응용프로그램에 존재하는 문자열 삽입 공격 취약성(String injection attack vulnerability)을 자동으로 검출해주는 정적 분석기를 개발하는 연구로서 소프트웨어 무결점연구센터의 지원으로 수행하고 있다. 본 연구에서는 문자열 분석을 위한 분석 도메인(domain) 설계에 있어 관심 보안취약성에 최적화된 도메인을 사용하여 분석의 정밀도와 효율을 동시에 얻고자 하였으며, 요약 파싱(abstract parsing)과 같은 문법 기반 문자열 분석 방법을 채용하여 동적 웹페이지에 대한 취약점 분석 범위를 넓히는 방법을 모색하고 있다. 주요 대상 취약성은 SQL 삽입 공격 취약성, 교차 사이트 스크립팅(Cross Site Scripting) 공격 취약성 등 입력값 검증(Input Validation)과 관련된 다양한 취약성을 대상으로 하고 있으며, 현재 PHP를 대상 언어로 하여 분석기를 개발하고 있다.

– 소프트웨어 보안 취약성 DB구축
본 연구는 소스코드 내에 존재하고 있는 보안취약점을 자동으로 검출해주는 소스코드 취약점 자동진단 시스템의 개발과 관련한 연구로서, 2009년 행전안전부의 주관 및 지원 아래 인터넷진흥원을 중심으로 (주)지티원, (주)파수닷컴과 정보보호학회 소프트웨어보안연구회의 협동연구로 진행되었다. 본 연구실에서는 검사 대상이 되는 취약성에 대한 검출 규칙을 RDL(Rule Description Language)을 사용하여 개발하고, 검출 규칙 및 취약성 관련 DB를 XML의 형태로 구축하는 작업에 참여하였다. 구축된 정보는 취약성 분석엔진의 입력 정보로서 Java 및 C 프로그램에 존재하는 약 270여종의 대표적인 취약성을 검사하는데 사용된다. 현재 진단시스템은 행전안전부에서 발주하는 모든 소프트웨어에 적용될 예정으로 1차년도 개발이 완료되었으며, 성능 향상 및 확장을 위한 연구를 계속 진행할 예정으로 있다.

이러한 연구 외에도, 본 연구실에서는 항공우주 및 통신 소프트웨어에 대한 분석 및 지원도구의 개발과 같이 항공우주 특성화 대학의 강점과 전공의 특성에 부합하는 연구를 장기적인 발전방향으로 추진하고 있다. 이러한 연구들을 통하여 안전한 소프트웨어 개발을 위한 작지만 알찬 결과들을 쌓아감으로써, 정보통신 주요 수출국의 위치에 부합하는 소프트웨어기술 선진국으로의 도약에 미약하나마 공헌하고자 한다.

04 Bruno C. d. S. Oliveira

Research Professor, ROSAEC Center

A Programming Language Adventure in the East

September 14, 2009

Prior to joining the ROSAEC Center, I was working as a research assistant at the University of Oxford (UK), where I also did my PhD (or DPhil, as it is called there). Before that I worked for about one year at the IBM DB2 team, and I also did some programming work back in Portugal (my home country).

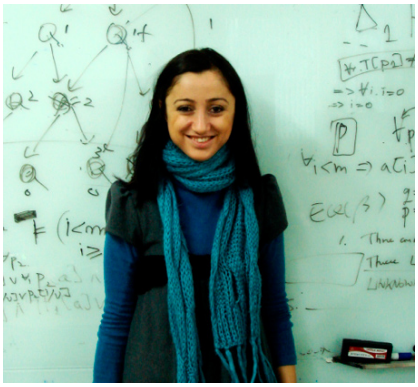
My decision to do research in programming languages had mostly to do with my dissatisfaction with the current state-of-the art of software development. The experience that I had in industry showed me how fragile the whole development process was. At IBM I was part of a performance team and I was responsible for the development of a scripting tool for writing both stress testing and regular testing scripts. Unfortunately, during my time at the performance team, we hardly managed to conduct any kind of stress testing. This was because our main product was essentially crashing all the time. In the end we spend all the time helping the testing team to conduct regular testing. To make a long story short, my group had 2 out 5 teams dedicated to testing, which is quite a significant investment just for testing! Most of the bugs in our product were derived from a poor understanding of the whole system by the programmers, and to the lack of support of existing programming tools for the development of scalable and reusable concurrent software.

Arguably programming language research can help to improve the state-of-art of the current software development process in two ways: by developing new programming languages or programming language features; or by improving the tools and techniques used to develop software with existing programming languages. Back in Oxford I primarily worked on Datatype-Generic Programming (DGP), which is a programming style in which mundane programming tasks such as defining equality or a toString function for a newly added data structure are automated by the definition of a generic function that works for every possible data structure. From a software engineering point of view generic functions are useful because they remove the need for a lot of boring to write and error-prone code, which has the additional benefit of removing the need for testing, verifying and maintaining that code.

My decision to join the ROSAEC center had of course a significant personal factor involved, since I wanted to have a living experience in east Asia. However, from a professional point of view, the ROSAEC center distinguished itself from other research institutions in Asia for its focus on the practical application of programming tools to an industrial setting. I was quite impressed to learn that the research conducted in ROPAS (the predecessor of the ROSAEC center) lead to the development of Sparrow: a commercial static analyzer for C programs. This is very much in line with Cristopher Strachey's (the founder of the Programming Research Group at the University of Oxford) ideals, which once said: "It has been my personal view that the separation of practical and theoretical work is artificial and injurious.". I am hopeful that I can combine my expertise in generic software components with the programming language research done in the ROSAEC center, to help with the continuous effort to conduct excellent programming language research that can have an impact in an industrial setting. —

05 Cristina David

Ph.D. Student, National University of Singapore, Singapore
01/10/2009 ~ 30/12/2009



My SNU experience

My name is Cristina David, and I was happy to visit the ROPAS laboratory in SNU for three months, from September to December 2009.

At the end of my visit to ROPAS laboratory, I have to admit that, when i arrived here three months ago, I was a bit worried about the interaction with my new colleagues. Given that my stay here was going to be pretty short, I had a few doubts on whether or not I will be able to integrate in the research group and collaborate with the rest of the members. However, in just a few days, I realized that my worries were unfounded, as everyone here was friendly and helpful. I was continuously baby-sited by my colleagues, always giving me directions, printing maps for me, advising me on what is interesting to visit, worrying on whether i like the food or borrowing me blankets when the weather became too cold. It is very difficult to imagine just how kind and helpful they are without actually meeting them.

Another aspect that surprised me about ROPAS is how close the members are among each other. In three months I have never heard any argument between them. It felt like a big family.. always helping each other, discussing every problem and searching solutions together. Overall, research was lots of fun with lots of exchanged ideas. What I appreciated most is the welcoming atmosphere, where every opinion is appreciated . Everyone is open to collaborations and any member interested in a particular project is welcome to join the team. In all these three months I enjoyed everything from the hours-long research discussions at the whiteboard, to the entertaining lunches and dinners, from the technical seminars to the karaoke sessions. ▁

06 Lucas Brutschy

M.S. Student, RWTH Aachen University, Germany
27/09/2009 ~ 31/03/2010

The first time I thought about going to Korea was when I was listening to a talk by Prof. Yi in Aachen, Germany. At that time I was working as a student intern on State Space Abstractions for a Model Checker based on Abstract Interpretation, and was very interested in Sparrow, a tool for finding Bugs in C Programs developed at the group of Prof. Yi. When I asked him, if I could come to Korea for 6 month he immediately said yes.
I have been in Korea for the last three months now, and I am both personally and professionally very satisfied with my life here. Since accommodation and many other organizational problems were handled by the admin staff, the only problem for me left to solve was getting an Alien Registration Card, which turned out to be a time-consuming but rather easy-to-handle problem.
With the help of my colleagues it was very easy for me to get integrated in the current research work at the ROPAS laboratory and after a few days I was already working on an interesting problem together with my colleague Hakjoo Oh.

The last three months have been very interesting for me. In our weekly Show&Tell sessions I learned about the different research projects in my group and I am impressed by many of them. My own work also turned out to be quite successful, due to the great teamwork with my colleagues and the advise of Prof. Yi. I not only learned a lot about static analysis but also about Korea. I learned to love the live in Korea, which is not always easy for a foreigner, mostly due to the language barrier. As long as you stay on the campus, everything is fine, but for example getting a haircut in the city can be an adventure. But as I found out, most Koreans are very helpful and all problems were solved very fast. Now that I know some Korean words and the way things go in Korea, life is very easy and enjoyable.

The last three month were a very good experience for me, I learned a lot and we were quite productive, and I hope the remaining three months will be equally satisfactory. —

참여교수 연락처

제1총괄과제

1-1세부과제	교수 백윤흥	02-880-1748,	ypaek@ee.snu.ac.kr	서울대학교 전기컴퓨터공학부
1-2세부과제	교수 이광근	02-880-1857,	kwang@ropas.snu.ac.kr	서울대학교 컴퓨터공학부
	교수 우치수	02-880-6573,	wuchisu@snu.ac.kr	서울대학교 컴퓨터공학부
1-3세부과제	교수 도경구	031-400-5667,	doh@hanyang.ac.kr	한양대학교 컴퓨터공학과
	조교수 류석영	042-350-3538,	sryu@cs.kaist.ac.kr	한국과학기술원 전산학과

제2총괄과제

2-1세부과제	조교수 이욱세	031-400-5234,	oukseh@hanyang.ac.kr	한양대학교 컴퓨터공학과
2-2세부과제	교수 문병로	02-880-8793,	moon@snu.ac.kr	서울대학교 컴퓨터공학부
	조교수 정교민	042-350-3544,	kyomin@kaist.edu	한국과학기술원 전산학과
2-3세부과제	교수 박종우	02-880-7133,	fcp@snu.ac.kr	서울대학교 기계항공공학부
	교수 김유단	02-880-7398,	ydkim@snu.ac.kr	서울대학교 기계항공공학부
2-4세부과제	교수 최광무	042-350-3520,	choe@kaist.ac.kr	한국과학기술원 전산학과
	교수 황규영	042-350-3522,	kywhang@cs.kaist.ac.kr	한국과학기술원 전산학과

제3총괄과제

3-1세부과제	조교수 박성우	054-279-2386,	gla@postech.ac.kr	포항공과대학교 컴퓨터공학과
3-2세부과제	조교수 김문주	042-350-3543,	moonzoo@cs.kaist.ac.kr	한국과학기술원 전산학과
	조교수 최윤자	053-950-7549,	yuchoi76@knu.ac.kr	경북대학교 전자전기컴퓨터학부
	교수 최진영	02-3290-3200,	choi@formal.korea.ac.kr	고려대학교 컴퓨터통신공학부
3-3세부과제	조교수 황승원	054-279-2385,	swhwang@postech.ac.kr	포항공과대학교 컴퓨터공학과
	부교수 한욱신	053-950-7572,	wshan@knu.ac.kr	경북대학교 컴퓨터공학과

위탁과제

위탁과제1	부교수 안준선	02-300-0144,	jsahn@kau.ac.kr	한국항공대학교 정보통신공학부
위탁과제2	부교수 이승용	054-279-2245,	leesy@postech.ac.kr	포항공과대학교 컴퓨터공학과

오시는길

대중교통 이용
낙성대역 (4번 출구) → 마을버스(2번) → 후문 통과 → 302동 정류장 하차
서울대입구역 (3번 출구) → 버스(5511, 5513) → 정문 통과 → 302동 정류장 하차

발행일 | 2010년 5월
발행인 | 이광근
편집인 | 이민수
발행처 | ROSAEC CENTER
서울시 관악구 관악로 599 서울대학교 138동 202호
tel 02-880-7290
fax 02-882-7234
mail rosaec@snu.ac.kr
web rosaec.snu.ac.kr

ROSAEC center

Research On Software Analysis for Error-free Computing

소프트웨어 무결점 연구센터 NRF ERC

151-742 서울시 관악구 관악로 599 서울대학교 138동 202호
Rm 202 Bldg 138, Seoul National University, 599 Gwanak-ro Gwanak-gu, Seoul 151-742, KOREA
Tel +82 2 880 7290 Email rosaec@snu.ac.kr
Fax +82 2 882 7234 Web rosaec.snu.ac.kr