TRIP REPORT

POPL/VMCAI 2011

Austin, Texas, USA

오학주 pronto@ropas.snu.ac.kr
서울대학교 프로그래밍 연구실

개요

2011.01.22-30 일정으로 미국 텍사스 오스틴에서 열린 POPL/VMCAI 2011에 참석했습니다. POPL (Principles of Programming Languages)은 프로그래밍 언어 전 분야의 주제들을 다루고 있는 우리분야의 대표적인 학회입니다. VMCAI(Verification, Model Checking, Abstract Interpretation)는 프로그램 분석/검증의 주요 세가지 분야의 협업을 도모하는, 프로그래밍 언어 분야내에서도 분석과 검증에 집중하는 학회입니다.

올해에는 한국인들의 논문이 많이 발표되었습니다. 작년에는 한국논문이 VMCAI에 1편이 발표되었던 반면, 이번에는 모두 3편이 발표되었습니다. 우리 연구실의 최원태 학생이 POPL에, KAIST의 김세원 박사님과 제가 VMCAI에 한편씩 발표했습니다. 외국에 몸담고 계시는 한국분들의 논문까지 합하면 모두 5편입니다. 영국 퀸메리대학교의 양홍석 박사님, 독일 막스플랑크 연구소의 허충길 박사님의 논문이 POPL에 한편씩 있었습니다.

개인적으로는 작년에 이어 두번째로 참가해서인지, 좀 더 학회분위기에 익숙해 졌던 시간이었습니다. 이 글에서는, 논문발표 등 학회기간중 있었던 일을 흥미로웠던 논문들 소개와 함께 전달해 보도록 하겠습니다.

VMCAI

Austin 도착 다음날인 1/23부터 VMCAI가 시작되었습니다. 프로그램위원장의 간단한 환영 인사가 있었고 바로 Francesco Logozzo가 마이크로소프트에서 개발중인 정적분석툴을 소개하는 강의로 학회가 시작되었습니다. Patric/Radhia Cousot 교수님들과 함께 작업을 해서인지 툴 이름이 Clousot 인게 재밌었습니다. 이 툴을 기반으로 한 논문이 이번 VMCAI와 POPL에 한편씩 발표되었습니다.

제 발표는 첫째날 마지막 세션에 있었습니다. 그래서 사실 이날은 초청발표 빼고 다른 논문 발표들은 귀에 잘 들어오지 않았던것 같습니다. 첫째날에 발표는 둘째날 부터는 마음이 편해진다는 장점이 있지만 도착직후 발표를 해야하는 부담도 있었습니다. 특히, 제 경우는 출발 직전에 슬라이드가 많이 바뀌는 바람에 긴장이 더 되었습니다.

이번에 발표한 논문은 정적분석기의 성능향상을 위해 새로운 필요한 부위만 가지고 분석하는 방법(state localization)에 대한 내용입니다. 사실 그 동안 랩 세미나 등등에서 여러번 발표해 보았던 내용이어서 준비가 쉬울줄 알았는데 막상 해보니 또 그렇지가 않았습니다. 두가지 내용을 전달하고 싶었습니다.

(1) 문제와 해결책을 쉽게 전달하기: 제 경우에 학회에서 논문발표를 들을때 가장 어려웠던 점은 해당 분야에 친숙하지 않은 상태에서는 풀고자 하는 문제가 무엇인지조차 감을 잡기가 어렵다는 것이었습니다. 문제도 이해되지 않으니 발표내내 멍하니 피상적으로 듣게 됩니다. 그래서 문제의 배경과, 해결하려는 문제가 무엇인지에 대해 설명하는것에 신경을 많이 썼습니다. 문제를 잘 이해하면 해결방법도 자연스럽게 잘 이해가 될 것이라 생각했기 때문입니다. 이를 어느정도로 얘기해야하나는 많은 고민이 되었습니다. 해당 분야 또는 인접 분야 전문가들이 모여있는 곳이기 때문에 너무 당연한 내용을 설명해도 적절치는 않을 것이기 때문입니다. 어쨋든 우리분야(요약해석)에 친숙하지 않은 사람들에게도 무엇을 어떻게 하겠다는 것인지가 잘 전달되도록 슬라이드를 구성하고 싶었습니다.

(2) 동시에, 문제와 해결책이 가지는 실제적인 임팩트를 강조하고 싶었습니다. 머리속에서는 간단한 아이디어라도 그것을 "실재"에 적용해보고자 할때는 생각지도 못했던 변수들이 생기게 마련입니다. 이런 실제적 디테일들에 대한 내용은 무시하고 내용을 너무 이상적이고 단순하게만 구성하면 자칫 문제와 해결책이 뻔해보일수가 있는데 이것도 피하고 싶었습니다. 그래서 장난감 프로그램을 분석할 때 일어나는 문제와 해결이 아니라, 실제 프로그램을 분석에서 비로소 드러나는 문제이며 따라서 그 해결이 실제 프로그램 분석에 큰 영향을 가짐을 보여주고 싶었습니다.

출장 바로전 랩세미나에서 가진 리허설 때 이 두가지를 염두에 두고 준비를 했었는데, 예상했던 준비기간보다 더 오래걸리는 바람에 이 둘을 자연스럽게 섞지 못한상태에서 발표를 했고 결과적으로 이도저도 아닌 경우가 되어버렸습니다. 이후 많은 부분들을 고쳐야 했습니다. 출발 직후까지 슬라이드를 고쳤고 비행기에서 그리고 도착후 발표전까지는 주로 영어발표 연습을 했습니다.

역시나 이번 발표에서도 영어가 큰 부담가운데 하나였습니다. 영어라는게 오히려 별 신경을 쓰지 않으면 떠듬떠듬하긴해도 큰 문제를 일으키는 것 같지는 않은데, 영어자체에 신경을 많이 써 버리면 더 큰 문제를 일으키는 것 같습니다. 영어에 신경쓰다 오히려 해야할 말을 못하게 되는 상황이 되어버립니다. 그래도 학회에서는 평소처럼 버벅거릴수는 없었기에 시나리오를 고정하고 그것은 다 말을 할 수 있도록 연습했습니다.

결과적으로 내용자체는 비교적 쉽게 전달했던것 같은데 임팩트를 강조하는 부분이 부족했던것 같습니다. 그래서 발표가 끝나고 나서 후련하면서도 뭔가 아쉬움이 남았습니다. 쉽고도

강렬한 인상을 주려면 더 철저하게 준비를 해야함을 깨달았습니다. 하지만 어쨌됐든 발표를 끝내니 둘째날부터는 한결 편안히 학회를 즐길수 있었습니다.

흥미로웠던 논문들

Refinement-Based CFG Reconstruction from Unstructured Programs

실해파일분석의 중요한 이슈가 실행흐름그래프 재구성 (CFG reconstruction) 문제입니다. 실행파일을 분석하기 위해서도 프로그램의 실행흐름그래프가 필요합니다. 하지만 소스파일로부터 그래프를 만들때와는 다른 어려움이 있습니다. 주된 이슈는 동적점프(dynamic jump)인데 문제는 실행파일에는 이것이 많다는 것입니다. 따라서 동적점프를 잘 처리하지 않으면 소스파일로부터 그래프를 만들때와는 달리 부정확한 그래프를 만들게 되고 따라서 같은 분석기법을 적용하더라도 더 부정확한 결과를 얻게 됩니다. 또 만에하나 잘못된 그래프를 만들게 되면 안전하지 않은 분석결과를 얻게 됩니다. 이 논문이 주목한 것은 동적점프를 결정하는 정보는 아주 일부에서 비롯되며 이 정보의 작은 정확도 차이가 생성된 그래프의 정확도를 크게 좌지우지 한다는 사실입니다. 이러한 특성은 자연스럽게 refinemento-based approach의 적용을 떠올리게 하고, 결국 이 방법을 그래프 재구성 문제에 적용해 본것이 논문의 주 내용입니다. 즉, 전체 정확도 향상에 꼭 필요한 부분은 아주 일부정보이기에, 이것을 처음부터 전체분석할 필요없이 점짐적으로 국소지역의 정확도를 높여가는 방법을 사용하자는 것입니다. 문제가 명확했고 그에 따른 적절하고도 자연스런 해결책을 도입한 좋은 논문이라고 생각합니다.

Automata Learning with Automated Alphabet Abstraction Refinement

실제로 사용되는 시스템들중 상당수는 스펙이 자세히 주어지지 않기 마련입니다. 예를들어, 논문의 저자들이 통신회사의 프로토콜 규약들을 살펴본결과 자연언어로만 기술되어 있을뿐 명확히 기술된 스펙을 찾기가 힘들었다고 합니다. 때문에 이들 시스템을 모델체킹등을 이용하여 엄밀히 검증하기가 어렵게 됩니다. 검증을 하려면 엄밀한 스펙이 필요한데 그것이 없는 상황입니다. 이 논문은 active learning의 일종인 automata learning을 이용해서 스펙이 구체적으로 명시되지 않은 시스템의 행동양식(behavior)를 추론하는 방법에 대한 것이었습니다. 세세한 내용은 잘 이해되지 않았지만, 우리랩에서 진행했던 algorithmic learning과 유사하게 두가지 쿼리에 답해가면서 학습해가는 구조를 보니 친숙함을 느꼈습니다. 그리고 헐렁한 요약부터 시작해서 점차 자세한 단계로 진행하는 abstraction refinement를 활용합니다. 무엇보다 문제 자체가 재미있었던 발표였습니다.

Precondition Inference from Intermittent Assertions and Application to Contracts on Collections

소스코드내의 사용자의 assertion으로부터 함수의 실행전상태(pre-condition, 만족하지 않으면 assertion이 만족하지 않도록 하는)를 유도하는 문제에 대한 논문이었습니다. 해결책 보다는 문제 자체를 엄밀하게 정의하는데 더 초점을 둔 것 같았습니다. 그리고 그 문제에 대해 요약해석틀 내에서 해결책을 두어가지 제시하였습니다. Patrick Cousot교수님께서 직접 발표하셨는데, 수식으로 가득한 슬라이드를 술술 말로 풀어 설명하시는 모습에서 대가의 면모를 엿볼 수 있었습니다. 하지만 역시 너무 세부적인 내용은 이해하지 못해습니다. 참고로 Cousot교수님은 VMCAI의 거의 모든 발표에 질문을 하셨습니다. 가장 앞자리에 앉으셔서

발표들을 주의깊게 들으신 후 분야를 넘나들며 핵심적인 내용을 질문하시는 모습이 인상적이었습니다.

Sets with Cardinality Constraints in Satisfiability Modulo Theories

POPL/VMCAI를 통틀어서 가장 발표를 깔끔하게 한 논문가운데 하나라고 생각합니다. 문제의 동기는 집합이 프로그램에서 많이 사용되는 개념인데 (특히 ML같은 언어에서) SMT solver가 집합 (특히 집합의 크기와 관련된) 연산을 지원하지 않는다는 것입니다. SMT solver인 Z_3에 이를 지원하도록 확장한 연구입니다. 집합연산을 효율적으로 다루기 위해서 문제를 쪼개어서 푸는 알고리즘을 고안했습니다. 문제와 해결책을 직관적인 그림들로 잘 표현했고, 설명도 막힘없이 깔끔하게 진행했습니다. 참고로 이 논문을 쓴 Victor Kuncak 그룹은 모델체킹등에서 잘 활용할 수 있도록 SMT를 확장하는 일을 하고 있다고 합니다. 작년 VMCAI에서 초청발표를 했었는데, 앞으로의 SMT확장계획에 대해서 장황하게 설명했는데 이 논문도 그 가운데 하나였습니다.

Static analysis papers

정적분석 분야에서는 총 3편의 논문이 발표되었습니다. 첫번째로 Eric Goubault가 또 플로팅포인트 연산에 대한 논문을 썼습니다. 이번 논문에서는 그동안 구현에서 쓰던 플로팅포인트 연산을 위한 요약공간 두 개를 정리했습니다. 하나는 변수들간의 관계를 무시하는 공간이고, 다른 것은 관계를 고려하는 공간입니다. 중요한 분야이고 관련분야를 연구하는 사람들에게 도움이 될만한 논문일 듯 싶습니다. 하지만 발표자체는 Eric Goubault의 학생이 했는데 너무 디테일에 치중하여 잘 전달이 되지 않았습니다. 두번째와 세번째 논문은 스트링 분석에 대한 것이었습니다. 특히 세번째는 KAIST 김세원 박사님 논문이었는데 요약해석 기반으로 문자열 분석을 디자인 한 예입니다. 요약공간을 PDA기반으로 디자인 하였는데 문자열을 직접 요약한 공간을 다룬다는 점, 그리고 요약해석에 자연스럽게 올라탄 스트링 분석이어서 기존의 기술들이 별다른 노력없이 적용되어 시너지효과를 낼 수 있다는 점이 강점인것 같았습니다. 전체적으로 상당히 이론적인 내용이라 듣는 사람 입장에서는 어려움이 있었겠지만 역시 Cousot 교수님은 관심있게 이것저것 질문을 하셨습니다.

POPL

작년과 달리 이번 POPL은 두 개의 세션이 나란히 진행되었습니다. 두 방에서 따로 진행되므로 한번에 하나의 세션만 참가할 수 있었는데, 다행히 나란히 진행되는 세션들은 큰 관련이 없는 분야들로 묶어져있어서 듣고 싶은 발표는 거의 다 들을 수 있었습니다.

세션 중간중간 휴식시간, 점심시간등의 분위기는 작년과 비슷했습니다. 활발히 다른사람들과 교류하는 모습은 여전히 인상적이었고 재미있었습니다. 학회에 처음갔을때는 처음 보는 사람에게 말을 거는게 어색해서 쉬는시간이 길게 느껴졌는데, 이번에는 예전보다 눈에 익은 사람들이 몇몇 보이고 학회분위기에 좀더 익숙해지니까 쉬는시간이 재미있었습니다. 다른 사람에게 말을 걸고 발표자에게 궁금한 것을 물어보는것도 예전보다 더 자연스럽게 느껴졌습니다.

또한, 학계가 생각보다 좁음을 느꼈습니다. 학회에서 우연히 고등학교 선배님을 만났는데, 제가 지금 연구하고 있는 주제와 아주 관련이 많은 주제를 연구하는 사람과 대학원 시절 옆방 친구셨다는 것을 알았습니다. 이번 POPL에 왔길래 인사라도 하고 이것저것 물어보려고 했지만 먼저 갔는지 찾을수는 없었습니다. 또한 교수님과 Microsoft India의 G. Ramalingam의 대화에 우연히 참여하게 되었는데 한국에 돌아와서 보니 그 분이 지금읽고 있는 논문의 저자였습니다. 학회 전에 논문을 읽었으면 하는 아쉬운 마음이 들었습니다.

흥미로웠던 논문들

Points-to analysis with efficient strong update

발표전에 잠깐 초록을 읽어보았는데 내용이 진부해보였습니다. 하지만 발표를 듣고 논문을 읽어보니 왜 POPL에 실렸는지를 알게되었습니다. 내용자체는 참 간단한 것이었습니다. 실행흐름을 구분하는(flow-sensitive) 분석이 가지는 가장 큰 이점은 강한갱신(strong update)을 할 수 있다는 것이고 실행흐름을 구분하지 않는(flow-insensitive) 분석은 빠른것이 강점입니다. 이 둘을 효과적으로 결합하여 새로운 방식의 포인터 분석 알고리즘을 만들었습니다. 기본적으로 흐름을 구분하지 않으면서 "강한갱신이 가능한" 변수들에 대해서만 흐름을 구분합니다. 강한갱신이 가능한 조건은 변수가 가리키는 포인터 집합의 크기가 1일 때입니다.

그 동안 포인터 분석의 주류는 흐름을 구분하지 않는 경우였는데, 최근들어 흐름을 구분하는 분석의 성능향상에 대한 논문이 많이 나오고 있습니다. 이 논문도 그 중 하나의 부류라고 볼 수 있습니다. 제가 연구하는 주 분야가 흐름을 구분하는 요약해석기의 성능향상이고 최근의 포인터 분석분야에서 사용하는 기술들과 유사점이 많기때문에 이 분야 최신논문들을 주의깊게 살펴보던 중이었는데 이 논문을 POPL에서 보게되어 반가웠습니다. 하지만 엄밀하게 말해서 이 논문은 흐름을 완전히 구분하는 것은 아니고, 우리의 경우는 그것이 필요한 경우이므로 직접적인 도움은 되지 않을것 같았습니다. 하지만 흐름을 구분하지않은 분석과 거의 유사한 비용으로 흐름구분분석과 거의 유사한 정확도를 보이는 실험결과는 충분히 인상적이었습니다. POPL마지막날 이 논문의 발표자(Ondrej Lhotak)와 얘기할 기회가 있었는데, 이름도 없는 제 이야기를 주의깊게 들어주어서 고마웠고, 또 VMCAI에 비슷한 분야의 논문을 발표했다고 하니까 꼭 읽어보겠다고 했습니다.

Learning minimal abstraction

문제 자체가 흥미로운 논문입니다. 프로그램 분석은 보통 프로그램의 어떤 성질을 증명하기 위해서 사용됩니다. 그런데 분석의 정확도에 따라 이 성질이 성립하는지 알수 있거나, 성립하는지 안하는지 알수없거나가 결정됩니다. 보통의 분석들은 정확도를 조절하는 메커니즘(예를 들어 함수컨텍스트를 k개까지 구분한다던지하는, kCFA)을 가지게 마련인데, 문제는 이 메커니즘이 천편일률적으로 분석에 적용되어 때로는 과도한 정확도를 가지게 된다는 것입니다. 예를 들어서 어떤 성질을 증명할 때 함수 f의 컨텍스트는 구별할 필요가 있지만 g는 굳이 구분할 필요가 없을수 있습니다. 하지만 만약 1CFA로 분석을 하면 f와 g모두 구분하게 됩니다. 이 경우 원하는 성질을 증명할 수 있겠지만 불필요하게 비용을 지불하는 셈이 됩니다.

이 논문은 어떤 성질을 증명하기 위해서 필요한 최소의 요약상태가 무엇인지를 알아내는 알방법에 대한 것입니다. 다시 예를들어, f는 컨텍스트를 구분하고 g는 구분하지 않아도 된다는 사실을 알아내는 것입니다. 그리고 알아낸 요약이 딱 알맞은 요약입니다. 이를 최소의 요약(minimal abstraction)이라고 부릅니다. 문제 자체도 흥미롭지만 결과도 인상적입니다. kCFA에 대해서 최소의 요약을 구해보니 98%정도가 해당 성질을 증명하는데 불필요한 군더더기였습니다. 즉, 분석기가 평소에 안해도 되는 일을 너무 많이 하고 있다는 것을 의미합니다. 사실 이같은 결과는 어느정도 잘 알려진 사실이고, 실제로 우리 연구실에서도 이와같은 현상을 자주 관찰해 왔기에 딱히 새롭다할만것은 아니었지만, 한편으론 아무도 직접 해보지는 않았던 것을 구체적인 수치로 보여주니 멋지다는 말밖에 할수 없었습니다.

하지만 방법 자체의 실용성은 아직 생각하지 않는 듯 했습니다. 왜냐하면 이를 위해서는 해당 분석을 여러번 해봐야 하기 때문입니다. 나중에 발표자에게도 이를 물어보았는데, 아직 본인들도 어떻게 쓸지에 대한 생각은 없고 이를 해결해서 실제로 최소의 요약을 분석전에 알아내는것이 앞으로 할 일중 하나라고 답했습니다.

A parametric segmentation functor for fully automatic and scalable array content analysis

Cousot교수님의 POPL논문입니다. 발표는 Francesco Logozzo가 했습니다. 분석에서 배열(혹은 버퍼)을 요약할 때 자주 쓰는 방식은 크게 두 가지 정도입니다. 하나는 배열의 모든 원소를 구분하지 않고 하나의 셀로 요약하는 방법, 두번째는 k개까지 구분하는 방식입니다. Sparrow도 이 두가지 방식을 모두 지원하는데 둘다 성능이 불만족스럽긴 마찬가지입니다. 하나로 요약하면 정확도가 너무 낮습니다. k개까지 구분할때는 k의 값에 따라 정확도가 너무 떨어지거나 비용이 너무 비싸지기 일쑤입니다.

이 논문은 이러한 배열구분분석을 정확도도 유지하고 비용도 많이 들이지 않으면서 하는 방법에 대한 것입니다. 발표전에 우석이한테 이 논문의 결과에 대해서 들었었는데 비용이 배열을 모두 뭉칠때보다 1-2%정도밖에 증가하지 않으면서도 정확도는 상당히 높아진다는 얘기를 듣고 그 방법이 매우 궁금했습니다. 또 우리가 직접 겪고있던 문제라 더욱 관심이 갔습니다. 방법은 생각보다 간단했는데, 분석중에 구분할 필요가 생길때 구분하고, 필요가 없을 때 합치는 방법입니다. 모양분석에서 요약주소를 출생지기반으로 미리 고정시키지않고 분석중에 동적으로 만들어내는 방법과 유사합니다. 우리도 배열구분문제에 대해서 좀 더 제대로 고민해보았더라면 비슷한 해결책을 낼 수 있었을지도 모른다는 생각이 들었습니다.

Automatig string processing in spreadsheets using input-output examples

이번 POPL에서 가장 실용적 내용의 논문인듯 합니다. 마이크로소프트 엑셀 사용자의 커뮤니티에서 가장 많이 제기되는 불편함을 조사해보니, 사용자가 입력해놓은 자료들을 다른 포맷으로 변경하려고 할 때 이를 자동으로 어떻게 할수 있는가에 대한 것이었다고 합니다. 엑셀에서 물론 매크로나 비주얼베이직 등을 이용한 프로그래밍환경을 지원하지만 일반 사용자가 이를 이용하는 것은 거의 불가능합니다. 따라서 이를 자동으로 해 주고 싶은것이 이 연구의 동기입니다. 사용자가 원래자료와 변형된 자료의 예를 몇개 보여주면 시스템이 이를 학습하여 다른 자료들에 대해서도 변형을 시도합니다. 하지만 몇가지 예제를 통해 일반적인 방법을 유추할 수는 없으므로 사용자는 시스템이 혹시 잘못 변형한 부분에 대해서 몇가지 수정을 해 줍니다. 그러면 시스템은 사용자의 수정을 반영하여 다음부터는 그러한 실수를 하지

않습니다. 일종의 프로그램 합성(program synthesis)를 실제 개발현장에 응용한 것인데, 이 기능이 엑셀의 최신버전에 포함된다고 합니다.

맺음말

학회외에도 즐거운 일들이 많았습니다. 작년에 우리랩에 방문했던 독일 Aachen 대학교의 Lucas Brutschy를 다시 만날 수 있어서 반가웠습니다. 저녁마다 축제분위기인 오스틴의 거리도 인상적이었습니다. 겨울인데도 반팔을 입은 사람들이 많을만큼 음악과 젊음으로 가득한 도시 같았습니다. 오스틴 대학교 근처에서 먹은 피자를 비롯하여 텍사스-멕시칸 음식들(Tex-Mex)도 맛있었습니다. 특히 마지막날 저녁에 먹은 두께가 4cm는 됨직한 텍사스 스테이크는 잊지 못할 것 같습니다.