

Trip Report - Static Analysis Symposium (SAS) 2014



2014년 9월 9일 - 16일

뮌헨, 독일

서울대학교 이우석

1. 개요

이번 SAS는 몇개의 위성 워크샵들과 함께 9월 10일부터 9월 13일까지 독일 뮌헨에서 열렸다.

SAS는 규모는 작지만 단단한 학회라고 생각한다. SAS는 정적분석 분야 대표 학회로 유명하고 역사도 있지만, 작다. 올해 20편 남짓의 논문이 발표되었고 참가자는 대략 80명쯤 된 것 같다. 아마 다른 분야 같으면 워크샵 규모일 것이다. 하지만 위원회에 늘 이 분야 대가들이 포진해있고, rebuttal 과 artifact 제출 등 최고급 학회들과 동일한 절차로 운영된다. SAS에 참여한건 이번이 처음이었는데, 발표된 논문들도 초청 발표들도 재미있는 것들이 많았다.

나는 이번에 분석기의 진행율을 예측하는 연구 (A Progress Bar for Static Analyzers)로 논문게재를 승인받고 발표할 기회를 얻게 되었는데, 재미있는 경험으로 남았다. 주저자로서 학회에 논문을 발표한 것은 이번이 두 번째이다. 예전에 VMCAI라는 학회에서 발표할 때는 잘 즐기지 못했다. 발표 끝나고 영어 질문을 못 알아들으면 어쩌나, 영어 발표가 매끄럽지 못하면 어쩌나, 예상되는 질문들에 대해서 어떻게 방어를 해야 하나... 하는 걱정들을 많이했고, 발표를 방어적으로 임한 것이 결국 소통기회를 스스로 차단한 것이 아니었나 싶었다. 이번엔 영어에 관해서는 그냥 마음을 놔버렸고, 지적이나 코멘트에 방어적으로 임할 필요가 없다고 생각했다. 관심 자체가 매우 고마운 것이기 때문이다. 고마운 관심에 힘입어 토론을 통해 이 일의 의미, 문제점, 보완점을 같이 모색해보면 그보다 좋은일이 없다고 생각했다. 학회는 각자 알아낸 사실들을 가지고 소통하는 자리이지 업적 자랑 및 방어의 장이 아니라는 생각을 혼자하면서, 편하게 했더니 오히려 수월하게 된 것 같다.

2. 발표된 논문들

A Progress Bar for Static Analyzers

Woosuk Lee, Hakjoo Oh, and Kwangkeun Yi

정적 분석기에 프로그래스 바를 구현하는 방법에 관한 우리 논문이다. 복잡한 프로그램을 분석하는데 시간은 오래걸리는데, 사용자는 아무런 진척사항을 알 수 없다는 문제의식에서 출발했다. 방법은 분석결과가 갖는 래티스 높이의 비율을 이용하는 것이다. 분석에서 고정점은 밑바닥 원소(bottom)부터 시작해서 고정점에 도달할 때 까지 요약함수를 적용하는 방식으로 계산된다. 이 때 요약함수는 단조 혹은 팽창함수이기 때문에 래티스 높이가 시간이 지남에 따라 증가한다. 만약 최종결과의 래티스 높이를 알고 있다면, 높이의 비율로 프로그래스 바를 만들 수 있다. 우리는 최종결과의 래티스 높이를 정확히 파악하기 위해 기계학습을 이용하였고, 일정하게 증가하지 않는 래티스 높이를 일정하게 증가하는 모양으로 보정하기 위해 본분석보다 부정확하지만 저렴한 전분석을 사용하였다.

발표가 진행됨에 따라 사람들이 점점 흥미있어하는 것이 느껴졌다. 특히 전분석을 이용한 부분에서 신기해하는 듯 했고, 발표 후 질문시간에 머신러닝에 관심을 보이는 사람들도 있었다. 내용에 대해서 사람들이 어떻게 생각할지는 모르겠지만 일단 관심을 끄는 데는 어느정도 성공하지 않았나 싶다.

Program Info.

Line	# Variables	# Functions	# Blocks	# Commands
20K	2706	221	8736	11717

Steps

Step 1. Parsing *.i files

Step 2. Building control flow graph

Step 3. Building def-use graph

Step 4. Pre-analysis for progress estimation

Step 5. Main analysis

Step 6. Analysis report generation

Main Analysis Progress

40%

원래 데모를 보여주고 주의를 환기시키려고 위와 같이 UI를 간단하게 만들었는데 학회장 인터넷이 너무 느려서 보일 수 없었다. 분석의 어떤 단계가 진행 중이고, 프로그래스 바가 어떻게 차오르는지 보여줘서, 전분석이 빨리 끝나고 프로그래스 바의 길이 어느정도인지 파악할 수 있게하는게 목적이었는데 아쉬웠다. 인터넷이 필요한 데모를 보여줄 때는 발표장 인터넷이 느릴 경우도 생각해서 미리 동영상 녹화를 해놓는 것도 좋겠다는 생각이 들었다.

데모를 못하는 바람에 계획한 발표가 25분이 아닌 20분만에 끝났고, 남은 10분동안 질문을 계속 받았다. 혹시라도 궁금하실 분들을 위해서 기억나는 질문과 코멘트들, 그에 대한 나의 답변을 아래에 써보았다.

Q1 : 매 고정점 계산 이터레이션마다 동일한 계산시간이 필요할 거라는 가정이 실제로 정말 맞는 가정인가? 실험해보았는지? 다각형같은 보다 복잡한 도메인을 사용할 경우도 이 가정이 맞는지 살펴봐야 할 것 같다.

A : 실험을 해보지는 못했지만 우리는 경험적으로 어느정도 맞다고 생각하고 진행했다. 다각형 도메인의 경우는 잘 모르겠으나 템플릿 도메인(Template Constraints Matrix)분석이나 육타곤 분석은 우리의 가정이 맞을 것이라고 생각된다.

Q2 : 간단한 와이드닝만 썼는데, 보다 복잡한 와이드닝을 사용할 경우 어떻게 될지 살펴봐야 할 것 같다.

A : 복잡한 와이드닝을 쓸 경우 양상이 달라질 수 있다는데 동의하고, 그에 대한 조사가 필요하다고 생각한다. (와이드닝에 대해서 발표 때 자세히 언급하지 않았는데, 논문 리뷰와 동일한 와이드닝 질문이었다. 사람들에게 이 일의 한계를 정확히 언급하라는 의도 같았고, 덕분에 리뷰어가 누구였는지 알 수 있었다)

Q3 : 최종 래티스 높이 예측을 선형회귀로 했는데 각 피쳐마다 갖는 가중치의 값이 어떤가? 어떤 피쳐가 크고 어떤 피쳐들이 작은가?

A : post fixpoint 성질과 관련된 피쳐들이 가중치가 크고, 나머지는 골고루 작다.

Q4 : 혹시 전분석은 비교적 빨리 끝났는데, 본분석은 (제한시간 안에) 끝나지 않는 경우가 있는가?

A : 흥미로운 경우 같은데, 아직 그런경우는 보지 못했다.

Q5 : 처음 분석할 때는 너의 방법으로 프로그래스 예측을 하되, 그 다음에 그 프로그램을 분석할 때는 이미 알고있는 고정점 이터레이션 수로부터 프로그래스 바를 만들면 되지 않을까? 왜냐하면 시간이 오래걸리는 큰 프로그램들 경우, 한번에 많이 고치는 일이 잘 없기 때문에 분석에 필요한 고정점 이터레이션 수가 거의 비슷하게 유지될 것이기 때문이다.

A : 실제 사용할 때는 그렇게 하는 것도 좋겠다. 좋은 생각이다.

Sawjacard: A Static Analysis Tool for Certifying Java Card Applications

Frederic Besson, Thomas Jensen, and Pierre Vittel

이 논문은 자바 카드라는 언어로 쓰인 스마트카드 어플리케이션을 위한 체커에 대한 것이다. 자바 카드(자바를 스마트 카드 프로그래밍 용도로 좀 변형시킨 언어) 프로그램은 몇 가지 특성이 있는데, 라이브러리가 전부 네이티브 코드로 주

어지고, 메모리 제약이 심해서 동적할당이 제한되며, 보안이 강조되어 어플리케이션 간 통신이 뻣뻣한 것들이 그것이다.

저자들은 NFC 분야의 한 협회에서 만든 스마트카드 프로그래밍 가이드라인이 자바 카드 프로그램에서 지켜지는지 검사하는 체커를 만들었다. 많은 라이브러리들을 직접 적절히 인코딩했다. 체커는 메모리 할당이 허용된 형태로 이루어지는지, 재귀호출은 없는지, 어플릿 객체가 static 필드로 접근될 수 있는지 등 약 60여가지의 규칙을 체크한다.

기술적으로는 대단한 것이 없지만 응용을 잘 찾은 예라고 생각된다. 분석기에는 익히 잘 알려진 인터벌, 포인터 분석과 부분적으로 경로민감한 분석, 아주 약한 수준의 관계분석이 결합되었다. 간단한 것들을 잘 결합해서 특정 프로그램 군에 대해서 요긴하게 쓰일 수 있는 체커를 만든점이 높이 살만하다고 생각했다.

Fully Automated Shape Analysis Based on Forest Automata with Data Constraints

Tomas Vojner (Brno Univ. of Technology, Czech Republic)

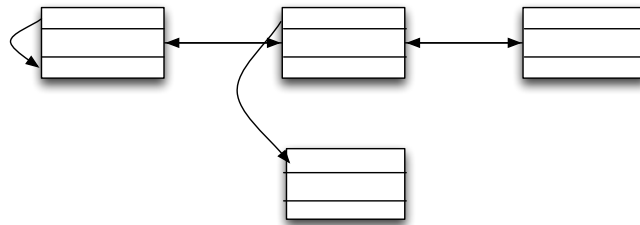
이 초청발표는 포레스트 오토마타를 이용한 새로운 모양분석에 관한 내용이다.

모양분석은 힙 자료구조 형태를 파악하는 분석이다. 모양분석은 널 포인터 접근 탐지, 힙 쓰레기의 부재 혹은 힙 자료구조 특성을 증명하는 목적으로 수행된다. 널리 알려진 모양분석 방법으로는 3-value logic 기반 방법, 그리고 분리논리를 이용한 방법이 있다. 그것들과 별개로 발표 연사의 연구팀은 새로운 방식의 모양분석을 연구해왔다고 한다. 발표에 따르면, 이 방법은 기존의 오토마타 기반 요약의 이점(일반성, 요약의 유연함)과 분리논리 기반방식의 이점(효율성)을 모두 이용할 수 있다고 한다.

이 방식에서 힙 자료구조는 포레스트 오토마타로 표현된다. 포레스트 오토마타란 트리오토마타의 튜플이다. 힙은 몇 개의 트리로 나누어서 표현되고, 각 트리

의 루트노드는 컷 포인트라고 불리는데, 어떤 포인터 변수들이 가리키는 노드거나 혹은 몇 개의 들어오는 간선(incoming edge)들을 갖는 노드이다. 각 트리의 말단 노드는 자신이 속한, 혹은 다른 트리의 루트노드를 가리킬 수 있다. 이렇게 함으로써 다양한 형태의(양방향 순환 리스트 같은 것 포함) 자료구조를 표현할 수 있다.

이 방식의 이점은 오토마타기반 요약의 유연함과 지역성(local heap manipulation)을 동시에 취할 수 있다는데 있다. 분리논리 기반 방식은 힙을 서로 독립적인 여러개의 영역으로 분리해서, 한 영역에서 일어나는 일을 다른 영역을 고려할 필요없이 분석함으로써 효율성을 얻는다. 포레스트 오토마타의 튜플 구조는 바로 이 분리성을 닮아있다. 하지만 분리논리기반 방식은 유연한 방식의 요약이 불가능하다는 단점이 있다. 자료구조가 미리 정해진 predicate 으로 표현되지 않으면 어려움이 있다. 예를들어 분리논리 기반은 아래와 같은 형태의 자료구조를 정확히 표현 못한다고 한다.



가장 왼쪽 노드의 가장 위쪽 필드는 자신의 다른 필드를 가리키는 반면, 가운데 노드의 가장 위쪽 필드는 다른 노드를 가리키고 있다. 가장 왼쪽과 가운데 노드의 차이를 분리논리는 잘 표현 못하는 반면 포레스트 오토마타는 잘 표현가능하다고 하였다.

이 방식이 효과적으로 돌아가게 하기 위해 컷 포인트를 어떻게 설정하는지, 접기와 뭉치기(fold, unfold)는 어떻게 하는지 등 해결해야 할 문제들이 있었고, 어떻게 풀어왔는지 발표자는 설명하였다.

모양분석을 많이 공부해오지는 않아서 온전히 이해하지는 못하였지만, 관심 갖고 지켜볼 가치가 있는 연구결과라는 인상을 받았다.

Automatic Analysis of Open Objects in Dynamic Language Programs

Arlen Cox, Bor-Yuh Evan Chang, Xavier Rival

동적언어에서 open object란 객체의 필드가 실행중에 생겼다 사라졌다 하는 객체를 말한다. 예를들면 다음과 같은 자바스크립트 프로그램에서

```
for(var p in s)
  if(p in c) r[p] = "conflict";
  else r[p] = s[p];
```

객체 r에 필드 p는 객체 s에 있는 필드들 중에서, 객체 c에 있느냐 없느냐에 따라 값이 다르게 결정되는데, 이 때 필드 p가 객체 r에 없었을 경우 새로 생성된다.

자바스크립트는 이렇듯 클래스 구조가 실행 전에 결정되지 않고 실행 과정에서 변화하는 일이 잦다. 만약 다른 객체들과 관계없이 필드가 추가되고 사라진다면 아마 비교적 간단하게 문제가 해결되겠지만, 위의 코드와 같이 다른 객체들과의 어떤 관계를 맺으며 필드가 변화하는 일이 잦다고 한다.

저자들은 다른 객체들과 관계를 맺으며 변화하는 필드구조를 잘 추적하기 위한 분석 틀을 제시한다. 저자들은 2013년도 ECOOP에 객체들의 필드가 갖는 값들의 집합들이 서로 갖는 관계를 추적하는 도메인을 만들었다고 하는데(먹집합 도메인에 관계요소를 좀 추가한 도메인 같다), 이것에 각 객체 필드가 어느 지점을 가리키는지 쫓아가는 모양분석을 결합했다는 인상을 받았다.

좀 자바스크립트 분석에서 중요한 문제를 다룬 것 같긴 하다. 하지만, 더 자세히 살펴봐야 하겠지만 해법이 이미 있는 것들의 결합이라는 인상을 받았고, 논문이나 발표도 그러한 어조로 설명되었으면 더 좋지 않았을까 싶다.

Cyclic Abduction of Inductively Defined Safety and Termination Precondition

James Brotherston and Nikos Gorogiannis

힙을 다루는 프로그램이 안전하게 실행되고 끝날 전제조건(precondition)을 구하는 방법에 관한 연구이다. 예컨대,

```
while x != nil do x = x.next;
```

와 같은 프로그램이 있을 때, 프로그램이 안전하게 끝날 전제조건은 x 가 nil이거나 길이 2의 리스트, 길이 4의 리스트 ... 등 여러 답이 존재할 수 있을 것이다. 이 중 가장 일반적인 답은 x 가 비순환 리스트라는 것이다. 이 연구는 이렇게 가장 일반적인 전제 조건을 찾는다고 한다. 방법은 기본적으로 분리논리에 기반하는데, 분리논리의 단점 중 하나인 자료구조를 기술하는 predicate의 형태가 미리 정해져있어야 한다는 점을 abduction으로 해결한다고 한다. abduction이란 참인 B 가 주어질 때 $A \Rightarrow B$ 를 참으로 하는 참인 A 를 찾는 것이다. 이미 주어진 predicate으로 증명을 구성할 수 없으면, 증명이 되게 하기 위한 것을 abduction으로 자동으로 찾아낸다고 한다.

기본적으로 이 연구는 충분전제조건(sufficient precondition) 즉, 만족될 경우 프로그램이 안전하게 돈다는 것이 보장되는 전제조건을 구한다. 하지만 쿠조 교수님은 이에 대해서 이의를 제기했다. 마이크로소프트에서 개발 및 사용되고 있는 정적분석기 CCchecker의 유저들이 얘기하길, 충분 전제조건보다 프로그램이 죽지 않기 위해 만족시켜야 할 최소한의 요건 (필요 전제조건, 이게 만족되어도 죽을수도 있음)이 실생활에서 더 요긴하다고 한다고 했다. 왜냐하면 보통 충분전제조건은 비현실적인게 나오는 경우가 많기 때문이란단다. 저자들은 이에 대해 필요조건을 지원하려면 좀 더 연구가 필요하다고 했다.

이 연구 또한 초청발표, 포레스트 오토마타를 이용한 모양분석처럼 분리논리 기반 모양분석의 경직성에 문제의식을 느끼고 진행된 일인 것 같다. 내용이 난해해서 다 따라가진 못해서 아쉽다.

3. 사람들

요약해석 창시자인 쿠조 교수 앞에서 발표해 본 것은 이번이 처음이었는데, 생각보다 관심있게 들어주고 피드백도 주셔서 감사하다. 발표가 끝난 다음에는 중국 국방과학기술 대학의 리관첸 교수님과 대만 아카데미아 시니카의 왕 교수님이 오셔서 발표 잘들었다고 격려해주셨다. 왕 교수님은 우리 연구실에 1년간 방문교수로 오셨었고, 리관첸 교수님도 우리 랩과 국방기술대학의 교류 워크샵, 그리고 다른 학회에서 간간히 뵈었기 때문에 반가운 얼굴들이었다. 리관첸 교수님과 국방기술대 출신으로 프랑스 INRIA에서 유학 중인 학생들과 만찬이 끝나고 더 얘기를 나눌 수 있었다. INRIA연구팀과 국방기술대 연구팀은 관계가 긴밀한 것 같다. 여러명의 학생들이 국방기술대에서 석사를 하고 INRIA에서 박사과정을 밟고 있었고, 리관첸 교수님과 INRIA의 연구 협력도 활발해 보인다.

그 외에도 학회 체어인 마커스와 점심시간에 얘기하면서, 제출되었던 cam-ready 버전 논문의 사소한 실수를 수정해주어서 고맙다는 말도 전했다. 우리 연구실에 종종 들렀던 자비에르 교수님과 뮐러 교수님도 얼핏 뵈었지만, 계속 학회에 계시지 않고 나타났다가 사라졌다 하셔서 아쉽게도 얘기를 나누지 못했다.

4. 뮌헨

학회가 끝나고 이틀 여유가 생겨서 구경을 하러 다닐 수 있었다. 10월에는 옥토버페스트라는 큰 맥주 축제가 여기서 열리는데, 안타깝게도 학회기간과 겹치지 않아 공사중인 행사장만 바라봐야 했다. 행사장은 무척이나 커서, 얼마나 많은 돼지와 맥주가 소비될지 가늠도 안됐다.



뮌헨에는 예전에 여행온 적이 있어서 시내보다는 그 밖으로 돌아다니는 데 주력했다. 약 3만원 정도로 독일 남부와 오스트리아 일부를 마음대로 돌아다닐 수 있는 기차표를 살 수 있다는 것을 알고, 학회가 끝난 다음날 기차를 타고 오스트리아 짤츠부르크에 다녀왔다. 사운드 오브 뮤직의 배경이 된 도시로, 모차르트의 생가, 미라벨 정원,

언덕 요새가 인상깊은 곳이다.

그 다음날에는 과학기술 박물관으로는 세계에서 가장 크다는 독일박물관을 구경하고 뮌헨 근교에 다카우 수용소에 다녀왔다. 독일박물관은 아이들 뿐 아니라 어른들도 보고 즐길 수 있는 과학기술 콘텐츠가 많았다. 우리나라에도 이런 종류의 과학기술 박물관이 많아져서 과학기술 친화적인 여론형성에 일조했으면 좋겠다. 다카우 수용소는 뮌헨 근교에 있는 수용소로, 2차세계대전때 나치가 정치범들 및 유대인을 수용하던 곳이다. 영화에서만 보던 나치의 만행과 전쟁의 참상을 직접 피부에 와닿게 느낄 수 있었다. 가스실, 시체화장실, 수용소 생활관, 감시탑, 희생자들 사진을 보고, 인간이 얼마나 악랄해질 수 있었나를 볼 수 있었다.



5. 맺으며

생산적인 자극을 받을 수 있도록 늘 지원해 주시는 센터장 이광근 교수님께 감사드립니다. 또한 이번 출장을 여러가지로 많이 도와주신 ROSAEC 행정팀 분들, 특히 장승영 사무장님께 감사드립니다.