



OOPSLA 2019

Athens, Greece

Junhee Lee

Software Analysis Laboratory

Korea University

19.10.21 - 19.10.25

1 개요

OOPSLA'19 학회에 포스터 발표를 하기 위해 다녀왔다. 이번 학회 참석은 꽤 기대를 하고 갔는데, 그 이유는 OOPSLA는 PL 학회이기 때문에 비슷한 분야의 연구를 많이 볼 수 있을 것 같았고, 리뷰 프로세스가 매우 까다로워 제대로 된 논문을 많이 볼 수 있을 것 같았다. 결론부터 말하자면, 학회 운영측면에서는 조금 마음에 들지 않았지만, 배울 것이 많은 유익한 학회였다.

2 포스터 발표



포스터 발표 포스터 발표를 하면서 배운 점은 당연한 이야기지만 듣는 사람에 맞춰서 설명을 하는게 가장 효과적으로 전달할 수 있다는 것을 배웠다. 처음에 발표할때 포스터에 적힌 motivation 쪽부터 천천히 설명을 하고 있는데, 듣는 사람 얼굴을 보니까 조금 지루해하더니 goal 설명을 듣더니 고맙다고 하고 가버렸다. 그 다음에는 풀고자 하는 문제만 잘 설명하고 바로 아이디어랑 실험 결과를 보여주자고 마음을 먹었는데, 이번에는 문제 설명을 듣더니 잘못된 패치를 하는 예제를 가리키면서 "바로 이렇게 패치하겠다는 거군요" 라고 해서 또 깜짝 놀랐다. 이 사람은 메모리 해제 오류를 잘 몰라서 알람에 적힌대로 패치하면 안된다는 것을 몰랐다. 그래서 문제부터 차근히 설명했다. 그래서 그 다음부터는 반응을 보고 설명하자 해서 문제 설명하고, 잘 알아들으면

실험 결과, 신기해하면 아이디어를 설명했는데 나름 괜찮았던 것 같다. 물론 오자마자 아이디어 부터 설명하라는 사람도 있었고, 문제를 듣더니 결과를 바로 물어보고 아이디어를 건너뛴 사람도 있었다.

코멘트 우리 연구에 대해 크리티컬한 질문을 던진 사람은 없었다. 물론 그런 의문을 갖는 사람들은 발표를 들으러 오지 않는 걸수도 있는데, 전반적으로 memory-leak에 대한 우리 연구는 단점이 크게 없어서 반응이 매우 좋았다. 제일 기분이 좋았던 코멘트는 "언제 이 기술을 써볼 수 있냐?" 였다. 또, 논문으로 있냐고 묻고 ICSE에 냈다고 하니까 좋은 결과를 얻을 수 있을거라고 말한 코멘트도 좋았다.

개선점 먼저 포스터에서 가장 미흡했던 점이라면, 우리 연구가 어떤 문제를 푸는 연구인지를 좀 더 한눈에 보이지 않았다는 점이다. 먼저 포스터 제목에 "for C"를 빠뜨린 점이다. 이걸 넣는게 그렇게 중요한가 싶어서 안 넣었는데, 포스터를 하면서 가장 먼저 들었던 질문이 "이 연구가 C의 메모리 리스를 고치는 것이냐?" 였다. 생각보다 Java의 메모리 리스를 연구하는 사람이 많은것 같았다. Managed language에서 메모리를 효과적으로 관리하는 문제가 조금 올드한 주제가 아닌가 싶었는데 요즘도 많이 하는지 대부분이 질문을 해왔다. 또, 우리가 풀고자 하는 문제를 명확히 드러내는 문장이나 그림이 없었다. 시스템 overview랑 motivation에서 다 설명을 하니깐 괜찮지 않을까 했는데, 실제로 설명하려고 보니까 motivation 파트와 overview를 왔다갔다 하면서 설명하게 되었다. 정말 잘 알려진 문제를 푸는 것이 아니라면, 문제의 input output는 확실하게 드러내는 것이 좋을 것 같다.

3 OOPSLA 발표

3.1 새로 알게된 연구들

A Path to DOT: Formalizing Fully Path-Dependent Types 이 연구는 문제 해결 방식이 정말 진지하고 제대로 했다는 인상이 깊게 남은 연구이다. 이 연구는 Scala 언어에서 path-dependent type을 허용하지만 현재의 타입 시스템 Dependent Object Type (DOT)이 이를 제대로 formalize 하는 못하는 점을 지적한다. 그리고, 이 문제를 해결하기 위해 DOT을 확장하여 path dependent type을 지원하는 pDOT을 제시한다.

Type system 분야의 문제를 그동안 잘 몰랐기 때문에 조금 더 자세히 기술한다. 최신의 프로그래밍 언어들은 추상화된 타입을 정의하고 (e.g., tree) 이 타입을 이용하여 다양한 instance의 type을 새로 정의할 수 있게끔 돕는다 (e.g., red-black tree). 이들은 프로그래밍을 체계적이고 효율적으로 할 수 있게 돕는다. 이때, 이렇게 복잡한 type system을 잘 formalize 하는 calculus (formal system)를 디자인 하는 것이 매우 중요하다고 한다. 이를 통해 type system의 오류를 찾거나, 새로운 언어를 디자인하는데 도움을 준다고 한다. 반면에 이러한 calculus를 잘 디자인 하지 못하면, unsound type system 위에서 프로그래밍을 하게 되거나 매우 제한적인 패턴만이 가능한 언어를 사용하게 된다.

어프roach를 잘 이해하지 못했지만 그래도 문제를 잘 풀었다고 생각이 드는 점은 구체적이고 motivating이 되는 목표와 이를 해결하기 위한 어프roach를 기존 연구와 끊임없이 비교하고 non-trivial한 부분을 계속 드러내면서 전달했기 때문이라고 생각한다. 나중에 type을 제대로 공부하게 되면 정독해볼만 한 연구인 것 같다.

Duet: An Expressive Higher-Order Language and Linear Type System for Statically Enforcing Differential Privacy Differential privacy가 요즘 정말 security 쪽에서 중요하고 반드시 지켜야 할 개념이라고 한다. Differential privacy가 무엇이나면, 개인 정보가 포함된 dataset을 받아 분석하여 유용한 정보를 주되, 해당 정보에 개인 정보가 드러나지 않게끔 하는 시스템을 말한다. 용어에 differential이 들어간 이유는 이 개념이 수학적으로 differential로

정의되어 있기 때문이다. Informal하게 말하면, 한 개인의 privacy만이 다른 두 데이터셋의 분석 결과가 거의 동일할 때 differential privacy를 지킨다고 말한다고 한다.

이 연구에서 지적하는 문제는 differential privacy가 매우 중요함에도 특정 프로그램이 differential privacy를 제공하는지는 전문가가 직접 확인한다는 것이다. 이 논문이 제시하는 해결책은 differential privacy를 증명할 수 있는 language를 디자인 한 것이다. 이 language의 특이한 점이라면, 먼저 differential privacy가 numeric한 정의이기 때문에 linear type system을 채택한 점과 privacy와 sensitivity를 각각 계산하는 두 언어를 제시한 점이다.

해당 논문의 자세한 내용은 differential privacy와 이를 구현하는 machine learning, 증명에 쓰인 type system 등 미리 알아둬야 할 내용이 많아 자세히 파악은 하지 못했지만, 흥미롭고 관심있게 지켜볼 주제인 것은 확실한 것 같다. 이 논문 또한 나중에 정독하려고 한다.

3.2 재미있는 연구들

BDA: Practical Dependence Analysis for Binary Executables by Unbiased Whole-Program Path Sampling and Per-Path Abstract Interpretation 이 연구는 바이너리에 대해 dependency를 분석하는 연구이다. 이 논문이 하고자 한 것은 간단하다. 바이너리에 대해 정확하면서도 sound한 분석기를 만드는 것이 매우 어렵기 때문에, 패스 별로 분석하겠다는 것이다. 이때 특정 패스에 치우치면 안되므로, 가능한 다양한 패스를 고르도록 가이드 하는것이 이 논문의 핵심이다.

이 연구의 키 포인트는 좋은 문제 설정인 것 같다. 패스 별로 분석하면 당연히 정확한 분석 결과가 나오지만, 일반적으로는 이를 단순 statement coverage를 올리는 방식으로 분석을 한다고 해서 분석을 잘했다고 말할 수 없다. 그렇지만, 문제를 컴파일러 최적화를 위한 data dependency를 분석하는 문제로 설정한다면, statement coverage를 잘 달성하는 정도로도 충분한 정보를 얻기에 부족하지 않게 된다.

Static Analysis with Demand-Driven Value Refinement 이 연구는 문제를 제대로 풀었다는 느낌이 들어 기억에 남는다. 이 연구가 하고자 하는 것은 Javascript를 정확하고 빠르게 분석하는 것인데, 문제를 정말 잘 풀었다고 느끼는 점은 concrete한 문제 현상을 바탕으로 어프로치를 제시한 점이다. 정확도가 망가지는 문제를 일반적인 해결책 (e.g., context-sensitivity tuning)을 이용하여 해결하는 것이 아니라, 정확도가 망가뜨리는 원인 (dynamic property read/write)을 파악하고 이를 잘 드러내는 example을 찾아서 이에 적절한 해결책을 제시했다는 느낌이 든다.

이 연구는 자바 스크립트에서 정확도에 큰 영향을 끼치는 것이 dynamic property read/write라고 지적하고 이를 잘 분석하기 위한 방법을 제시하였다. 예를 들면 `source[name] = func`이라는 코드는 `source` 변수의 dynamic field "name"에 함수 `func`을 저장하는 statement이다. 실행중에 정의되는 field값을 읽는 코드는 분석기의 정확도를 망가뜨리는 주된 원인이다. 이 연구는 이와 같이 정확도에 큰 영향을 주는 statement를 만나면, 분석 state를 `func`이 가질수 있는 값으로 나누고, 각각의 경우에 대해 더 정확한 값을 계산하는 방법을 제시하였다.

3.3 PL스러운 연구

Language-Integrated Privacy-Aware Distributed Queries 이 연구가 기억에 남는 이유는 database 분야의 문제를 가지고 와서 문제를 직접 풀지 않고 type system을 제안하고 이를 이용하여 문제를 해결한 점이다. 정말 PL 학회다운 연구라고 느꼈다.

이 연구가 풀고자 한 문제는 다음과 같다. 여러 데이터베이스에서 특정 쿼리에 대답하는 문제를 생각해보자. 이때, 각 데이터베이스를 어떤 순서로 접근할 지, 독립적으로 가져올지, 언제 join 할지에 따라 퍼포먼스가 많이 차이난다고 한다. 또, 이를 잘못 설정하면 privacy가 깨지기도 한다. 이 연구에서 풀고자 하는 문제는 프라이버시를 유지하면서도 퍼포먼스가 가장 좋은 placement를 고르는 문제이다. 이 논문은 이 문제를 해결하기 위해 type system을 제안하고 이를 이용하여 문제를 해결한다. 해결 방식은 프로그램 합성과 비슷하다: information-flow type system을 이용하여 candidate를 type-directed 방식으로 나열하고 constraint solving 문제를 풀어 가장 좋은 placement를 고르는 것이다.

4 그리스

그리스의 분위기는 작년에 갔던 스웨덴과는 많이 달랐는데, 전반적으로 지저분하다는 인상을 받았다. 벽에는 하나같이 낙서가 되어있고, 엄청 큰 쓰레기통이 길거리에 있고, 자동차들은 길 양쪽으로 전부 주차가 되어있었다. 또, 비둘기도 엄청 많아서 야외에서 밥을 먹으면 손님이 먹다 남긴 음식을 비둘기가 올라와서 먹는 것을 구경할 수 있었다. 밤에는 관광지라서 그런지 많이 어두컴컴하고 거지도 많아서 조금은 위험한 느낌이 들었다.

그리스에서 사실 제일 기대했던 것은 관광지였다. 구글지도로 검색했을때 아크로폴리스, 파르테논 신전 등 말로만 듣던 그리스 신화와 관련된 신전들이나 유물들을 볼 수 있을까 기대했었는데, 대부분 부셔져있어 많이 아쉬웠다. 그리스 음식은 수불라끼가 가장 맛있었다. 까르보나라에 계란을 풀어 넣는 대신에 계란 후라이를 얹는 그리스 전통 파스타도 있었고, 구운 채소를 굉장히 자극적인 소스에 절인 음식도 있었는데 이들은 입맛에 좀 맞지 않았다. 대부분 너무 짜고 자극적이었다. 그리스에서 특이했던 점은 수학 기호로만 보던 그리스 문자를 어디에서나 볼 수 있었다는 점이다. 예를 들어, 지하철 역을 나타내는 ST는 ΣT 로 쓰여있고, 영문자 L 대신에 Λ 를 쓰는 것도 심심치 않게 볼 수 있었다.

5 맺음말

전반적으로 훌륭한 학회였다. 전달력이 조금 떨어지는 발표들은 있었지만, 모두 문제를 잘 정의하고 괜찮은 아이디어를 내고 좋은 결과를 내는 연구들이었다. 의문을 남기는 발표들도 있었는데, 그런 발표의 질문에는 꼭 기존 연구로 이 문제가 해결이 되지 않는지를 물었다. 반면에 재밌는 발표에는 이 기술이 어디까지 적용이 되는지를 주로 관심 갖는 것 같았다. 다음 연구는 FSE 혹은 OOPSLA에 제출하게 될 것 같은데, OOPSLA가 개인적으로는 좀 더 재미있을 것 같다. 마지막으로, 좋은 경험을 하게 해주신 교수님께 감사드립니다.

