



SPLASH 2023 in Cascais, Portugal

이석현

2023.11.25. - 2023.11.27.

초록

포르투갈 카스카이스에서 진행된 SPLASH 2023에 다녀왔다. 이번에는 대학교 신입생 시절부터 친하게 지낸 강찬구 연구원이 본인의 양자 연구에 대해 발표했다. 이외에도 학회장 주변을 둘러본 것과 다른 연구들에 대해서 간략히 전달하고자 한다.



(a) 호텔의 전경

(b) OOPSLA 메인 학회장 문 앞에서 바라본 바다

(c) 음료를 고정하기 위한 장치

그림 1. SPLASH 2023이 개최된 호텔 카스카이스 미라젬 헬스 앤 스파

1 개요

SPLASH 2023에 참여하기 위해서 11월 22일부터 11월 30일까지 6박 9일간 포르투갈에 다녀왔다. 이번 학회 발표되는 수 많은 재미있는 논문들을 듣고 생각해 볼 수 있는 기회가 되었고, 포르투갈과 경유지인 핀란드의 모습도 짧게 둘러 볼 수 있는 기회가 있었다.

2 OOPSLA 2023

OOPSLA¹는 소프트웨어 공학, 프로그래밍 언어, 시스템 및 애플리케이션에 관한 국제 학술 대회인 SPLASH²의 중심이 되는 학회 중 하나로, 객체지향 프로그래밍과 소프트웨어 공학 분야에 중점을 둔 학회이다. 이번장에선, 이번에 OOPSLA 2023에 다녀온 경험을 간단히 정리하고자 한다.

학회장소 OOPSLA 2023는 포르투갈 리스본 근처의 휴양 도시인 카스카이스의 바다가 보이는 호텔 카스카이스 미라젬 헬스 앤 스파에서 개최되었다. 그림 1a는 SPLASH가 진행된 호텔의 전경이고, 그림 1b는 OOPSLA가 진행된 호텔 내부에서 바라본 바다의 모습이다. 호텔은 전체적으로 크고 깔끔했으며, 특히나 바다가 보이는 전망이 너무 좋았다. 또한, 호텔에서

¹Object-Oriented Programming, Systems, Languages & Applications

²Systems, Programming, Languages, and Applications: Software for Humanity

제공하는 кей터링 서비스도 좋았는데, 바닷가 도시인 만큼 해산물이 많이 나오는 것이 특징이었다. 한가지 인상이 깊었던 점은, 그림 1c와 같이 모든 접시에 음료를 담은 컵을 들 수 있도록 고안된 장치가 있어서, 식사를 위한 자리가 부족함에도 불구하고 음료를 편히 들고 다니면서 식사를 할 수 있었다. 단 한가지 아쉬웠던 것은 숙소에서 학회장까지의 길이 좁고 도로 포장 공사중인 곳이 많아 걸어가기가 불편했다는 것이다.

논문 OOPSLA에서는 프로그래밍 언어와 관련된 다양한 주제의 논문들이 발표되었다. 이 중에 내가 가장 관심있게 들었던 논문들을 몇가지 소개한다. 순서는 논문의 발표 시간 순이다.

1. **Grounded Copilot: How Programmers Interact with Code-Generating Models [2].** 다양한 개발자들이 Github Copilot [4]을 이용하는 방법을 가속과 탐색의 관점에서 서술한다. 20명의 개발자들을 섭외하여 스터디를 진행하고 이를 분석한 결과를 정리하였다. 스터디 결과는 Github Copilot을 사용해봤다면 모두 알고 있을 법한 내용을 한 문장으로 정리한 정도여서 큰 인상은 없었다. 하지만 이를 기록하고 보여주는 방법은 특히 인상적이었다. 각각의 이론을 실제 스터디에서의 사례와 연결하여 보여주는 방식은 학회장의 모든 청중이 공감하여 웃음을 자아내기도 하였다. 최근에 진행하고 있는 연구에서도 여러 사람들의 생각을 모으는 스터디를 진행하고 있는데, 이번 발표가 도움이 될 것 같다.
2. **Concrete Type Inference for Code Optimization using Machine Learning with SMT Solving [9].** 요즘 가장 뜨거운 관심을 받고 있는 Large Language Model(이하 LLM)과 SMT Solver를 결합하여 동적 타입 언어의 최적화를 위한 타입 추론을 수행하는 방법을 제안한다. LLM은 빠르게 타입을 추론 할 수 있지만 정확도가 떨어지는 반면, SMT Solver는 정확도가 높지만 느리다. 이 연구에서는 이 둘의 장점을 적절히 조합하여 빠르지만 여전히 정확한 타입 추론 방법을 고안하였다. LLM 기반의 기술에 기존의 프로그래밍 언어 분야에서 발전해 오던 기술을 접목하여 눈에 띄는 성능 향상을 이루는 것이 생각만큼 쉽지 않다는 것을 알고 있었는데, 이를 적절히 조합하는데 성공했다는 것이 인상적이었다.
3. **Explainable Program Synthesis by Localizing Specifications [6].** 프로그램 합성에서 합성된 프로그램을 이해하는 것은 어려운 일이다. 이 논문에서는 합성된 프로그램을 이해하기 위해 합성된 프로그램의 일부분을 이루는 부분 명세를 제안한다. 부분 명세를 통해서 합성된 프로그램의 일부분만을 이해하고, 수정하거나 더욱 효율적인 합성을 할 수 있다고 주장하고 있다. 앞서 언급한 논문(1.)과 비슷하게 제안한 기술의 유용성을 증명하기 위해 사용자 스터디를 진행하였는데, 두 경우 모두 작은 수의 사용자(20명)를 대상으로 진행하였다. 진행 중인 스터디에서 목표로 할 사용자 수에 관한 고민이 있었는데, 참고할 수 있는 좋은 사례가 되었다.
4. **Data Extraction via Semantic Regular Expression Synthesis [3].** 기존에 정규 표현식은 문자열의 문법적인 요소만 고려할 수 있었다. 이 논문에서는 기존의 정규 표

현식보다 표현력이 강력한 새로운 정규 표현식을 만들기 위해 기존의 정규 표현식에서 사용되는 문자열의 의미적인 요소를 고려하는 방법을 제안한다. 문자열의 구성요소의 의미를 파악하기 위해서 LLM을 활용하였는데, 이를 활용하는 방법이 과거에 내가 진행했던 연구와 비슷해서 기억에 남는다. 생성하고자 하는 정규 표현식이 만족해야한 스펙으로부터 LLM은 구멍이 있는 프로그램(이하 스케치, Sketch)을 만들어내고, 합성기로 이를 구체화하여 모든 스펙을 만족하는 정규 표현식을 만들어 내는 것이다. 하지만, 단순히 스케치만을 이용하는 것은 여전히 큰 성능 향상을 이루기 어려운 것을 경험하였는데, 이 논문에서는 만들어진 스케치를 분해하고 수정하는 방식으로 LLM의 도움을 최대한 활용하였다. 논문의 구체적인 내용과는 별개로 논문에서 제안하는 것이 “의미적인 정규 표현식(Semantic Regular Expression)”인데, 여기에 왜 “정규 표현식”이라는 이름이 붙었는지는 공감이 되지 않는다.

5. **Modular Component-Based Quantum Circuit Synthesis [5]**. 초록에 언급한 논문이다. 열심히 준비한 만큼 발표도 잘 했다. 질의응답까지 완벽하게.
6. **A Pretty Expressive Printer [7]**. 포맷터(Formatter)는 소스 코드의 스타일을 일정하게 유지하기 위해서 정말 많은 곳에서 사용하는 소프트웨어이다. 기존에는 포맷터는 엄청나게 많은 휴리스틱의 집합으로 이루어진 소프트웨어인 줄 알았다. 이번 발표를 통해서 다양한 형식의 출력을 지원하는 포맷터를 만들기 위한 연구분야가 있다는 것을 알게되었다. “세상의 모든 것이 연구 주제일 수 있다”는 말을 들은적이 있는데, 이 논문을 보고 그 말이 맞다는 것을 느꼈다.
7. **Bidirectional Object-Oriented Programming: Towards Programmatic and Direct Manipulation of Objects [10]**. 코드로부터 그래픽을 합성하고 합성된 그래픽을 수정하여 원본 코드에 반영하는 양방향 프로그래밍이라는 것이 있다. 이 논문에서는 객체지향 프로그래밍 언어에서는 클래스 간의 상속과 바뀔 수 있는 변수를 적극적으로 활용하는 특징 때문에 양방향 프로그래밍을 적용하기 어렵다고 주장하고 있고, 이를 해결하기 위한 방법을 제안하고 있다. 하지만 이번 발표에서는 언급하는 어려움이 실제로 어려운 것인지 잘 와닿지 않았고, 해결방법으로 제안된 것 또한 근본적으로 지적하는 어려움을 해결했다는 인상을 받지 못했다.
8. **Bring Your Own Data Structures to Datalog [8]**. Datalog에 새로운 데이터 구조를 추가하기 위해서는 datalog 실행기를 직접 수정해야 하는데, 이 과정은 매우 어렵다. 이 논문에서는 datalog에 새로운 데이터 구조를 쉽게 정의하기 위한 메타 프로그래밍 언어를 제안한다. 사실 자세한 내용은 이해하지 못했지만, 이렇게 프로그래밍 언어 연구자라는 느낌을 강하게 받았다.
9. **All the Languages Together [1]**. 여러 언어를 함께 사용하는 환경에서 언어 간의 상호작용을 검증하는 방법에 대해서 서술하는 기초연설이다. 처음에 기대했던 것은 여러 언어들이 공유하고 있는 공통적인 특징을 정리하여 이를 활용하여 다양한 언어에서 사용할 수 있는 검증기였는데, 실제 내용은 약간의 정보가 포함된 저급 언어를 디자인한



그림 2. 핀에어 항공의 북극 항로 인증서

후 여러 언어를 저급 언어로 컴파일하여 검증하는 방법을 제안하고 있었다. 발표를 들으면서 기존에 오랫동안 사용하고 있던 LLVM IR 등을 두고 이 언어를 사용해서 얻을 수 있는 장점이 없는 것 같다는 생각이 들었지만, 질문이 너무 공격적이어서 질문을 직접 하지는 못하였다.

3 여정

카스카이스에 다녀오는 긴 일정 중에 SPLASH 2023에 참여하는 시간 이외에는 주변을 둘러볼 기회가 있었다.

비행 학회가 열리는 장소인 포르투갈 리스본으로 가는 직항 비행기가 없어 핀란드 헬싱키를 경유하는 핀에어(Finnair) 항공을 이용하였다. 비행기의 좌석은 다른 항공사에 비해 꽤나 넓어서, 장시간의 비행에도 큰 불편함이 없었다. 또한 인천에서 헬싱키로 향하는 비행의 경우는 북극을 지나는 경로로 가는데, 그림 2와 같이 이를 증명하는 증명서를 만들어 주는 것 또한 재밌었다. 이번 비행에는 한가지 잊지 못할 추억이 하나 더 있는데, 내가 사용하던 태블릿을 비행기에 두고 내렸다는 것이다. 비행기를 타고 리스본으로 향하는 길에 태블릿을 두고 내렸는데, 포르투갈과 헬싱키에 머무는 동안 여러 곳에 연락을 하며 찾아봤지만 결국은 찾지 못하고 한국으로 돌아왔다. 이 글을 쓰고 있는 지금도 태블릿을 찾았다는 연락을 받지 못했다.



(a) 북쪽

(b) 동쪽

(c) 남쪽

그림 3. 숙소에서 본 풍경



(a) 카스카이스의 거리

(b) 유라시아 대륙의 끝, 호카곶

(c) 페나성

그림 4. 카스카이스의 풍경과 관광지

숙소 이번 숙소는 정말 만족스러웠다. 숙소를 처음 안내해주시는 분과 함께 문을 열고 들어가는 순간부터 바라보이는 풍경에 감탄을 금치 못했다. 그림 3과 같이 숙소에서는 입구가 있는 서쪽을 제외하고는 탁 트인 풍경이 보였다. 특히나 동쪽과 남쪽으로 보이는 풍경은 아침부터 밤까지 매 순간이 다른 풍경을 보여주었다. 혹여나 다시 카스카이스에 간다면 이 숙소를 다시 이용하고 싶다.

카스카이스, 포르투갈 카스카이스는 리스본에서 서쪽으로 30km 정도 떨어진 곳에 있는 작은 휴양 도시이다. 그림 4는 카스카이스의 풍경과 근처 관광지를 다녀온 기록이다. 카스카이스의 건물은 대체로 작고 밝아서, 우리나라의 거리와 달리 아기자기하고 따뜻한 느낌이 들었다. 카스카이스 시내 뿐만 아니라 근처의 유라시아 대륙의 최서단인 호카곶과 유네스코 문화 유산인 페나성도 다녀올 기회가 있었다. 호카곶에서 바라보는 대서양의 풍경은 말 그대로 적막함이었다. 특히, 하늘이 흐려 그런 느낌이 더 강했던 것 같다. 그런 바다를 탐험 할 생각을 한 과거의 사람들의 용기에 감탄했다. 페나성은 다양한 양식이 섞인 성과 그 주변을 둘러싼 정원으로



(a) 리스본을 지키던 벨렝탑



(b) 상조르즈성에서 내려다 본 리스본



(c) 수 많은 광장 중 리스본 시청 앞 광장



(d) 리스본의 트램을 위해 복잡하게 얽힌 전선

그림 5. 리스본의 다양한 풍경

이루어진 곳인데, 오전부터 흐리던 날씨가 점점 장대비로 바뀌어 구석구석 자세히 둘러보지 못해 아쉬웠다.

리스본, 포르투갈 학회의 마지막이 지나고, 귀국행 비행기를 타기 전까지 시간이 남아 리스본을 둘러보았다. 카스카이스와는 또 다른 분위기의 리스본은 매력적인 곳이었다. 카스카이스가 장난감 같은 도시라면, 리스본은 그 장난감의 배경을 보여주는 것 같았다. 리스본의 거리는 카스카이스보다는 조금 더 복잡하고, 건물들도 더욱 높았다. 그런 건물들 사이로 지나다니는 트램은 리스본의 특징이라고 말하기에 손색이 없었다. 리스본의 거리를 조금만 걸다보면, 어느새 리스본의 어느 광장에 도착하게 된다. 정말 많은 광장이 있었고, 각 광장마다 서로 다른



(a) 헬싱키 대성당



(b) 우연한 기회에 방문하게 된 150년 전통의 에크베르그



(c) 템펠리아우키오 교회



(d) 우스펜스키 대성당과 성당이 비친 발트해

그림 6. 헬싱키의 다양한 풍경

이야기를 담고 있었다. 리스본을 이곳저곳 둘러보면서 건물들이 옛모습을 잘 간직하고 있는 것이 부러웠다. 우리나라도 이렇게 옛모습을 잘 간직하고 있는 건물들이 많았으면 좋겠다는 생각이 들었다.

헬싱키, 핀란드 앞서 언급했듯 포르투갈까지의 왕복 여정 중에 헬싱키를 경유하였다. 환승 일정이 길어 헬싱키도 간단하게 둘러보았다. 핀란드는 포르투갈과는 정반대의 분위기를 가지고 있었다. 포르투갈은 따뜻하고 활기찬 분위기였다면, 핀란드는 겨울 햇살이 비치는 오후같이 차갑지만 포근한 분위기였다. 그림 6을 보면 알 수 있듯이 사람이 많지 않고 날씨가 청명하고 잔잔해서, 조용하고 평화로운 분위기를 느낄 수 있었다. 특히, 그림 6d의 성당이 비친 물이

호수가 아니라 발트해라고 생각하면 이곳이 얼마나 잔잔한 곳인지 알 수 있을 것이다.

4 마치며

이번 SPLASH는 매우 유익하고 재미있는 학회였다. 학회를 여러번 다녀봤지만, 다닐 때마다 매번 새롭고 점점 더 재밌어진다. 이번에도 많은 것을 배우고, 많은 것을 느꼈다. 마지막으로, 짧지 않은 일정동안 편하게 다닐 수 있게 많은 준비를 해준 연구실 동료들에게 고맙다는 말을, 좋은 기회를 지원해주신 오학주 교수님께 감사한다는 말을 전하고 싶다.

참고문헌

- [1] AHMED, A. All the languages together. The ACM SIGPLAN International Conference on Systems, Programming, Languages and Applications: Software for Humanity, 2023.
- [2] BARKE, S., JAMES, M. B., AND POLIKARPOVA, N. Grounded copilot: How programmers interact with code-generating models. *Proc. ACM Program. Lang.* 7, OOPSLA1 (2023).
- [3] CHEN, Q., BANERJEE, A., DEMIRALP, C., DURRETT, G., AND DILLIG, I. Data extraction via semantic regular expression synthesis. *Proc. ACM Program. Lang.* 7, OOPSLA2 (2023).
- [4] GITHUB. Github copilot your ai pair programmer. <https://github.com/features/copilot>. Accessed: 2023-10-25.
- [5] KANG, C. G., AND OH, H. Modular component-based quantum circuit synthesis. *Proc. ACM Program. Lang.* 7, OOPSLA1 (2023).
- [6] NAZARI, A., HUANG, Y., SAMANTA, R., RADHAKRISHNA, A., AND RAGHOTHAMAN, M. Explainable program synthesis by localizing specifications. *Proc. ACM Program. Lang.* 7, OOPSLA2 (2023).
- [7] PORNCHAROENWASE, S., POMBRIO, J., AND TORLAK, E. A pretty expressive printer. *Proc. ACM Program. Lang.* 7, OOPSLA2 (2023).
- [8] SAHEBOLAMRI, A., BARRETT, L., MOORE, S., AND MICINSKI, K. Bring your own data structures to datalog. *Proc. ACM Program. Lang.* 7, OOPSLA2 (2023).

-
- [9] YE, F., ZHAO, J., SHIRAKO, J., AND SARKAR, V. Concrete type inference for code optimization using machine learning with smt solving. *Proc. ACM Program. Lang.* 7, OOPSLA2 (2023).
- [10] ZHANG, X., GUO, G., HE, X., AND HU, Z. Bidirectional object-oriented programming: Towards programmatic and direct manipulation of objects. *Proc. ACM Program. Lang.* 7, OOPSLA1 (2023).