

PLDI 2024 Trip Report

2024.06.24 ~ 2024.06.30, Copenhagen, Denmark

고려대학교 소프트웨어분석연구실 김동욱



1 소개

전민석 박사님의 논문 “**PL4XGL: A Programming Language Approach to Explainable Graph Learning**”이 프로그래밍 언어 분야의 최상위 학술대회인 PLDI에 게재되어서 참여 할 수 있는 기회를 얻게 되었다. 나에게 이번 학회 참가는 생애 처음 해외 방문 경험이자 학회 참가였기 때문에 의미가 깊었다. 이번 학회 참가로 내가 경험했던 것과 느낀 바를 이 글을 통해 공유하려고 한다.

2 학회 발표들

Keynote-The Future of Fast Code: Giving Hardware What It Wants

하드웨어의 성능을 최대한 활용하기 위한 프로그래밍 시스템을 어떻게 설계할 것인지에 대한 발표였는데 이 발표가 던지는 메시지가 흥미롭게 느껴졌다. 1969년에 처음 생산을 시작한 보잉 747을 예시로 들면서 50년은 기술 발전의 역사에 있어서 길지 않은 시간이며 ‘소프트웨어’ 또한 50년 간 크게 변하지 않았다고 말한다. 하지만 컴퓨터 하드웨어의 아주 빠르게 변화하고 있고 소프트웨어, 특히 프로그래밍 모델은 그렇지 못했기 때문에 하드웨어와 소프트웨어의 간극이 점점 커지다는 주장이었다. 실제로 IBM의 듀얼코어 프로세서 POWER4가 2001년에 출시한 반면 C++에서 표준 템플릿 라이브러리의 병렬 알고리즘을 지원하기 시작한 것이 2017년 이후인 것을 생각하면 합당한 지적인 것 같다.

Paper-Syntactic Code Search with Sequence-to-Tree Matching

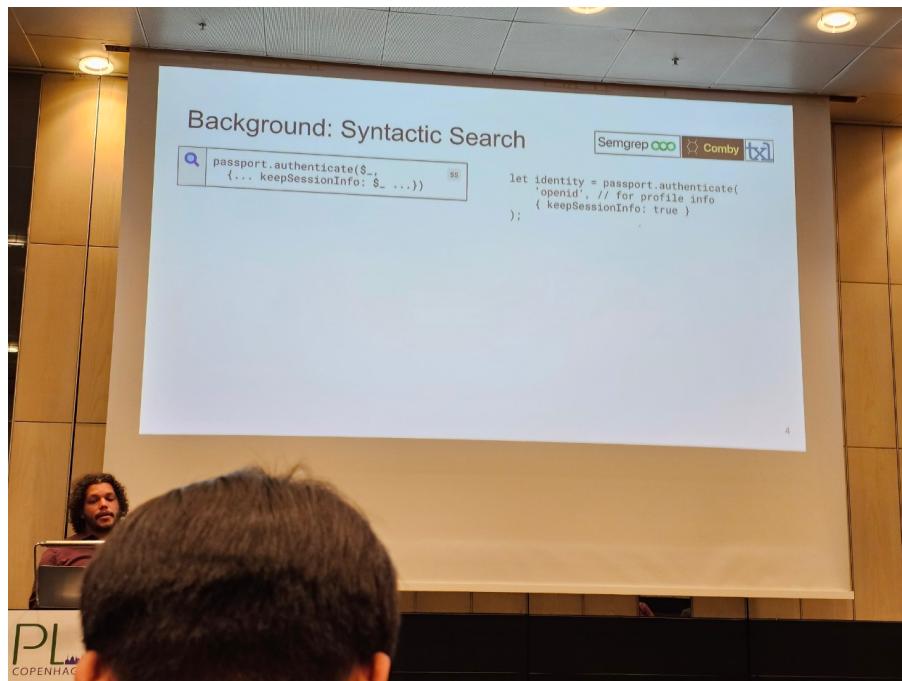


Figure 1. Syntactic Code Search with Sequence-to-Tree Matching, PLDI 2024

코드의 트리 구조에 기반한 구문 분석 도구인 Semgrep, Comby는 쿼리 언어를 사용하여 코드를 검색한다. 하지만 그림 2의 예제처럼 쿼리가 완전하고 파싱 가능한 상태여야만 하는데 이 연구는 미완성 쿼리 (작성 중인 쿼리)로도 동작할 수 있는 새로운 구문 매칭 알고리즘을 제안한다.

```
comby 'failUnlessEqual(:[a],:[b])' 'assertEqual(:[a],:[b])' example.py
```

```
--- example.py
+++ example.py
@@ -1,6 +1,6 @@
 def test(self):
     r = self.parse("if 1 fooze", 'r3')
-
+    self.failUnlessEqual(
+        self.assertEqual(
+            r.tree.toStringTree(),
+            '(if 1 fooze'
+        )
)
```

Figure 2. Comby를 이용한 코드 리팩토링

일반적인 검색 흐름인 “쿼리 -> 렉서 + 파서 -> 트리” 변환을 통한 tree-to-tree 매칭 알고리즘은 파싱 불가능한 쿼리를 처리할 수 없다. 대신, 파서를 제거한 “입력(쿼리)-> 렉서 -> 토큰 시퀀스” 변환을 통한 sequence-to-tree 매칭 알고리즘을 새롭게 도입하였다. 해당 매칭 알고리즘은 와일드카드를 도입한 새로운 쿼리 언어로 동작하는데 이 와일드카드를 검색할 코드의 서브 트리와 매칭하는 것이 알고리즘의 핵심이다.

내가 들은 발표 중에 문제 상황이 (내 기준에서) 가장 직관적인 연구여서 듣자마자 “맞아. 문자열 기반 검색 방식처럼 트리 기반 검색 방식도 검색 도중에 결과가 나오면 정말 편리할 것 같은데?”라는 생각이 들었다. 논문을 찾아보니 작성 중에 실시간으로 상호작용하는 검색 시스템을 표방한 것 치고는 특정 상황에서 오버헤드가 굉장히 심각했던 점이 (무려 230초. 하지만 24ms 이하인 경우가 99%라고 한다) 조금 아쉬웠지만 이 연구를 기점으로 후속연구가 계속 나와 보완되어 가면 좋겠다는 생각이 든 인상 깊은 연구였다.

Paper-SMT Theory Arbitrage: Approximating Unbounded Constraints using Bounded Theories



Figure 3. SMT Theory Arbitrage: Approximating Unbounded Constraints using Bounded Theories

SMT solver의 고질적인 문제인 시간 비용에 관한 문제는 우리 연구실에서도 연구하시는 분이 계셔서 친숙한 문제였다. 이 연구는 SMT solver가 잘 풀지 못하는 정수, 실수 등의 한정되지 않은 제약조건을 SMT solver가 잘 푸는 비트벡터, 부동소수점의 한정된 제약조건으로 치환하는 방법을 제안한다. 여기서 챌린지는 제약조건을 얼마만큼 한정시켜야 할지에 대한 정답이 없다는 것이다. 범위가 너무 크면 (예를 들어, 정수를 128비트 벡터로 변환하면) 비용 감소 효과가 작고, 범위가 너무 작으면 (예를 들어, 정수를 16비트 벡터로 변환하면) SMT solver의 정확도가 현저히 떨어진다.

$$\begin{aligned}
 \llbracket n \rrbracket &= (\lceil |n| (\log_2 10) \rceil + 1, \text{dig}(n)) \\
 \llbracket \text{var} \rrbracket &= (x_m, x_p) \\
 \llbracket n \rrbracket &= \lceil |n| (\log_2 10) \rceil + 1 & \llbracket (\text{ite } b E_1 E_2) \rrbracket &= (\max(\llbracket E_1 \rrbracket_m, \llbracket E_2 \rrbracket_m), \\
 \llbracket \text{var} \rrbracket &= x & & \max(\llbracket E_1 \rrbracket_p, \llbracket E_2 \rrbracket_p)) \\
 \llbracket (\text{ite } b E_1 E_2) \rrbracket &= \max(\llbracket E_1 \rrbracket, \llbracket E_2 \rrbracket) & \llbracket (-E_1) \rrbracket &= \llbracket E_1 \rrbracket \\
 \llbracket (\text{abs } E_1) \rrbracket, \llbracket (-E_1) \rrbracket &= \llbracket E_1 \rrbracket & \llbracket (+ E_1 E_2) \rrbracket, \llbracket (-E_1 E_2) \rrbracket &= (\max(\llbracket E_1 \rrbracket_m, \llbracket E_2 \rrbracket_m) + 1, \\
 \llbracket (+ E_1 E_2) \rrbracket, \llbracket (-E_1 E_2) \rrbracket &= \max(\llbracket E_1 \rrbracket, \llbracket E_2 \rrbracket) + 1 & & \max(\llbracket E_1 \rrbracket_p, \llbracket E_2 \rrbracket_p)) \\
 \llbracket (* E_1 E_2) \rrbracket &= \llbracket E_1 \rrbracket + \llbracket E_2 \rrbracket & \llbracket (* E_1 E_2) \rrbracket &= (\llbracket E_1 \rrbracket_m + \llbracket E_2 \rrbracket_m, \llbracket E_1 \rrbracket_p + \llbracket E_2 \rrbracket_p) \\
 \llbracket (/ E_1 E_2) \rrbracket &= \max(\llbracket E_1 \rrbracket, \llbracket E_2 \rrbracket) & \llbracket (/ E_1 E_2) \rrbracket &= (\max(\llbracket E_1 \rrbracket_m, \llbracket E_2 \rrbracket_m), \infty) \\
 \llbracket (\text{mod } E_1 E_2) \rrbracket &= \llbracket E_1 \rrbracket & \llbracket (\text{boolop } E_1 E_2) \rrbracket &= (\max(\llbracket E_1 \rrbracket_m, \llbracket E_2 \rrbracket_m), \\
 \llbracket (\text{boolop } E_1 E_2) \rrbracket &= \max(\llbracket E_1 \rrbracket, \llbracket E_2 \rrbracket) & & \max(\llbracket E_1 \rrbracket_p, \llbracket E_2 \rrbracket_p))
 \end{aligned}$$

(a) Abstract semantics for integer operations. (b) Abstract semantics for real number operations.

Figure 4. 연산자들의 abstract semantic

이 문제를 해결하기 위해서 abstract Interpretation을 통해 변환 전과 같은 결과를 유지하기 위해 필요한 범위를 추론한다. 예를 들어 정수 도메인의 “ $E_1 + E_2$ ”라는 식이 있다고 가정하자. 표현식 E_1, E_2 로부터 비트 길이를 전파 받는데 E_1, E_2 의 비트 길이가 모두 최대 4비트인 상황이다. 이 때 $E_1 + E_2$ 의 비트 길이는 최대 5비트이고 다시 이 비트 길이를 AST 상에서 위로 반복하며 전파한다. 아이디어가 심플하지만 비선형 정수 도메인에서 평균 1.4배의 속도 향상이라는 좋은 결과가 나왔다는 점이 특히 더 마음에 들었다. 내가 미래에 SMT solver의 타임 아웃에 고통받게 된다면 사용해보고 싶다.

발표를 돌아보며

발표들을 들은 후에 내가 느낀 점은 대략 두 가지이다: (1) 이런 방법으로 문제를 풀었구나. 근데 이 문제를 왜 해결해야 하지? (2) 꼭 필요한 문제를 해결했구나. 이 방법은 현실적으로 적용할 수 있을까?

나도 저런 연구를 하고 싶다는 생각을 들게 한 인상적인 발표는 후자의 느낌을 받은 연구들이다. 돌아보면 현재 내 Fault Localization 연구도 개발자의 사용 시나리오와 모델의 비용을 중점적으로 고려하면서 진행한 것 같다. 이번 학회에 참가하게 되는 기회를 얻으면서 이번 학회에서 내가 지향하는 연구의 방향은 무엇인가라는 질문에 대한 대답을 얻게 되었다는 생각이 들었다.

3 코펜하겐, 덴마크

독특한 여름 야경

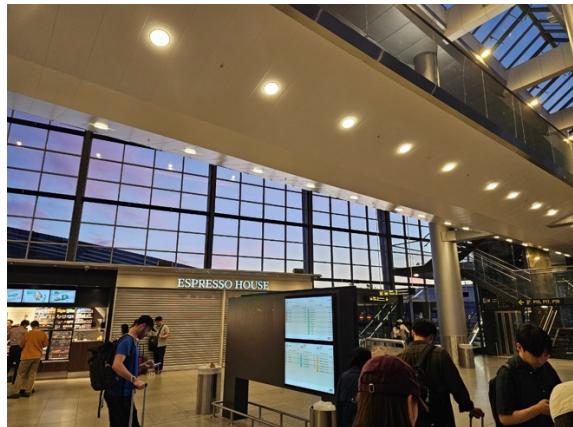


Figure 5. 카스트럽 국제공항, 22시 30분



Figure 6. 숙소 앞, 23시

코펜하겐 카스트럽 국제공항에 도착하여 북유럽의 시원하고 상쾌한 공기를 느끼며 창밖을 보는데 이게 무슨 일인가 싶었다. 시간은 분명 밤 10시 30분이었는데 하늘은 초저녁이었다. 고등학생 때 공부했던 ‘고위도 지역은 여름엔 해가 길고 겨울엔 해가 짧다’는 지구과학 기초상식은 나중에 기억났다. 덴마크는 공기가 맑으니까 별이 정말 잘 보일 것 같다는 내 막연한 상상은 실현되지 않았지만 개인적으로 서늘하면서도 약간 붉은 빛이 도는 덴마크의 밤하늘이 마음에 들어서 기회가 될 때마다 사진으로 남겼다

‘제주’스러운 식문화?



Figure 7. 치킨까스(?), 연어와 아보카도, 새우

출국하기 전 덴마크의 대표 음식을 검색해보았는데 죄다 호밀빵 밖에 없길래 먹을 건 별로 기대를 안하고 갔다. 직접 경험해본 바로 다른 유럽 음식을 먹어본 건 아니지만 원래 유럽 요리가 한국인 입맛에 짜다고 들어서 그걸 감안하면 음식은 맛있는 편이다. 요리가 단순해도 돼지, 소, 연어를 비롯한 육고기와 바다고기 모두 품질이 훌륭해서 그런 것 같다. 이런 점 때문에 묘하게 제주도 향토 음식이 생각나는데 공통점은 다음과 같다.

1. 쌀 이외의 곡물을 주력으로 함 (덴마카는 호밀, 제주는 메밀)
2. 돼지고기, 생선이 많이 생산되고 즐겨 먹음
3. 요리 방식이 단순해서 원재료를 거의 그대로 쓴

추천 여행지: 덴마크 국립 수족관



Figure 8. 덴마크 국립 수족관

코펜하겐에는 덴마크가 자랑하는 북유럽 최대 규모의 아쿠아리움인 덴마크 국립 수족관이 있다! 사실 우리나라의 씨라이프 부산 아쿠아리움보다 한참 작아서 크기를 기대하고 가면 실망한다. 대신 시설 디자인도 예쁘고 불가사리, 아기 상어 (종은 모르겠다) 심지어는 가오리까지 직접 만져보는 체험을 할 수 있기 때문에 색다른 체험을 하고 싶다면 한번쯤 가보는 것을 추천한다.

4 마치며

처음 해외 출국이자 학회 경험이었던 이번 덴마크는 뜻깊은 경험이었다. 글을 쓰고 있는 지금을 기준으로 내 처음 논문 제출이 한 달도 남지 않았는데 연구를 마무리 하는 과정에 있어서 이 경험이 자극과 원동력이 될 것 같다. 마지막으로 최고의 학회에 참석하는 귀중한 기회를 주신 오학주 교수님과 숙소와 일정 준비로 고생한 연구실 동료들과 선배님들께 감사드립니다.