



VMCAI AND POPL 2011

@ AUSTIN, TEXAS, USA

서울대학교 프로그래밍 연구실 이우석

(wslee@ropas.snu.ac.kr)

요약

이번 학회는 내가 두 번째로 경험하는 국제 학회였다. 지난번에는 서울대에서 열린 APLAS에 참가하였는데, 늘 있던 곳에서 늘 같이 지내던 사람들과 함께 학회에 참가했기 때문에 이번과는 많이 달랐다. 내 여정과 감상, 그리고 학회에서 들었던 발표들 중 인상적인 것들 몇 개에 대한 정보를 공유하려 한다.

미국에 도착

미국의 첫인상은 별로 좋지 않았다. 입국심사에서 나를 인터뷰하는 공항직원은 나를 매우 의심하는 눈빛으로 쳐다보며 이것저것 꼬치꼬치 다소 화난 어조로 물어보았다. 나중에는 학회 참가한다는 것을 못 믿어 학회 프로그램을 보여주고 나서야 나를 보내주었다. 면도를 못해서 험상궂어 보여서 그런 듯 하지만 나는 미국에 '돈을 써주러' 오는 사람인데 이런 대우를 받으면서도 미국에 들어가야 하는 상황이 다 국력차이 때문인 듯 하여 좀 우울해졌다.

텍사스 오스틴에는 밤 11시가 거의 다 되어서 도착하였다. 생각보다 날씨가 매우 따뜻해서 한국에서의 가을과 흡사했다. 시차적응 할 새도 없이 바로 다음날 아침에 학회가 시작되었다.



1 학회장소였던 호텔의 전경

발표들이 있었다. 준비 기간이 부족해서 많은 논문을 읽기 못하는 못했던 나로서는 적은 노력으로 시야를 넓힐 수 있으니 마냥 감사할 따름이었다. 당연하겠지만, 보통 이런 발표들은 호응의 정도가 어느 정도 청중의 전문성을 가정한 발표들 보다 훨씬 좋았다.

우리나라에서 이번에 논문을 두 개나 발표했는데 하나는 학주 형의 분석의 지역화(Localization)에 관한 발표고 다른 하나는 KAIST의 김세원씨의 문자열 분석을 요약해석의 틀 안에서 설명한 것이다. 두 분이 발표할 때는 한국인으로서 자랑스러웠다. 학주 형의 발표는 다소 긴장감이 느껴졌는데 끝까지 무리 없이 해내셨고 청중들 중에는 주의 깊게 들으면서 흥미로워 하는 사람들도 꽤 있었다.

그 외 흥미로웠던 발표들은 다음과 같다.

다음날 아침에 VMCAI가 시작되었는데, 생각보다 사람이 많지는 않았다. 대략 5~60명에 불과한 사람들이 등록을 한 듯 했다. Cousot 부부를 비롯해서 여러 유명한 사람들이 환담을 나누고 있는 모습을 보니 무슨 동화책 같았다. 발표장 분위기는 APLAS 때와는 사뭇 달랐다. APLAS때는 사람들이 자신이 하는 일과 직접 관련된 것들에만 관심을 보이고 그 외에 것에는 무관심한 양상을 보였고, 발표자들도 자신의 연구결과를 전달하는데 그렇게 열심이지 않았던 반면, VMCAI에서는 거의 모든 발표에 사람들이 관심을 보이면서 여러 질문들을 적극적으로 했고 발표자도 효과적인 전달에 매우 공을 들였다.

그리고 간간히 매우 직관적이고 쉬운 설명으로, 나같이 정적 분석 외에는 잘 모르는 문외한도 무엇을 했는지를 잘 이해할 수 있게 만든 훌륭한



Refinement-Based CFG Reconstruction from Unstructured Programs

– Sebastien Bardin, Philippe Herrmann, and Franck Vedrine

이 논문은 비구조화 프로그램(일반적으로 실행가능한 이진 프로그램)의 흐름 제어 그래프(CFG)를 만드는 새로운 방법에 대해 얘기한다. 비구조화 프로그램의 흐름 제어 그래프를 만드는 것은 이진 프로그램 분석을 위해 필요하다. 그런데 이런 프로그램들은 대개 동적 점프(dynamic jump)를 갖고 있고, 그 대상주소를 정적으로 알기 힘들어서 CFG를 만들기 어렵다. 이런 동적 점프는 효율성을 위해 컴파일러가 집어넣거나, 함수 포인터나 가상함수처럼 필요에 의해 들어간다. 동적 점프의 대상 주소를 알기 위한 값 분석이 정확치 않으면 매우 부정확한 CFG가 나오기 때문에 정확한 값 분석은 중요하다. 저자들은 가능한 주소들을 다른 요약 도메인으로 나타내지 않고 집합으로 나타내되(K-SET), 집합 원소의 최대 개수를 분석도중 필요할 때 마다 늘린 후 재 분석하였다. 이 때 집합의 크기는 동적 점프 지점마다 다를 수 있고, 처음에는 모두 0으로 시작하여 동적 점프의 정확성이 손실될 때 마다 커진다. 이런 방식으로 특정 프로그램 군들에 대해서 충분한 크기의 K-SET을 이용한 분석과 비교하여 정확도의 차이가 없거나 작게 하면서 CFG를 만들어낼 수 있었다. 이들은 분석이 다항시간안에 끝나고, 그 결과가 안전하다는 것 또한 증명했다.

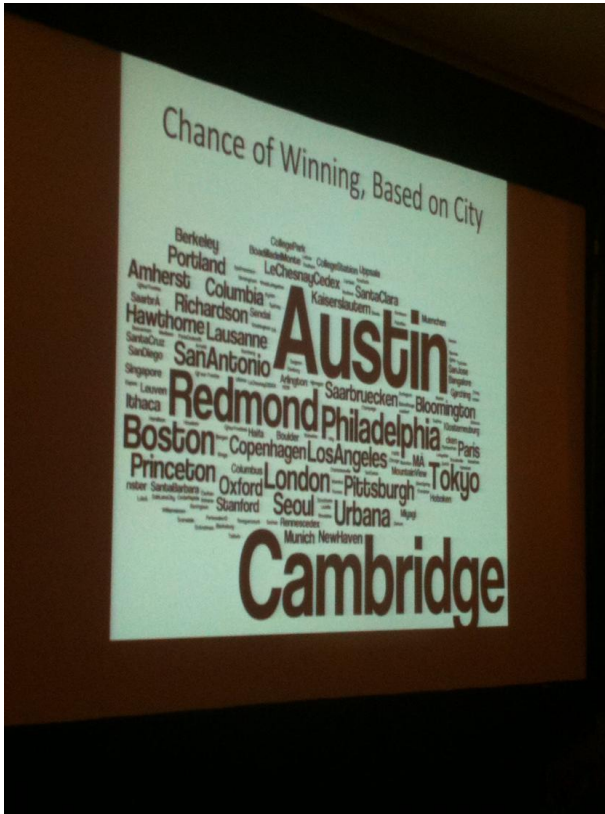
Practical Verification for the Working Programmer with CodeContracts and Abstract Interpretation (Invited Talk) – Francesco Logozzo

초청 발표이다. 사용자가 Code Contract 라이브러리를 이용하여 함수의 pre, post condition 및 기타 단언문(assertion)들을 넣으면 이를 검증해주는 정적 분석기를 소개하는 발표였다. 이 분석기는 .NET 프레임워크의 바이트코드를 대상으로 하기 때문에 C, C++, C#, Visual Basic 등 여러 언어에 대해 작동 가능한 것이 장점이다. 그런데 전역 분석(global analysis)이 아닌 함수 단위 분석(modular analysis)이고, 함수 간 분석(inter-procedural analysis)도 하지 않기 때문에 어려운 문제는 다 뻥 감이 있다. 그런데도 이 발표가 흥미로웠던 이유는 점점 개발자가 제약조건을 프로그램에 입력하고, 그 정보를 정적 분석이 이용하는 것이 앞으로 나아가야 할 한 방향이라는 목소리가 힘을 얻고 있고, .NET 프레임워크를 대상으로 언어 제한 없이 현재 사용되고 있기 때문이다. 복잡하지 않은 예제에 대해 데모를 보여주는 것조차 좀 뼈격거릴 정도긴 했지만, 최전선의 개발자들에게는 저 정도도 적지 않은 도움이 될 수 있겠다는 생각이 들었다.

POPL

VMCAI가 끝나고 POPL이 시작되니 분위기가 확 바뀌었다. 사람들이 갑자기 많아졌는데 등록자는 300명 가까이 되는 되어보였다. 사진은 참가자들 명단에서 출신도시만 뽑아서 워들로 만든 것인데, 서울도 아주 작지 않은 비율로 있다. 그리고 후원 회사가 많아서 그런지 전반적으로 프로그램

이 더 다채롭고 풍성해졌다. POPL은 받아들이는 논문 수가 너무 적어서 경쟁이 지나치게 과열되었다는 의견이 반영되어 이번 해부터 기존에 비해 두 배의 논문을 받아들였다. 대신에 학회가 길어지지 않게 하기 위해 서로 다른 두 개의 세션이 병렬적으로 진행되게 하였다. 그 사실을 알았을 때에는 사람들이 나란히 진행되는 두 세션을 번갈아 왔다갔다하면서 듣느라 정신없지 않을까 했는데 다행히 그런일은 없었다.



Verified squared: does critical software deserve verified tools?

- Xavier Leroy

초청 talk는 Xavier Leroy의 verification를 주제로 한 발표였다. Program Verification이 지금까지 어떤 문제들을 당면해왔고, 이를 어떻게 해결했는지, 그리고 현재는 어떤 상태에 있는지 매우 매끄럽게 설명하였다. 특히 슬라이드의 구성이 좋아, 발표를 많이 놓치고 슬라이드만 봐도 흐름이 잡힐 정도였다. 내용 중 인상 깊었던 것은, 5명이 2년 동안 검증된 컴파일러를 만들었는데, 성능이 gcc에 뒤지지 않는다는 것이다. 실용성을 조금도 포기하지 않으면서 소프트웨어의 신뢰성을 위해 고군분투했던 모습이 인상깊다. 또 Coq을 이용한 검증된 Ocaml 프로그램 생성에 대해서, 그럼 Ocaml 컴파일러는 어떻게 만냐는 어떤 사람의 질문이 기억에 남는다. 끝없는 의심과 회의로 프로그램은 점점 더 믿음직스러워 질 것 같다.

그 뒤 첫 번째 세션인 Pointer analysis는 나에게 있어 학회 기간 내 모든 세션을 통틀어 가장 인상깊었던 세션이었다. 이 세션에 속한 논문들은 모두 핵심이 통한다. 실제 분석에서 낭비되고 있는 부분에 대한 관찰과 이에 대한 공학적 해결이다.

Points-To Analysis with Efficient Strong Updates

- O. Lhotak, K. Chung

이 논문은 특히 시사하는 바가 많다. 이 논문은 포인터 분석에서 흐름 둔감(flow insensitive) 분석의 속도로 흐름 민감(flow sensitive)분석의 정확도를 얻는 것에 대한 것인데, 분석 과정에서 포인터 변수가 가리키는 대상이 한 개 인 경우가 많다는 관찰로부터 연구를 시작했다. 처음에는 흐름

둔감 분석으로 가다가 포인터 변수가 가리키는 대상이 하나인 것들에 대해선 흐름 민감 분석을 수행한다. 가리키는 대상이 하나면 효과적으로 strong update를 할 수 있고 흐름 민감 분석의 비용이 그다지 비싸지 않다. 그러다가 가리키는 대상이 여러 개가 되면 흐름 둔감 분석으로 대체한다. 이런 방식으로 정확도는 흐름 둔감 분석과 거의 차이가 나지 않는 반면 성능은 몇배 뛰어난 포인터 분석을 하는데 성공했다. 포인터 분석에서 포인터 변수가 가리키는 대상이 하나인 경우가 많다는 것은 많은 사람들이 다 알고 있는 사실이다. 그런데 이 관찰에 기인하여 직접 효과적인 포인터 분석을 시도해 본 사람은 그 동안 없었던 것이다. 반면 저자는 익히 알려진 사실과 간단한 아이디어를 직접 구현하여 좋은 결과를 얻었다. 구슬이 서 말이라도 꿰어야 보배라고, 온갖 아이디어를 생각해내면서 될까말까 재고 있기보다는 어느 정도의 확신만 서면 바로 행동을 해야 한다는 것을 확인했다. 발표도 아주 잘했고, 실험 결과도 실제적인 벤치마크 대상으로 수행한, POPL에선 보기도문 실험결과도 사람들의 관심을 끌었다.

Learning Minimal Abstractions - P. Liang, O. Tripp, M. Naik

이 논문의 핵심 아이디어가 재미있다. 원하는 분석에 필요한 최소한의 정확도로 효율적인 분석을 하는 것에 대한 아이디어이다. 분석의 정확도를 향상시키기 위해 더 정확한 요약물 사용하면 이것이 모든 부분에 적용되어 비용이 비싸지게 된다. 그런데 이러한 정확한 요약물이 우리가 원하는 답을 얻기 위해서 필요한 최소한의 것에 비해 지나치게 비싼 것일 수 있다. 저자는 기계학습 기법을 이용하여 문제를 풀기 위해 필요한 최소한의 요약만 구하는 방법을 고안했다. 이 아이디어를 k-CFA에 적용해보았는데 필요한 성질을 증명하는데 필요한 요약은 단 2% 였다. 원태 형은 이 논문의 결과 자체가 어떤 프레임워크로 기능하기 보다는 실제 분석에서 사용되는 요약 도메인의 원소는 구체적으로 얼마나 작았는지에 대한 관찰이 더 기여하는 바가 크다고 보았다.



오후에는 원태 형의 발표가 있었는데 다단계 언어를 단계 없는 언어로 변환한 후 그 언어에 대해 정적 분석을 함으로써 다단계 언어에 대한 정적 분석을 안전하게 할 수 있다는 내용이다. 원태 형은 이 발표를 위해 매일 4번씩 리허설을 했다고 한다. 그래서 말도 한번도 더듬지 않으시고 목소리는 침착하셨는데 쓰고 계신 레이저 포인터의 점이 사정없이 흔들렸다. 목소리는 침착하신데 떨리는 손이 그 긴장감을 보여주니까 웃음이 나왔다. 그래도 발표는 아주 성공적이었고 반응도 좋았다.

첫 날 모든 세션이 끝난 후에는 리셉션이 있었는데, 그 전에 간단한 다과와 맥주, 와인 등을 마시면서 사람들끼리 이야기하는 시간이 있었다. 그리고 흥을

돋우기 위해 재즈 밴드의 연주도 있었는데, 흥미로운 것은 재즈 밴드 구성원이 연구자들이란 것이다. 우선 General chair인 Tom ball이 색소폰을 불었고, 그 외 VMCAI/POPL 참가자들이 각각 키보드와 기타 베이스, 드럼 등을 맡았다. 연구도 잘하면서 취미도 열심히 그들이 멋져보였다. 또 이번에 후원사 중에 마이크로소프트가 있는 만큼, 마이크로소프트에서 야심차게 내놓은 Kinect를 직접 해보는 공간이 있었다. Kinect는 컨트롤러가 필요없는 플레이가 가능하게 하는 게임기인데 사용자의 움직임을 카메라로 인식한다. 게임기 홍보 겸 참가자들이 편하게 서로 친해지게 만들려고 한 배려 같았다.

둘째 날 가장 인상 깊었던 발표는 Gulwani의 발표인데, Input-Output example로 프로그램 합성을 하는 것이다.

Automating String Processing in Spreadsheets using Input-Output Examples - S. Gulwani

마이크로소프트 엑셀을 사용하는 사용자 대다수가 프로그래밍을 모르는 초보들인데, 보통 스프레드시트 포럼에서는 초보자가 입력-출력 예제를 올리고, 이에 대응되는 문자열 조작 스크립트를 고수가 올린다. 이에 착안하여 Gulwani는 입출력 예제로부터 문자열 조작 스크립트를 만들어내는 방법을 고안했다. 핵심 아이디어는 분할과 정복(Divide and conquer)인데, 원하는 문자열들을 만들어내는 프로그램을 만드는 문제를 서브문자열들을 만들어내는 프로그램을 만들어내는 문제로 나누어 해결한다. 알고리즘 개요는 다음 그림과 같이 나타낼 수 있다.

Goal: Given input-output pairs: $(i_1, o_1), (i_2, o_2), (i_3, o_3), (i_4, o_4)$, find P such that $P(i_1)=o_1, P(i_2)=o_2, P(i_3)=o_3, P(i_4)=o_4$.

Algorithm:

1. Learn set S_1 of string expressions s.t. $\forall e \text{ in } S_1, [[e]] i_1 = o_1$.
Similarly compute S_2, S_3, S_4 . Let $S = S_1 \cap S_2 \cap S_3 \cap S_4$.
- 2(a). If $S \neq \emptyset$ then result is $\text{Switch}((\text{true}, S))$.
- 2(b). Else find a smallest partition, say $\{S_1, S_2\}, \{S_3, S_4\}$, s.t. $S_1 \cap S_2 \neq \emptyset$ and $S_3 \cap S_4 \neq \emptyset$.
3. Learn boolean formulas b_1, b_2 s.t.
 b_1 maps i_1, i_2 to true and i_3, i_4 to false.
 b_2 maps i_3, i_4 to true and i_1, i_2 to false.
4. Result is: $\text{Switch}((b_1, S_1 \cap S_2), (b_2, S_3 \cap S_4))$

이 방법은 전 과정이 자동이고 거의 실시간으로 수행된다. Gulwani는 직접 Excel에 add-in으로 구현하여 발표 때 시현했는데 반응이 좋았다. 발표 초반에 자신이 무엇을 했다고 말하는 대신에

데모를 보여줌으로써 그가 뭘 했는지 모두가 아주 확실하게 알게 하였다. 동시에 청중의 집중을 이끌었다. 백문이 불여일견. 배울만한 발표 기술이라 생각했다.

모든 세션이 끝나고 저녁 늦게 학생 발표(student presentation) 세션이 있었는데 올해 새로 생긴 듯 했다. 몇몇 박사 졸업 예정인 학생들이 자신이 하고 있는 일을 말하면서 구직 홍보를 하는 모습이 눈에 띄었다.

마지막 날 아침 초청 발표는 최근에 돌아가신 Robin Milner라는 분을 기억하는 발표였다. 그는 최초로 증명 보조 도구(proof assistance tool)을 만들고 ML을 만들고, concurrency에 관한 일반적인 이론을 구축한 듯 하다. 발표장 분위기가 숙연해졌다.

모든 프로그램이 끝난 후에는 Xbox kinect 1개와 남은 프로시딩 6개를 제비뽑기하여 나눠 주었다. 여러모로 학회를 지루하지 않게 하기 위해 애쓴 노력들이 보였다.

느낀 점

이번 학회기간 동안 내가 가장 절실히 느낀 것은 세 가지이다.

1. 여러 분야에 관심 갖기
2. 영어 공부
3. 논문 되기

우선 여러 분야에 관심 갖는 것은 발표를 알아 듣기 위해서도 필요하지만, 사람들과 대화하기 위해 더 필요하다는 것을 느꼈다. 나는 아직 많이 부족하지만 조금이나마 아는 것이 정적 분석이다. 그런데 학회에서는 정적 분석 말고도 다양한 연구를 하고 있는 사람들이 오고, 그들과 대화하기 위해선 폭넓은 관심이 필요하다는 것을 느꼈다. 정적 분석보다 다른 분야들이 더 많이 회자되는 듯했다. 나는 별로 아는 것이 없어서 많은 대화를 할 수 있는 기회를 놓친 감이 없지 않다.

그리고 또 아무리 강조해도 지나치지 않은 영어 공부의 필요성을 절감했다. 수도 없이 듣는 얘이지만 그 필요성을 몸으로 느낀 것은 처음이었다. 발표 알아듣기가 힘들기도 했고, 발표자에게 질문하고 나서 답변 알아듣는데 애를 먹기도 했다. 길게 얘기하는 것을 알아듣는 능력이 필요함을 절감했다. 그리고 유창하게 말 못하는 이상, 말할 때는 정말 필요한 말들만 쉬운 단어를 써서 직접적으로 얘기해야 단번에 상대가 알아들을 확률이 높아진다는 것도 확인했다.

그리고 또 어서 논문 실적을 내야겠다는 생각을 했다. 나를 상대방에게 알리기 위한 가장 좋은 방법인 동시에, 쟁쟁한 사람들이 모이는 국제 저명 학회에서 주눅들지 않기 위해서도 필요하겠다 싶었다.

Austin, Texas



학회가 열린 오스틴은 생각보다 관광할 거리가 별로 없었다. 학주 형이 한국에서 가져오신 관광책자에는 텍사스에 가볼만한 도시들에 대한 소개가 있는데 오스틴만 없다. 그래도 학회 첫날 맛있는 것도 찾아보고 바람 좀 쐬어보려고 지난 여름부터 가을까지 우리 연구실에 와 있었던 Lucas를 가이드 삼아 탐방을 했다. 그나마 구경할만한 곳은 주 청사 건물(이곳이 텍사스 주의 주도이므로)과 텍사스 대학교 캠퍼스이다. 주 청사 건물은 밤에 아주 아름다운데 지어진 지 약 120년 된 건물이다. 텍사스 대학교도 꽤 역사가 깊은 학교라고 하는데 세련된 건물들과 큼직한 공간 활용이 두드러져 보였다.



처음 가본 맛집은 Tex-Mex 레스토랑이었는데 Tex-Mex란 Texas와 Mexico를 혼합한 단어이다. 텍사스 식 멕시코 음식을 파는데 맛이 약간 자극적이긴 했지만 꽤 괜찮았다. 치즈, 고기, 콩, 향료를 강하게 쓰는 것이 특징이라고 한다.

그 다음날 저녁에는 텍사스 대학교 근처 피자집에서 식사를 했는데 피자 스타일이 우리나라 것과 아주 흡사하다. 크기와 맛도 비슷하고 토핑이 풍부한 것도 똑같다. 그리고 한가지 깨달은 것이 있었는데 우리가 들어갔던 음식점, 바들은 모두 음악이 시끄럽게 틀어져있었다. 미국인들은 음악이 시끄럽게 나와야 밖에 외출한 기분이 드는 모양이다. 우리는 정신을 못차리겠는데 미국인들은 대화하는데 문제가 없다. 이런 것들 때문에 미국인들의 목소리가 일반적으로 큰 것 같다.

모든 학회가 끝나고 마지막 날 저녁에는 교수님께서 제안하셔서 허충길, 김익순 박사와 텍사스 대학교에서 박사 과정을 하고 계신 분과 박사 졸업하시고 IBM에서 일하고 계신 한국인 분들과 모두 같이 저녁식사를 했다. 스테이크 하우스를 갔는데 스테이크가 어찌나 두껍고, 또 칼은 어찌나 무식하게 생겼는지 마치 우리가 소 한 마리를 빙 둘러싸고 직접 썰어먹고 있는 듯한 느낌이 들었다. 고기는 두껍고 양도 많은데 퍽퍽함을 해소해줄 사이드 음식 없이 먹으려니 약간 괴롭긴 했지만 스테이크 맛은 매우 만족스러웠다 ☺



마치며

이번 학회는 내가 처음 겪은 국제 학회였다. 내가 확인했던 것은 목표로 해야 할 우리 분야 최고 학회의 실제 모습과, 상상 이상으로 세상에 똑똑한 사람이 정말 많다는 것, 학계를 미국이 거의 주도하고 있다는 것, 그래도 그 중에 한국의 활약도 만만치 않다는 것이다. 이번에 VMCAI, POPL을 통틀어 순수 국내 연구성과로 발표된 논문이 3편, 외국에서 활약하는 한국인이 발표한 논문이 2편이다.

POPL이 끝나면서 배부된 설문지에는 '환태평양 지역(Pacific rim(china, korea, japan, australia, etc))에서 학회가 열리면 참가하시겠습니까?' 라는 질문이 있었다. 우리나라 이름이 일본보다도 앞에 있는 것이 인상적이었다. 자랑스러웠지만, 한편으로는 내년 POPL도 미국 필라델피아에서 열린다는 발표를 들으면서 한 가지 다른 생각이 들었다. 엉뚱한 생각일지도 모르지만, ACM은 미국 컴퓨터 협회이고 엄밀히 말해 이 기관에서 열리는 학회는 미국 국내 학회인데, 전 세계 저명한 학자들이 이 학회에 기고해서 미국 학회의 위상을 더 드높이고 미국이 아닌 다른 나라와의 격차를 더 벌이고 있는 것이 지금 상황이라는 생각을 했다. SAS 같은 국제학회나, APLAS 같은 아시아 지역 학회 등, 미국 기관의 이름이 붙지 않은 다른 학회들도 지금보다 더 발전해서, 미국 밖 다른 나라들의 학계에 대한 기여도와 미국의 기여도가 지금 보다 더 균형잡힌 모습이 되었으면 하는 바람이 생겼다. 거기에 아주 약간이라도 일조하려면 지금보다 정말 더 열심히 해야겠다.