APLAS 2018 Trip Report

Jaemin Hong

1 Conference and Venue

I attended APLAS 2018—Asian Symposium on Programming Languages and Systems—from December third to fifth. It was held at Victoria University of Wellington in New Zealand. We stopped over in Tokyo and Auckland on the way to Wellington and in Auckland and Hong Kong on the way back to South Korea. Each trip took almost a day. Since New Zealand is in the southern hemisphere, it was fall and the temperature was between 10 °C and 20 °C. Although it was not cold, the wind was very strong so that we were hard to walk. Fortunately, it took only two minutes to get the venue of the conference from our hotel. The lunch given at the conference was worse than other conferences I attended before.

2 Keynote Talks

2.1 *Compositional Compiler Verification for a Multi-Language World* by Amal Ahmed

Ahmed's talk was about compiler verification. Compiler verification was a well-known topic but it was the first time for me to think about compositional one. I thought that defining the correctness of compilers when more than one language is involved is an interesting problem. Thus, the compositional compiler correctness theorem mentioned in the talk could be a valuable approach to the problem.

2.2 *Correctness in a weakly consistent setting* by Azalea Raad

Raad's talk was about weak memory consistency. The talk started with an explanation about the concept of weak memory consistency with some code examples. Since I already heard a few talks about the topic, it was easier to understand the concept than before. However, the remaining of the talk introduced lots of definitions and theorems and it was hard to follow all the details although she gave concrete examples. I thought that weak memory consistency is a difficult but important problem and it looks very beautiful that she expressed all the desired properties mathematically.

During the tour, I seated next to her on the bus. It was a really nice chance to discuss the talk but sadly I could not ask technical questions about the talk because I did not understand the detail. Instead, I said that I liked that her approach contains mathematical formalism for weak memory consistency and asked about her institute, MPI-SWS. Also, she asked a few questions to me and we discussed my future plan and Korean education system. It was a good experience to talk to the keynote speaker of the day.

2.3 *Souffle: A Datalog Engine for Static Analysis* by Bernhard Scholz

Scholz's talk was about static analysis via Datalog. Datalog is a logic language and static analysis can be expressed in logical relations. One problem to apply Datalog to program analysis was its performance so that his work suggested an improved Datalog Engine, Souffle. It was not an interesting technical approach for me.

3 Other Interesting Talks

3.1 *Scallina: Translating Verified Programs from Coq to Scala* by Youssef El Bakouny and Dani Mezher

Scallina was a tool to translate Coq code to Scala code. Coq is an automatic proof assistant. Programmers can implement their algorithms and verify the correctness of them in Coq. However, implementing a full program in Coq is inconvenient so that people usually implement only the core of a project in Coq and extract it to other languages like OCaml and Haskell. There was no support for extraction to Scala. The Scallina project targeted translating majority of Coq code to Scala type-safely using its rich type system. I was interested in participating in the project because my favorite language was Scala and I studied Coq. Hence, I asked about the project to Bakouny, the speaker of the talk, after the talk. He kindly explained his approach and gave me some related papers and the address of the GitHub repository of the project. Most of the project was already implemented and the translation mechanism seemed more trivial than I expected at first. Therefore, I planned not to contribute to the project.

3.2 *Traf: a Graphical Proof Tree Viewer Cooperating with Coq through Proof General* by Hideyuki Kawabata et al.

Traf was a visualization tool for Coq. It automatically created a proof tree for a proof script written in Coq. It did not involve any technical challenge but was a valuable approach since one big problem of Coq is the bad readability of proof scripts. I was inspired by the talk for a new idea. To deal with multiple similar cases simultaneously and to simplify trivial cases, Coq programmers use a bunch of tactical—a function-like-feature for proof scripts—and automation tactics including `auto`. However, especially for novices, how these features work is not trivial since Coq hides internal behaviors and shows only results. Therefore, I thought that a tool to visualize the internal behavior of Coq would be helpful for Coq novices and planned to study Coq internal and to implement the tool.

4 Student Research Competition

There was a poster presentation in the Monday evening and judges scored the presentation of each participant. Students with top three scores had a chance to give a 10-minutes-talk in the Tuesday morning and they were awarded the first, second, and third prizes.

I participated in the student research competition. The title of my work was *Path Dependent Types with Path-Equality*. I proposed an extension of DOT calculus, the core type system of Scala, by adding path-equality checking. I presented my work at Scala Symposium of the year as well. I really enjoyed the poster presentation. I discussed the features of Scala, the detail type-checking mechanism of my system, and related works with audiences. It was helpful for my future direction.

After the poster presentation, I became one of the top three scorers. It was a great honor, which I never expected. I was happy but had to prepare for the Tuesday's talk. Fortunately, since I already gave a talk at the Scala Symposium, preparing the talk was easy. The talk included some technical details and omits basic explanation about DOT because it targeted audiences familiar to Scala. Thus, I removed some details and added a brief explanation about DOT.

The Tuesday's talk was a little fast but quite successful. It was the first time to speak in front of more than fifty people in English but I already talked about the work multiple times so that I could be confident. At the banquet, I was awarded the second prize.

5 Miscellaneous Comments

Compare to PLDI and ICFP, which I attended before, the overall quality of the talks was bad. The keynote talks were great but most of the other presentations were boring. One problem was that the speakers did not have good English skills. Since it was the Asian conference, there were many Japanese researchers and most of them were not fluent in English. Moreover, it seemed that they did not prepare their talk much. They stuttered a lot during the presentations. The other problem was that their talks were too deep. Their slides looked like lecture notes. They contained lots of letters and formulas and the presentation directly dived into technical details without enough motivation and background information. In ICFP on September this year, many presentations considered a lot about the audiences and tried to explain their works easily even though the works are theoretical and mathematical. The style of presentations could be one of the characteristics of the conference but, still, it is hard to understand for the majority of the audiences except for some experts of the field related to the talk. Therefore, I could learn again about the importance of English skills and presentations considering audiences.

The poster presentation was a good experience to improve my research abilities. I highly recommend you, the students of I2R, to attend conferences actively, participating in a student research competition for instance, instead of simply listening talks of other researchers.