SPLASH RENO 2010

BRUNO OLIVEIRA'S TRIP REPORT

About the Conference

This was an interesting year for OOPSLA since it has undergone re-branding from OOPSLA to SPLASH (a re-arrangement of the OOPSLA letters, minus OO (because who programs with objects any more?), and appended with "for Humanity"). A main motivation for this change was that the conference wanted to broaden its scope to encompass essentially most of the Software Engineering and Programming Languages field. In recent years, this change has already been visible in OOPSLA's research programs and, with this change, it now becomes clearer that the scope is broader. In practical terms, this seems to have worked out well, as the conference has got 164 paper submissions (which has been a record number for the past 10 years). I believe that this change of scope is good news for the ROSAEC center because the traditional type of research done here (static analysis of C programs) now falls with the scope of SPLASH, which is one of the top-tier venues in our field. So, I'd like to invite everyone to consider submitting to SPLASH next year (specially since I'll be a PC member!).

Another interesting point is the recent trend in PL conferences to try to accept more papers. As Martin Rinard observed in his PC chair report, the quality of submitted papers has increased considerably in recent years in the PL conferences. Therefore, to promote faster dissemination of results it is important that papers do not have to go through several rounds of submissions just because there is a limit on the number of papers that the conference can accept. So, this year's OOPSLA has accepted a record number of papers (45), which is much in the same line with what other top-tier conferences such as POPL are doing (POPL accepted around 50 papers this year). I personally view this as a good thing for the reasons that Martin mentioned. A consequence of the increased number of papers, however, is that SPLASH also had parallel sessions for the first time, which some may argue is not so good because you cannot see all the paper presentations.

About the Venue

This year's SPLASH was at the Nugget Hotel in Reno. Reno is in Nevada, which is a US state where gambling and casinos are allowed by law. Therefore, like any other hotels in Nevada, the Nugget is also a casino filled with slot machines, blackjack tables and other gambling things. Most people that I've talked to didn't really like the venue. I guess they didn't particularly like gambling and the crowd that usually hangs out in casinos. Nevertheless, the obvious advantage was that accommodation was considerably cheap because a

Nugget Hotel/Casino

Slot machines

casino's businesses model is to sell cheap rooms with the hope that people will spend their money gambling later on.

The room was very spacious and comfortable, so I cannot complain about that. I've paid more in the past for much worse rooms.

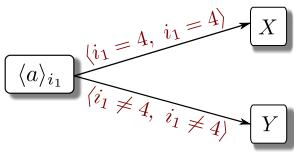
Technical Highlights

My own paper presentation was on wednesday 1:30pm. I believe the presentation went nicely and I had a few interesting questions from the audience; plus I had a some nice chats afterwards.

I'll try to describe some of the papers that may be of interest for the ROSAEC center that have also been presented in the OOPSLA technical sessions. In particular, I'll discuss the papers on this year's session on heap analysis.

Symbolic heap abstraction with demand-driven axiomatization of memory invariants

This paper by Isil Dillig, Thomas Dillig and Alex Aiken, was probably the one that caught my eye the most in this session. They propose an alternative to traditional relational analysis techniques, which keeps most of the precision offered by relational analysis, but it is much more scalable and efficient. The key idea is to enforce the memory invariant that every concrete memory location stores one unique value directly on the heap abstraction by adding constraints to the mapping between variables and the pointed locations. Essentially, the constraints are what ensures the unique value invariant. See the Figure on the side for an example. They claim that their technique is good for analyzing real-world programs with intricate use of arrays and pointers; in particular, they verify the absence of buffer overruns, incorrect casts, and null pointer dereferences in OpenSSH (over 26,000 lines of code). Maybe this is something worth checking out, since there have been a few people trying to use relational analysis.



Constraints

A dynamic evaluation of the precision of static heap abstractions

This paper by Percy Liang, Omer Tripp, Mayur Naik and Mooly Sagiv conducts a study on static heap abstractions with the goal of investigating how various refinements of allocation sites can improve precision. The abstractions considered keep track of the following information: call stack,

object recency and heap connectivity information. They used the 9 Java programs of the DaCapo benchmark. Some of their conclusions are that for abstractions based on k-CFA the critical value for k in which the precision no longer benefits from larger values of k is between 3 and 6. A digression regarding this point is that someone in the audience commented that this is not very helpful, because those values of k are still infeasible to be used in practical analysis. The authors agreed with this, but pointed to some paper that has been accepted at POPL this year (I believe this paper is Learning Minimal Abstractions) where apparently they devise a technique which, instead of using a fixed k for all points in the analysis, picks higher values of k only at certain points, retaining most of the precision of an analysis using an higher value of k, but being much more efficient. Ending the digression, some other conclusions seem to be that recency is an important dimension that offers the best tradeoff between precision and size.

Parallel inclusion-based points-to analysis

This paper by Mario Méndez-Lojo, Augustine Mathew and Keshav Pingali suggests a way to parallelize inclusion-based points-to analysis. The authors observe that inclusion-based points-to analysis can be formulated entirely in terms of graphs and graph rewrite rules, which exposes the data-parallelism in the algorithm for the analysis. They claim that their parallel implementation achieves speed ups up to 3x on a 8-core machine with ten large C programs. Yannis Smaragdakis was apparently not very happy with their claims because he mentioned that his work on Strictly Declarative Specification of Sophisticated Points-to Analyses (presented at OOPSLA'09) already implied such result. In that work, Martin Bravenboer and Yannis, developed a framework for points-to analysis where ponter analysis algorithms were declaratively specified using Datalog, which allowed for agressive optimizations (I believe algorithms in Datalog can naturally be parallelized due to Datalog's declarative nature).