

ECI 2013 - IGERT Report
by Felix GREZES

ECI is a computer science event held at the Universidad of Buenos Aires in Argentina. Over the course of one week the students can follow up to three courses, choosing from nine covering a very wide range of topics, ranging from the very theoretical such as game theory to the more practical like software engineering. My personal choices reflect this diversity as I opted for a subject that I enjoy in Pr. Milos Stojakovic's Positional Games class, and in a more practical one in Pr. Gordon Pace's Runtime Verification course. The diversity also showed in the teachers' origins, with Pr Stojakovic hailing from Serbia, and Pr. Pace from the island of Malta. In my classes, a handful of other students were foreigners like me, however they came from Uruguay, Bolivia, and other Latin-American countries. I was the only student who did not speak Spanish, however the courses were in English, and everyone spoke it well.

Notes on Buenos Aires, the UBA, and the trip that were not in my diary:

Life in Buenos Aires is just like life in any other major city. If you have ever lived in one (and odds are the person reading this is living in New York City), then it should not be difficult adapting. There are of course many tips and tricks to learn, and sometimes unpleasant surprises, but these should never be reason enough to dissuade someone from travelling to Buenos Aires. Tourists are not harassed to buy stuff, personal safety was never an issue despite the usual warnings from fellow travelers. I was personally fine perusing the city alone, but traveling in groups might be more comfortable for some people.

I didn't have the chance to visit much of the UBA campus, as it's a bit spread out. It's not as glamorous as some US campuses, but very similar to French ones which I'm familiar with.

Notes on Runtime Verification:

Runtime verification is the idea of verifying the correctness of a program during its execution, checking if the program conforms to its specifications. I chose this course as an introduction to software engineering, my background being in math and theoretical computer science. While we did cover some of the theory behind runtime verification, the most productive sessions were the practical ones. We worked in Java on a makeshift financial system, started with the more simplistic methods that are little more than rewriting the same code twice, then moved on to the more abstract, powerful and sound tools such as JAspect and LARVA, a logic and rules based verification tool.

Since I am not a software engineer, my expectations for the content of the class were bound to be inaccurate. Runtime verification is less about catching bugs in the program than it is ensuring that it does not do anything undesirable, such as allowing spending money to be spent twice for example. In a sense, we aim to catch design bugs rather than programming ones, though the line between the two is obviously not clear cut. For my own research, I usually work in the Python programming language, and many runtime verification tools are available for it. I don't usually write large, mission critical software, but I still find it very useful to know about these kind of tools and techniques, especially when I am working as part of a team. Pr. Pace was also quite a good pedagogue, very involved in the course and it was a pleasure to follow his course.

Note on Positional Games:

Positional Games is a subcategory of game theory focusing on two player games played on a hyper-graph, the abstraction of the board to play one. Examples of such games tic-

tac-toe, connect four and Shannon's switching game. I personally enjoy this type math a lot, and even though it might seem unconnected to any other research I do, I find that learning about all kinds of different proof methods, new ways of thinking, actually help me approach my own problems more efficiently. It also allows me to stay up to date on the research in other fields, preventing me from locking myself in a small field. Some of the most diverse mathematicians of the 20th century, such as John Nash and Paul Erdos, have worked on the topic of positional games.

Pr. Stojakovic covered the three main types of positional games: strong games, avoider-enforcer games and maker-breaker. One proof method that was completely novel to me was Erdos' probabilistic proof. The classes were sometimes difficult to follow, as theoretical math classes can be, but it was also refreshing and entertaining.