Interview with Rajeev Alur and David Dill 2016 Alonzo Church Award Recipients

Luca Aceto
ICE-TCS, School of Computer Science,
Reykjavik University
luca@ru.is

Rajeev Alur (University of Pennsylvania USA) and David L. Dill (Stanford University, USA) are the recipients of the 2016 Alonzo Church Award for Outstanding Contributions to Logic and Computation for their work on timed automata, a decidable model of real-time systems that combines beautiful, new and deep theory with widespread practical impact.

In particular, Alur and Dill are honoured for their paper "A theory of timed automata", Theoretical Computer Science 126(2):183–235, 1994. Fundamental Study. http://dx.doi.org/10.1016/0304-3975(94)90010-8. That paper proposes a variation on classic finite automata to model the behaviour of real-time systems. The resulting notion of timed automaton provides a strikingly simple, and yet powerful, way to annotate state-transition graphs with timing constraints using finitely many real-valued clocks. The fundamental study by Alur and Dill developed the model of timed automata as a formalism for accepting languages of timed words, studied the model from the perspective of formal language theory by considering closure properties and decision problems for the full model and some of its sub-classes, mapped the decidability boundary for the considered decision problems, and introduced a fundamental abstraction, the so-called *region graph*, that has since found application in virtually every decidability result for models of real-time and hybrid systems, and that is at the heart of coarser abstractions that are embodied in the verification engines of several industrial-strength tools for automatic verification of real-time systems.

In the twenty five years since their invention, timed automata have become the standard model for the analysis of continuous-time systems, which underlies hundreds of papers, tens of tools, and several textbooks.

In order to celebrate the award of the Alonzo Church Award to this hugely influential work and to give the theoretical computer science community at large a glimpse of the history of the ideas that led to it, I interviewed Rajeev Alur (abbreviated to RA in what follows) and David Dill (referred to as DD in the text below) via email. I hope that the readers of the Bulletin of the EATCS will enjoy reading the text of the interview and will find it as interesting as I did.

The interview

LA: You are receiving the 2016 Alonzo Church Award for Outstanding Contributions to Logic and Computation for your invention of timed automata, which, as far as I know, was the first decidable formalism for reasoning about the ubiquitous class of real-time systems. Could you briefly describe the history of the ideas that led to the development of the formalism of timed automata, what were the main inspirations and motivations for its invention and how timed automata advanced the state of the art?

DD: The interesting part of the story for me is the inspiration that led to the question in the first place and the initial results. Things worked out so magically that I'm still amazed, so I hope that the story will be interesting to others. I also learned a lot of lessons about research, which I hope I can convey here.

I am not a logician and don't consider myself a theorist. I generally want to do research that has practical impact, but I like being able to apply theory to help with that. And, in this case, I ended up creating some theory as part of that process.

My PhD research under Ed Clarke at CMU was on using finite automata for formal verification of speed independent circuits. I started working on them using CTL model checking, and then decided during my thesis work to abandon model checking and used finite automata (I am grateful that Ed accepted this change in direction, since his primary research agenda was CTL model checking). Speed-independent circuits are digital circuits with no clocks, which are supposed to meet their specifications for all possible gate delays. Ed had recently co-invented CTL model checking and was exploring speed-independent circuits as an application, because speed-independent circuits are among the simplest concurrent systems with bugs. But speed independence is a very conservative model, because engineers often know some constraints on delays, even if they can't specify exact values for the delays. A circuit designer (Prof. Chuck Seitz of CalTech) had some ways of designing speed-independent circuits that relied on timing constraints without precisely specified delays, and asked me whether I could prove such circuits correct.

I looked at the literature, and there were a number of people who had used finite automata with a special alphabet symbol representing clock ticks, and they would simply count the clock ticks using the states of the automaton (what was later called a discrete time model). But I wasn't comfortable with that, because asynchronous circuits don't have clocks! I felt that events in circuits happened in continuous time, which might be different from discrete time – but I didn't know.

While I was working on my PhD, I sat down to try to work out a method to verify timed asynchronous circuits. The research was painless compared to a lot of my other PhD work. I imagined that each gate had a timer that was set to a real value between constant delay bounds, and that these timers decreased continually with time until the reached 0, at which point the gate output would change. Very quickly, I came up with an algorithm that kept track of the set of all possible timer values for various discrete states, and, magically, the analysis of the regions could be easily solved using the all-pairs shortest paths problem. I think I just got lucky and happened to think about it the right way, so it was easy to do. (Later, I learned that Vaughan Pratt had observed that certain systems of linear inequalities could be solved using shortest paths, and that a similar algorithm was used in a kind of timed Petri nets – but I didn't know that at the time).

I left this method out of my PhD thesis, because my thesis seemed coherent and the timed model didn't seem to mesh with the other material. Then, in my first few years at Stanford, I pulled it out again and tried to prove that it was actually sound. Getting the details right was harder than working out the original idea. It took several weeks. The paper eventually appeared in CAV '89.

Verification with linear time (as opposed to branching time) models seemed beautiful to me because all the problems reduced to standard closure properties and decision properties that had been solved for finite automata: Closure under intersection, complementation, and projection, and the decidability of language emptiness. Implicit in my CAV paper were closure under intersection, projection, and the decidability of emptiness for timed regular languages, but I couldn't prove closure under complementation. I felt very strongly that timed regular languages were closed under complementation and would work the same way as conventional finite automata.

Then Rajeev Alur walked into my office. He was a second year student who was ready to do some research, and his research advisor, Zohar Manna, was out of the country for a while. I explained timed automata to Rajeev and asked whether he could resolve the question of whether they were closed under complementation. Rajeev very quickly came up with a nicer definition of timed automata, with clocks that counted up and predicates, and an "untiming" construction for deciding language emptiness that had lower complexity than mine. I remember it took him several months to prove that, in fact, timed automata were not closed under complementation and that, surprisingly, the universality problem was undecidable. He proved several other amazing results, and then we wrote the "Timed Automata" paper. The paper was rejected twice from conferences with pretty harsh reviews (I think FOCS was one conference — I don't remember the other one) and was eventually accepted at ICALP.

I'm not sure why it was so poorly received. I don't remember changing it much when we resubmitted it. I think there is a problem when you formulate a new problem and solve it, but reviewers don't understand why the problem is important. If the solution is a bit difficult, they look for reasons to reject the paper. If you attack a well-known problem so everyone knows why you want to solve it, it's sometimes easier to sell the paper (but maybe it won't have as much impact). Or maybe we just had some bad luck — everyone gets bad reviews, and I've unfortunately written some bad reviews myself.

RA: I joined Stanford University in 1987 as a graduate student primarily interested in logic and theory of computation. Topics such as model checking, temporal logics, and automata over infinite strings were hot topics then, and extending these formalisms to reasoning about real-time systems was a natural research direction.

For me, the key starting point was Dave's wonderful paper titled "Timing assumptions and verification of finite-state concurrent systems" that appeared in CAV 1989. Dave has already explained how he arrived at the idea of difference bounds matrices that appeared originally in his paper, and I will explain two new ideas in our follow-up work aimed at generalizing the results. Dave's original paper modeled timing constraints by introducing timers each of which could be set to a value chosen nondeterministically from a bounded interval, decreased with time, and triggered an event upon hitting 0. We changed the model by introducing clock variables each of which could be reset to 0, increased with time, and could be tested on guards of transitions. In retrospect, this led to a simpler model, and more importantly, led naturally to many generalizations such as hybrid automata. The second idea was the definition of region equivalence that defines a finite quotient over the infinite set of clock values. This proved to be a versatile tool and forms the basis of many decidability results, and in particular, algorithms for model checking of branching-time temporal logics.

LA: According to Google Scholar, the journal paper for which you receive the Alonzo Church Award, which was published in 1994, has so far received over 6,500 citations, whereas the ICALP 1990 on which it was based has been cited over 1,250 times. A Google Scholar query also reveals that at least 135 publications citing your landmark TCS paper have more than 135 citations themselves. Moreover, the most successful and widely used tools for modelling and verification of real-time systems are based on timed automata. When did it dawn on you that you had succeeded in finding a very good model for real-time systems and one that would have a lot of impact? Did you imagine that your model would generate such a large amount of follow-up work?

DD: It will be really interesting to hear what Rajeev says about this.

At the time, I just felt happy that we had a nice theory. I was looking around for other uses besides asynchronous circuits to which timed automata would be applicable, and, based on the name, "real-time systems" seemed like a good candidate. But I didn't know much about the practical side of that area, and timed automata didn't seem to be directly applicable to many of the problems they worried about (there were a **lot** of papers on scheduling!).

It took a very long time before I was confident that timed automata were useful for anything, including real-time systems. So, for me, it was based on feedback from other people. I got some grant money to work on the problem, which was a good sign. People from control theory and real-time systems invited me to come and talk to them about it. It may be that timed model checking (which is very closely related) has had more practical impact than timed automata themselves. Other people built tools that were better than anything I created.

After a few years, the area got too hard for me. Others, especially Rajeev, were doing such a good job that I didn't feel that my efforts were needed. So, I moved on to new topics in formal verification and asynchronous circuit design. Now I only have a fuzzy idea about the impact of timed automata, embarrassingly.

RA: The initial reception to our work was not enthusiastic. In fact, the paper was rejected twice, once from FOCS and once from STACS, before it was accepted in ICALP 1990. There was also a vigorous argument advocated by a number of prominent researchers that modeling time as a discrete and bounded counter was sufficient. By mid 1990s though the model started gaining acceptance: in theory conferences such as CONCUR, ICALP, and LICS a steady stream of papers studying complexity of various decision problems on timed automata and extensions started, and a number of implementations such as KRONOS and UPPAAL were developed. However, I could have never imagined so much follow-up work, and I feel very grateful to all the researchers who have contributed to this effort.

LA: What is the result of yours on timed automata you are most proud of? And what are your favourite results amongst those achieved by others on timed automata?

DD: I'm most proud of coming up with the question and the initial (not so hard) results. To me, the question seemed so completely obvious that I couldn't believe it hadn't been answered. I contributed to some of the later results, but Rajeev took off like a rocket and I couldn't keep up with him. At some point, there was so much work in the area that I didn't feel I had much to add. I know there are a lot of amazing results that I haven't studied.

I like the fact that timed automata formed a paradigm that others followed with hybrid automata and so on. The idea of extending finite automata with other gadgets, like clocks, and analyzing the resulting state space seems to have influenced people as much as the results on timed automata.

RA: There are lots of strong and unexpected results that I like and it is hard to choose. But since you asked, let me point out two which are closely related to our original paper. The paper by Henzinger et al. titled "Symbolic model checking of real-time systems" (LICS 1992) introduced the idea of associating invariants with states. I think this is a much cleaner way of expressing upper bounds on delays and, in fact, is now part of the standard definition of timed automata. In our original paper, we had proved undecidability of timed language equivalence. Cerans showed in 1992 decidability of timed bisimulation which I did not expect and involves a clever use of region equivalence on product of two timed automata.

LA: Twenty five years have passed since the invention of timed automata and the literature on variations on timed automata as well as logics and analysis techniques for them is huge. Do you expect any further development related to theory and application of timed automata in the coming years? What advice would you give to a PhD student who is interested in working on topics related to timed automata today?

DD: I'm sorry, but I haven't actively pursued the area and I just don't know. If I were giving advice to a younger version of me, knowing my strengths and weaknesses, I would actually advise that person to go and find a new problem that hasn't been worked on much. Maybe you'll start a new field, and you'll get to solve the first problems in the area before it gets too hard. What has worked for me was looking at an application problem, trying to find a clean way to formalize it, and then looking at the problems stemming from that formalization. It's an amazing fact that there are fundamental questions that haven't been addressed at all. Starting with a practical problem and then thinking about what theory would be helpful to solve it is a good way to come up with those questions. Also, watch for cases where existing theory doesn't exactly fit. Instead of trying to pound nails with a wrench, imagine a more appropriate, but simple, theory and invent that.

RA: I feel that theoretical problems as well as verification tools have been extensively investigated. Given that, my advice would be to first focus on an application. When one tries to apply an existing technique/tool to a real-world problem, there is invariably a mismatch and that insight can suggest a research idea. The emerging area of cyber-physical systems is a rich source of applications. For instance, formal modeling and analysis of medical devices can be a fruitful direction.

LA: You have both been involved in inter-disciplinary research with colleagues from biology and control theory. What general lessons have you learned from those experiences? What advice would you give a young researcher who'd like to pursue that kind of research?

DD: On reflection, my research in formal verification was not as collaborative as it could have been. But in computational biology, if you don't have a wet lab, you **really** have to collaborate with other people! Collaboration can be rewarding but also frustrating. My advice would be: learn as much about the other area as you can, so you can at least talk the same language as your collaborators. And prepare to invest lots of time communicating and understanding what motivates them – and make sure they're willing to do the same, otherwise, it's a collaboration that is not going to work out. What keeps you working together is **mutual** benefit.

Most collaborations don't work out. If you have a good collaborator, be grateful. And, sometimes, you may want to choose among research directions based on which ones have the best collaborators.

LA: David, Rajeev, many thanks for taking the time to share your views with the members of the TCS community and congratulations for receiving the first Alonzo Church Award!