

## Bruno Courcelle

It is always risky to talk about the work of a scientist and to try to explain how and why he came to solve this rather than that problem.

I attempt to do that today on a very special occasion for several reasons:

–It is the 40<sup>th</sup> anniversary of the European association for Theoretical Computer Science: the first colloquium organized by this association was held in Paris in July 1972 under the name of ICALP, international colloquium on Automata, languages and programming. The series of ICALP has been going on since then, the next one will be held in Warwick soon.

–Bruno Courcelle is retiring and Bruno is a very special fellow in my mind: we have known each other for 40 years, he attended the Paris ICALP I already mentioned and I hired him as a researcher at IRIA shortly afterwards. He was of my three first students to get his doctor's degree, doctorat d'Etat as it existed at that time, with Luc Boasson and Irène Guessarian. We have been friends since then and he asked me recently to write a foreword to the huge book he coauthored with Joost Engelfriet. The book has just appeared, published by Cambridge University Press.

Obviously I have to recall what Theoretical Computer Science looked like forty years ago. Before doing that I wish to recall the memory of Philippe Flajolet whom I hired at IRIA nearly at the same time as Bruno and who died unfortunately sixteen months ago. Philippe was nearly the same age as Bruno, and also wrote a huge book, together with Roger Sedgewick, which gathers the ideas and results of a lifework.

Two theories really existed in 1970 which could be taught without shame, with nice theorems lying on good definitions and axioms:

–Recursive function theory had been built by logicians and there was a very good reference book written by Stephen Cole Kleene on the subject: in France the most knowledgeable people were Daniel Lacombe and Jean –Louis Krivine.

–Automata theory which was younger, may be the earliest founding paper, by the same S.C.Kleene, was published in 1956 by Princeton University Press in a collection of papers entitled Automata Studies. Marcel Paul Schützenberger was a pioneer of this theory and already famous, in the world, for his joint work with Noam Chomsky. Besides the theory of finite automata and regular events (which we called in France “langages rationnels”) the linguist Chomsky had proposed a definition of context-free languages by means of grammars, and these languages had been proved to coincide with the languages recognized by the so-called push-down automata( we called these languages “algébriques” and the corresponding PDA's “automates à pile”).

I shall not say more about recursive function theory as I am not a logician and have never worked myself in that area. But I can talk for a long time about the early years of automata theory and I believe that the state of the art in automata theory can explain the first steps of Bruno when he started doing research. Not the very first

steps for he started working with Jean-Louis Krivine on the famous theorem of Cohen on the independence of the axiom of choice and the method of “forcing”.

Bruno was not convinced by the application of Cohen’s method to cumbersome problems in set theory and started looking for some other subject to tackle and he read the “Gross et Lentin” a very nice green book which had been recently written by two people I have known very well Maurice Gross and André Lentin. The book “Notions sur les grammaires formelles” was well written and contained nearly all what was known at that time (1967) on formal grammars and the corresponding recognizing automata, it was an attractive book which has attracted several young men and women to the two fields of automata theory and formal linguistics.

Bruno was seduced and he came to see me, not because I was famous but because the number of people with a doctor’s degree, able to serve as a director when writing a thesis in theoretical computer science was extremely small, in France in general and, particularly, in Paris. MPS, who was famous, directed already a dozen students, he had directed my own dissertation completed in 1967, and after the “events” of May 1968 had decided not to take any more student (he really did it!)

The fact is that, though captivating, automata theory was a mess. Apparently easy problems were still to be solved, no one knowing how to handle them, and indeed some of them took twenty years or more to find a solution. There were fashionable theories which I never understood such as the Krohn-Rhodes decomposition theory (despite the fact that I have spent time with John Rhodes and he patently tried to explain it to me). Being older and more experienced than the other participants in MPS’s seminar, in IHP in Paris, I understood may be one third of what he was saying and no more.

Ideas were coming too fast and they were spreading before they had been properly worked out, the terminology was not fixed and the same idea could appear in so different forms that it was difficult to recognize it. The appearance in 1960 of the high level programming languages which were Algol, FORTRAN and Lisp had been the starting point of a big rush towards compilation methods: the goal was not only a scientific one but also à financial one since it was obvious that the computing machines (as we called them at that time) which would win on the market were to be those which have the best, most efficient and more reliable compiler. Both engineers and researchers started developing methods and comparing them using benchmarks: only a few among these people were taking the time to worry about the soundness, or well-founded ness of their methods which were presented usually as recipes. Symposia and colloquia in which engineers and researchers met were a bit strange, everybody claiming that his method was better without providing any clear and precise argument to sustain his claim. A good thing was that engineers and university researchers did meet since the need for ideas in order to build compiler, to improve them or simply to understand what strange features of the programming languages was enormous, a bad thing was that the details of industrial compilers were kept secret for obvious reasons of concurrency.

It happens that the syntax of Fortran was described in the so-called Backus normal form, BNF (from the name of John Backus who, according to The IBM company, is the author of Fortran), and BNF looks pretty much like what Chomsky called context free grammars and was intended, in his mind, to describe the syntax of natural languages. And many people started to invent a number of devices or algorithms to perform the syntactic analysis of both sentences in natural languages and programs in Algol or Fortran which are sentences in these languages; but it was very unclear what should be the output of this syntactic analysis (in both cases of natural and programming languages), the notion of syntactic tree did not stem out immediately and it took some time before it appeared to almost everybody as the right object to represent the structure of a sentence (in both cases). Any how the output of the syntactic analysis phase of compilation serves as the input of the next phase which is code generation and it was tempting to try several ways of describing the structure of a program leading to different code generation mechanisms. As it was tempting to restrict the power of grammars in order to get better algorithms, in order to perform the compulsory syntactic analysis. The models of grammars were flourishing, all alterations of CFG's, and several models of automata to recognize the generated languages were also introduced which were alterations of pushdown automata, PDA's, or deterministic pushdown automata, DPDA's with the aim to obtain a powerful enough grammar and a correspondence between the grammar and the automaton more efficient and clearer than the awkward, hard to prove, equivalence between the set of languages generated by CFG's and the set of languages recognized by PDA's.

Here I wish to say a word of Samuel Eilenberg who was a very famous mathematician. I do not know how it happened Sam spent a year in Paris in the Institut Blaise Pascal where was MPS and where I had started to do computer science in 1959 with Louis Nolin. Sam was a good friend of Marco but thought that all the literature on languages and automata was sloppily written, including all Marco's papers. He was certainly right and I am thankful to him, for during that year, 1967, I learned a lot from him in order to improve the presentation of my results. Sam undertook to rewrite in his own way, as mathematics should be written. I was present at many discussions between Sam and Marco in front of a blackboard which were often violent but obviously fruitful. The theory of varieties of languages and the correspondence between varieties of languages and varieties of automata was born from hours of discussion in the small classroom of the institute and it is a nice one though it was not as fruitful as their genitors thought and claimed. Eventually Sam wrote two books, Automata, Languages, and Machines part A and B, which appeared in 1974 and 1976 on finite automata and the (rational, regular) languages which are recognizable by finite automata giving a clear mathematical treatment of the subject (not all, the non-deterministic FA's are not really treated). He had planned other volumes, especially a volume C which was to be dedicated to CF languages and PDA's: but he never wrote it. The many discussions I had with him in 1967, when I was writing my dissertation on CF languages (submitted in December 1967) convinced me that the book was not written because Sam could not write it, the time was not ripe for it and, may be, it is still not ripe today (despite the beautiful work of Géraud Sénizergues who is here and his long intricate proof of the

decidability of the equivalence of two DPDA's which was honored by the Gödel prize granted to him in 2002).

I know I am rather long but I wish to describe as faithfully as possible the state of the art of TCS when Bruno started working as a young researcher. And the picture is not yet complete. If the syntax of high level languages was described in a formal and precise way using some sort of grammar their semantics was described in plain words which lead to many possible interpretations (or as the "own" variables in Algol could not be given a precise meaning) and the very nature of a program was a matter of endless and inconclusive discussions. The idea that it is a functional in an appropriate computation space and that the computed function was the fixed point of this functional was not yet there or rather it took form around 1970: I heard first about it in 1969 in a meeting of the working group 2.2 of IFIP where I met a number of people who played an important role in my life and , by transitivity in Bruno's life. When Dana Scott was developing his fix point semantics in ordered functional spaces, Mike Paterson was working on recursive program schemes, after submitting his dissertation on this subject to Warwick University under the direction of David Park. The idea is quite simple: forgetting about the meaning of the function symbols which occur in a program, one consider the program as a grammar generating something which happens to be an infinite tree which is rational if the program is a non recursive one, and an algebraic tree if the program is recursive. Giving a meaning to the function symbols is giving an "interpretation". This way of considering programs was extremely appealing to me for my work had been until then entirely dedicated to CF grammars and CF languages.

1969 is the year of my election as a professor in Paris University, and shortly afterwards MPS who was one of the four founding fathers and directors of The newly created IRIA asked me to work part time there as a "chief of project". These gave me possibilities of hiring some young researchers and also to organize meetings, schools, and conferences given by foreign scientists whom I could invite.

My enthusiasm for the beginning formal semantics whose existence I had discovered in Colchester was such that my project was dedicated to the verification of properties of programs and certainly when we started talking, Bruno and myself, about what he was going to do in IRIA and as a subject for his future doctoral dissertation one of the first problems we talked about was recursive program schemes. The project was to use some of the half cooked ideas and some of the awkward methods we had about CF grammars to say more about RPS's and the algebraic infinite trees which corresponded to them, and were undoubtedly new objects whose theory had to be developed from the very beginning. Several people were already at work in the world on similar questions and among them Joost Engelfriet who gave a talk at ICALP 1972 on RPS's and CF-grammars. Completely different ideas concerning the verification of properties of programs were developed elsewhere and a young Frenchman, Jean Vuillemin, came back at INRIA, I believe also in 1972, with a PhD from Stanford University prepared under the supervision of Zohar Manna : less algebraic than mine these ideas were mainly borrowed from

classical proof theory in mathematical logics. Jean certainly influenced Bruno in these crucial years when Bruno was finding his own way to science.

My memory of this period, 1972 to 1976, when Bruno completed his dissertation, is that we formed already a sizable group, all very young, I was the oldest, 35, surrounded by several students working towards a doctor's degree, in fact I had two groups of such students, those who worked in "pure" automata and languages theory and those working more in semantics, rather close to Bruno, Irène Guessarian, Gérard Boudol, Laurent Kott, Guy Cousineau, Jean-Jacques Lévy, Gérard Berry. We communicated very often with other researchers who were all very young too, around 25, working both at IRIA and at the university Paris 7 and a number of also young researchers whom, thanks to IRIA we could invite or visit. And this situation was close to paradise: we were all enthusiastic, we were all working hard, we were all sharing freely our own ideas, and the ideas we could grasp from discussions with others or from our reading of hardly readable preprints and reports coming from the United States or elsewhere. Weekly meetings could last for hours since the ideas were incoming very fast and we had to discuss them all. And no one was there to "evaluate" what we were doing, nor even ask a question about our goals, question we would have been unable to answer for, indeed, we did not know at all where this intense activity would lead us to! We were happy, our meetings were joyful; especially when at last we could understand something new.

I strongly believe that there is no miracle in the fact that all these young people became university professors, known researchers and some of them "excellent" ones. Happiness and hard work lead to good results. To our governments, who are so worried about the quality of research, I would give the only advice: make your researchers happy and they will be good!

Bruno was a bit singular among these young people by a very strong taste for well-built theories, a taste which apparently he has kept all his life until now: good definitions should lead to a better understanding and, thus, in a natural way to nice results. He works hard and is very patient, writing and writing again until he is satisfied. Many researchers, including myself, become feverish when they come close to an idea or to a result, they stop sleeping, eating, talking to others and spend all their time digging to grasp the gold nugget they can see shining in their mind. That is not at all the case of Bruno: he works steadily, leaving his office at five, forgetting about his work to spend the rest of the day with his family, and he resumes his work the next day punctually at nine. His doctoral dissertation, submitted in 1976, cleaned up the links between algebraic trees, recursive program schemes and strict deterministic context-free grammars, a notion introduced by Michel Harrison: this is a rare case of a dissertation which achieves the goals which were defined when started!

Irène Guessarian, who had submitted her dissertation one year earlier, Bruno and me works then on what we called algebraic interpretation of recursive program schemes and wrote a few papers on the subject, but the attention of the still rather small community of researchers in formal semantics was then much more focused on the fixpoint semantics of Dana Scott and the link with the lambda-calculus of

Church. And Bruno left to enter the Labri in Bordeaux, as a professor. In a wise way he decided to change slightly his field of research, all new doctors have to do that at some time to find an area of activity which is different from the area of their supervisor. The occasion was given to him when he was asked to supervise the work of Paul Franchi-Zanettacci on attribute grammars: these strange objects were used by many people for compiler writing. In plain words they tell more about the syntax than an ordinary grammar, they are for programming languages the equivalent of the “lexique-grammaire” that Maurice Gross suggested in order to describe the syntax of natural languages, after criticizing the lack of power of Chomsky’s grammars. But they were in a very messy state, desperately looking for a proper, neat and precise definition. They were in such a messy state that this would have discouraged most people, but it did not discourage Bruno.

I shall not tell the rest of Bruno’s story. The attributes attached to the elements of the language turns them into small graphs and the grammar generates then a graph rather than a sequence. Whence the idea of formulating them as graph grammars and, since Bruno is Bruno, the will to set up a complete formalism to generate sets of graphs and getting for graph languages the notions of recognizability, algebraicity (context-free ness) and definissability in a logical calculus which were successful in languages (sets of words). This took him thirty years and the result is the big book which has just been published: the best you can do is to read the book. It will take you some time but it is worth it. We need be thankful to Bruno and Joost to have completed this book which will serve a source of knowledge and inspiration to many young scientists. For years to come.

I believe Bruno is a happy man: he worked hard but he did what he wanted to do. When retiring he offers to science, to his colleagues, to his former students and students, and to the younger generation a beautiful book. And in the meantime he has been rewarded by the joy of discovery and by a large number of quotations of his papers by researchers in the whole world. This large number is explained by a miracle, which is indeed no miracle: Bruno’s very theoretical work happened to be very useful to many people working on graph algorithms who form a much larger community than the community of formal semanticists. It is no miracle for good theories, which make things clearer and easier to teach or understand are extremely useful even to people who are very far from theories. He was, a bit painfully chosen as a senior member of the Institut Universitaire de France and this gave him time to work on the book. All his life he has been helped by his beautiful, quiet and charming wife Dominique.

I wish them both, Bruno and Dominique a very long “retraite”: they deserve it!