Logic, Formal Linguistics and Computing in France: From Non-reception to Progressive Convergence

1 author:

Pierre Mounier-Kuhn

French National Centre for Scientific Research

20 PUBLICATIONS   **69** CITATIONS

Some of the authors of this publication are also working on these related projects:

History of fluid mechanics / aerodynamics in France (1920-1971) View project

Logic, Formal Linguistics and Computing in France: From non-reception to progressive convergence

Pierre Mounier-Kuhn

Chargé de recherche, CNRS & Université Paris-Sorbonne
Associate Researcher, Centre Alexandre Koyré-CRHST
mounier@msh-paris.fr

Abstract. How did the theory and practice of computing interact to generate a new discipline, computer science? Studying the French scene in comparison **to** other countries, reveals that in most cases computing developed initially as an ancillary technique of applied mathematics, with little awareness of the path-breaking theories of computability elaborated in the 1930s. This was particularly clear in France, where mathematical logic was almost inexistent and disregarded by the Bourbaki group.

It was only in the early 1960s that researchers in the emerging field of computing felt the need for theoretical models, and discovered the Turing machine and recursive functions. Simultaneously, an interest for language theories and information structures, fostered by practical projects such as machine translation, converged with issues raised by software development and the nascent theory of automata.

The convergence of these diverse intellectual agenda was central in the process of construction of the new discipline.

Keywords: discipline building, reception, computer science, informatique, mathematical logic, machine translation, computational linguistics, France.

Introduction

A previous paper suggested a historical model to analyze the early development of computer science in universities [1]. In all local cases studied in this typology, computing began as an ancillary technique of applied mathematics. This until the early 1960s, when a cross-fertilization process began as different intellectual and socio-political agendas converged around this new "boundary object", the computer, hybridizing into a new "science" (arguably) and institutionally into a new discipline: Computer science or *informatique*.

The present paper will focus on two such intellectual agendas in the French post-war environment: Mathematical logic and machine translation, one motivated by fundamental queries, the other by practical concerns[1]. It is based on archival research and oral history interviews providing a detailed investigation on the case of France – a mid-size country where computers appeared a few years later than in Britain and in the USA, which makes it

---

[1] This paper stems out of an early version presented at the *Symposium on the History and Philosophy of Programming* (Birmingham, 5-6 July 2012, available online), with three years of research added, and of questions addressed in my book [2].

unconspicuous regarding spectacular "firsts", but perhaps historically more representative of the average emergence process of computer science.

This narrative differs markedly from the representation of computer science as an offspring of mathematical logic, as a development of the breakthroughs made during the 1930s by Kurt Gödel, Alonzo Chuch, Alan Turing and others in the theory of computability. While this representation holds true in a few important cases, particularly in some American universities[2], the vast majority of people who built early computers or started to teach their use had hardly any knowledge of these mathematical logic concepts. This problem has already been discussed by historians who studied pioneer countries, particularly by Michael Mahoney, Thomas Haigh, Edgar G. Daylight and others [5-8, 63]. On the spectrum of the different *histories of computing*, France constitutes a case where mathematical logic played no part at all in the early development of this technology.

I. "The desert of French logic"

Was the "Turing machine", to sum up a common narrative, a decisive source of inspiration for electronic computers designers? When faced with such an assumption, a historian spontaneously relates it with the more general *linear process of innovation*, a mental model which spread after the second world war. This model stressed the role played by basic science in the development of revolutionary technologies (such as atomic energy), then became hotly controversial as other actors highlighted the role played in the same innovations by engineering and incremental progress. Meditating on the vision of the abstract "Turing machine" materializing into hardware between 1936 and 1949, a theologian could even understand it as a secular version of the Christian process by which the Verb became Flesh. To remain in the computing realm, I tend to consider this model as a founding myth of computer science, an *a posteriori* reconstruction, more than an accurate historical account. It holds true only for theoretical computer science, which blossomed from the 1960s on.

Research suggests rather a late *encounter* than a *filiation* process between logic and computing. As in most other countries, computing in the 1950s emerged in a few French universities as an ancillary technique of applied mathematics, mainly of numerical analysis, to answer the needs of electrical engineering, fluid mechanics and aeronautics. In the science faculties of Grenoble and Toulouse, then of Lille and Nancy, at the CNRS' Institut Blaise Pascal in Paris, small teams of applied mathematicians and electronic engineers endeavoured to get unreliable vacuum tube calculators to process algorithms written in binary code or assembler: Their concerns were far removed from the high abstractions of mathematical logic.

Of course we must distinguish between several branches of mathematical logic. Boolean and propositional algebra was taught and used as soon as the first binary digital calculators were developed in French firms and laboratories, around 1950. At the Bull company, engineers specialized in

circuit theory and design were commonly called "logicians". From 1956 on, a British logician, Alan Rose, published several notes in the *Comptes-Rendus de l'Académie des Sciences de Paris*: in 1956 on propositional calculus, and, in 1959, on an "ultrafast" calculator circuit [9,10]. Binary circuit logic was common knowledge among computer designers by 1960, and would soon be implemented in CAD software based on graph theory, pioneered in France by Claude Berge.

Things went differently with the theories of computability and recursive functions which had "revolutionized" mathematical logic in the 1930s; they remained almost ignored in France until the mid-1950s, and did not seem to interact with computing until the early 1960s. The present paper aims at describing their progressive reception (particularly of the Turing machine concept), through individual trajectories and institutional developments.

In the beginning, we should call it a non-reception. A specific feature of the French mathematical scene was that logic had nearly disappeared since Jacques Herbrand's premature death in 1931. Moreover, it was banned from mathematics by the Bourbaki group, and rejected toward philosophy [11]. Erring "in the desert of French logic" was the feeling of a doctoral student desperately seeking for a supervisor in this field around 1950 [12]. Until 1954, the *Comptes Rendus de l'Académie des Sciences*, a true mirror of French academic research, contain no mention whatsoever of recursive functions or computability theory. The same goes with specialized mathematical periodicals, including university journals.

Only a couple of savants, Jean-Louis Destouches and Paulette Février, worked on the logical foundations of physics [13, 14]. Février also published translations of foreign logicians with whom she had friendly relations (E. W. Beth, Hao Wang & Robert Mc Naughton, A. Robinson, A. Tarski [15]), in a collection of books she directed at a Paris publishing house, and organized a series of international conferences: *Applications scientifiques de la logique mathématique* (1952) [16], *Les Méthodes formelles en axiomatique (logique mathématique)*, *Le Raisonnement en Mathématiques* (1955), etc. Thanks to her, research in logic remained present in France, at least as an imported product[3].

Note that Alan Turing himself was familiar with France and had visited the country repeatedly in the 1930s and after the war ; yet he did not seem to have any contact with French mathematicians [18]. Only three mentions of the Turing machine appeared in France in the first half of the 1950s, with little or no apparent effect. Let us evoke them briefly.

In January 1951, at the CNRS international conference on calculating machines, F. M. Colebrook, a delegate from the British National Physical Laboratory, introduced his presentation of the ACE computer by mentioning Turing's paper of 1936 – "a most abstract study which was in no regard a preview of modern automatic digital calculators", yet attracted the interest of the NPL director [19] (Colebrook headed the construction of the ACE computer initially designed by Turing at NPL). This mention raised no visible echo in the 600 pages of the conference proceedings, nor in the memory of the participants. There is no hint, in the CNRS archives, that Turing himself was

---

[3] In a brief note in the *Comptes Rendus de l'Académie des Sciences*, Paulette Février had merely mentioned Turing in a list of mathematicians using "constructions" in their demonstrations [17].

invited at all. In short, this considerable cybernetics meeting established no link between theories of computability and calculating machines.

More important perhaps, at the end of the same year, the Bourbaki seminar invited a German-French-Israeli logician, Dov Tamari, to speak about "*Machines logiques et problèmes de mots*"[4]. Tamari described the Turing machine and remarked that the term was misleading – it essentially consisted of logical schemes representing a simplified "ideal calculating man". It belonged to pure mathematics and offered a new perspective on algorithms. Yet Tamari noticed that Turing's theory might have a "possible application in the field of calculating machines". In short, these two glimpses of the Turing machine were very far from presenting it assertively as the model for modern computers. Tamari's lectures received no visible echo in the French mathematical community.

Figure 1. CAB 2022 computer developed by SEA in Paris (1955).

The Frenchman who was the most likely to grasp the implications of Turing's discoveries was François-H. Raymond, an electronics engineer and entrepreneur with a deep mathematical culture. He only heard of Turing once his company had designed its first computers. Let us focus for a moment on this micro case, in light of a recently discovered volume of technical reports. After the war, Raymond had been deeply impressed by the EDVAC report of von Neumann and Goldstine, and had created a start-up company, Société d'électronique et d'automatisme (SEA), to develop computing and automation devices. In November 1949, he wrote an internal note, the first sketch for a stored-program machine in France [21]. This note described briefly the architecture of a computer, provided von Neumann's table of order codes and gave an example of a numerical application in this code. Anecdotically, it was at about that time that von Neumann, travelling in Europe, paid a visit to the SEA.

This study was developed over the following years in a set of technical reports, exploring solutions for the design of a big computer, CUBA, which would eventually be installed five years later in the French Army's central laboratory. A young mathematician, Claude Lepage, who had attended the CNRS conference of 1951 (was he the only listener who caught the mention of Alan Turing?), was commissionned to imagine principles of programming.

---

[4] See [20]. Dov Tamari (1911-2006), né Bernhard Teitler, had prepared his doctorate in Paris in the 1930s.

Starting with reports from the von Neumann and EDSAC teams, Lepage compared the merits of the different programming methods, and embarked on "rationalizing" them to elaborate a better one [22].

By 1952, Lepage mastered the topic well enough to propose exploring new computer structures, still in a dialogue with the works conducted at Princeton and Cambridge (UK)[5]. His aim was to escape path dependency (to use the vocabulary of present historians of technologies[6]) : computers are not bound to follow the old organisation model of computing bureaus or of office machines, "as early automobiles conserved the silhouette of the horse carriage." If only, because their application field is much wider : "There is a net change when we consider the machine from a general informational point of view, that is as a device made for transmitting, after a transformation, a certain *quantity of information*." This was the first time Lepage went beyond his point of view as a mathematician designing a *calculator*. In this report, Lepage considered the problem from the fundamentally logical point of view, that of the machine "of Professor Thuring", which he described briefly by mentioning the 1936 article on computable numbers : "a device which circulates and transforms words (collections of a finite number of symbols belonging to a denumerable set)", words which present two sorts of properties, those linked to the *state* and those linked to the *location*.

Turing's influence thus appears here, through a paragraph in a technical report within a small company in a Parisian suburb, very limited (I did not find anything similar in the archives of Bull, the major French computer manufacturer then). Turing's influence did not intervene at the design stage of SEA's first computers, but merely as an inspiration for a future program of investigation on possible architectures. And as an inspiration for a remarkable change of perception, from computers as calculators to information-processing automata, and from code to language, a change which Raymond explicited in conferences in the mid-1950s in Paris and Milan[7]. Strangely, the young mathematician who found Turing's paper inspiring for his reflexion eventually disappeared soon after from the nascent computing scene: Lepage wrote internal SEA reports until about 1957, then left the company and none knows what became of him afterwards. Still the inspiration remained, and SEA never ceased to explore novel computer architectures during the next decade.

The mid-1950s: A revival of mathematical logic

The first paper of importance dedicated at that time to automata theory and computability, in France, was presented in 1956 in Paris by a Swedish cybernetician, Lars Löfgren, at the *International Conference on Automatic Control* organised at CNAM by F.-H. Raymond[8]. Lars Löfgren worked then at the Stockholm Institute for Defense Research. His paper surveyed and

---

[5] See [23]. Lepage was interested by symmetrical ternary logic (-1, 0, 1), which he envisioned as a mean to facilitate mathematical programming. From the electronic point of view, he added, the ternary system may require more components than binary logic, yet in many operations it allows to reduce their number. The concept of a ternary architecture was eventually developed soon afterwards at the Moscow University by N.P. Brusentsov, with his Setun computer (1958).

[6] The concept of *path dependency* has been elaborated in many publications, particularly the seminal paper [24].

[7] See [25]. The evolution from code to language in the USA is analyzed in [26].

[8] See [27]. The same article appeared as [28]. Lars Löfgren became later, in 1963, professor of system theory at the University of Lund (Sweden).

discussed particularly the programmatic articles of A. Turing, "On computable numbers […]" (1936) and "Computing machinery and intelligence", *Mind* (1950), of J. von Neumann, "The general and logical theory of automata" (1951), and of C.E. Shannon et J. McCarthy, "Automata studies" (1956). Starting from the practical concerns of engineers regarding the limits of what is *automatable*, the problems of circuit reliability and of coding errors, he explained the usefulness of these theories which, he stressed, would become more necessary with the growing complexity of automatic systems. This paper, given in English in a grand conference gathering many French pioneers of the discipline, then published in 1959, seems to have had little echo in French-speaking territories, if we judge by the fact that no French publication ever quoted it. We can only suppose that this introduction to automata theory was read without being quoted from 1959 on, and contributed to the introduction of these concepts into the culture of French *informaticiens* and *automaticiens*.

Yet mathematical logic had started a revival in 1955, when Bourbakist Henri Cartan invited the Austrian-American Georg Kreisel to teach in Paris. At the same time, the Polish-American logician Alfred Tarski was enticed (perhaps through Paulette Février) to give a series of conferences at the Institut Henri Poincaré. Simultaneously, three French doctoral students – two mathematicians, Daniel Lacombe and Jean Porte, and a philosopher, Louis Nolin – dared to embrace this marginal matter. Let us introduce the first two, to have a glimpse at their trajectories (we will meet the third later).

Daniel Lacombe graduated from Ecole Normale Supérieure in 1945, and initially studied number theory and other mathematical themes well established in the French school. In 1955 he started to publish brief texts on recursivity[9], likely under the influence of Kreisel with whom he co-signed two papers. After a sabbatical year at the IAS in Princeton, he presented in 1960 a complete overview on "La théorie des fonctions récursives et ses applications" (75 pages), reviewing Gödel's, Church's and Herbrand's theorems, Turing's machine, Kleene's works, etc. The only French author he quoted was Jean Porte, which confirms there was no other. The introduction stressed that the theory of recursive functions was "*à la base de la majorité des résultats intéressants obtenus en Logique mathématique au cours des trente dernières années*", in other words a paradigm in this branch of mathematics. This considerable article also mentioned briefly that this theory was useful for the formal representation of electronic calculators, which in turn stimulated reflexions on the old, intuitive concept of calculation. Lacombe was not seeking to "sell" this theory to computer specialists, but the fact that he exposed it in the *Bulletin de la Société Mathématique de France* allowed it to touch numerical analysts as well as pure mathematicians [30].

Jean Porte studied logic within philosophy, in which he graduated in 1941. He then took mathematics while participating in the French resistance in the Toulouse region. In 1949 he joined the French statistics institute (INSEE) where he invented the *catégories socio-professionnelles* for the 1954 census – an essentially empirical work. Meanwhile Porte began research in mathematical logic and presented a paper on modal logic at a 1955 conference on *Reasoning in Mathematics* [31]. This conference marked a renaissance of

---

[9] See [29]. When I interviewed Lacombe in 2013, I asked him about Dov Tamari who had lectured on "logical machines" at the Bourbaki seminar, yet the name did not strike any chord in his memory.

mathematical logic in France, particularly as the French admitted that problems of logic could be expressed in algebraic form and that mathematicians were interested [32]. In 1956 Porte proposed "A simplification of Turing's theory" at the first international Cybernetic conference in Namur (Belgium) [33]. This paper reveals that at least *one* Frenchman had read the major works by Church, Curry, Gödel, Kleene, Post, Robinson, Rosenblum and Turing on computability, lambda-calculus and recursive functions theory. It is also worth noting that Porte was adressing a *Cybernétique* audience, which still included specialists of computers (who would soon keep cybernetics at bay as a set of vague speculations). Porte's conclusion, however, mentioned no practical implication of these theories which might concern them. On the contrary, he suggested to "reach an even higher level of abstraction than Turing's machines". If he talked to cyberneticians, it was from the balcony of the logicians' ivory tower.

In 1958 he received a CNRS researcher position, at the Institut Blaise Pascal in Paris, where another philosopher turned logician, Louis Nolin, had just been appointed to manage the computer pool. Porte and Nolin soon began writing programs for the Elliott 402 and IBM 650 computers of the institute. This was the first recorded interaction of logicians with electronic computers in France ; yet, we do not have clues about the relationship they possibly established between their research in logic and their practice as programmers.

Even if they did, they remained exceptions for several years. Most computer experts were either struggling with vacuum tube circuit and magnetic drum problems or focused on developing numerical analysis, so that computability theories made little sense to them. Their learned society, Association Française de Calcul (AFCAL), created in 1957, reflected these concerns through its journal, *Chiffres*, and its first meetings, where computability theories remained invisible for several years.

As for mathematical logic, its intellectual status within mathematics remained low. In 1961, Jean-Jacques Duby, a bright young *Normalien*, had the fancy idea of chosing logic for his doctoral research under Lacombe's supervision. "The head of mathematics at Ecole Normale Supérieure, Cartan, was quite fond of me, but when he heard of this weird choice he became apoplectic and didn't speak to me for weeks !" [10] Cartan had invited Kreisel in Paris, but could not tolerate that a "real mathematician" among his *protégés* wandered in this backwater of algebra.

The 1960s: A convergence with computer science

Things changed in the early 1960s, when a series of events manifested a convergence between logic and the nascent computer science.

In October 1961, IBM's European education center at Blaricum (Netherlands) hosted a meeting on the *Relationship Between Non-numerical Programming and the Theory of Formal Systems*[11]. The initiator was Paul Braffort, a mathematician whose curiosity ranged from logic to linguistics, including formal poetry, song writing and private jokes[12]. Braffort had created

---

[10] Author's interview with Jean-Jacques Duby, 24 March and 21 October 2009.

[11] The proceedings were published two years later with an even more explicit title [34]. The proceedings are only a selection of the papers presented (author's interview with P. Braffort, 26 June 2012).

[12] See [35]. For an overview of Paul Braffort's life and activities, see his website.

an analogue computing laboratory at the Commissariat à l'Energie Atomique, near Paris, and was now heading Euratom's computer center in Brussels, for which he had ordered an IBM system. D. Hirschberg, then scientific advisor at IBM Belgium, offered him out of courtesy the use of IBM's facility at Blaricum for any meeting he might wish to organize. Braffort seized the opportunity to gather logicians and computer scientists.

Several French computer scientists and logicians participated, mostly from Paris. Among the speakers, they heard Noam Chomsky and Marcel-Paul Schützenberger lecture on "The algebraic theory of context-free languages", and John McCarthy present his vigorous manifesto, "A Basis for a Mathematical Theory of Computation", which proclaimed the foundation of a new *science of computation* based on numerical analysis, recursive function theory and automata theory.

Figure 2. Meeting on the *Relationship Between Non-numerical Programming and the Theory of Formal Systems* (October 1961) at IBM's European education center at Blaricum (Netherlands). P. Braffort & D. Hirschberg. 1st row: Paulette Février (pearl necklace), next to E.W. Beth, and half-masking P. Braffort. 2nd row: M.-P. Schützenberger, P. Dubarle, S.J. (photo: courtesy of P. Braffort).

In June 1962, a mathematics conference held at the science faculty of Clermont-Ferrand included sessions on computing and on logic, the latter represented by a constellation of international stars – Tarski, Beth, Bernays, Rabin, etc. In his keynote address, René de Possel, the head of the Paris computing institute, Institut Blaise Pascal, explained that mathematical logic, hitherto a field of pure speculation, had become useful to mathematics in general and to information processing in particular[13]. De Possel stressed that Von Neumann, "the first promoter of electronic computers", was also a logician; and that, at a humbler level, programmers proved more efficient when they knew some logic – "to my great astonishment", De Possel confessed (very likely with the examples of Porte and Nolin in mind). With Von Neumann, Turing and others had emerged a general theory of machines, which interested computer designers as well as users. It appeared in several new application fields. While attempts to make machines reason were still embryonic, actual work on machine translation, automatic documentation, artificial languages and their compilation, revealed problems resorting to mathematical logic and linguistics. "To the point that special courses in logic should be created for this purpose", concluded De Possel.

---

[13] The conference was divided in four sessions covering Pascal's domains: Logic, Numerical analysis and automatic computing, Probabilities, Differential geometry and mathematical physics. The proceedings were published as [36].

Implicit in De Possel's lecture was the questioning of old disciplinary categories. If even mathematical logic was becoming useful for a matter as technical as computing, what became of the established difference between "pure" and "applied" mathematics? This epistemological question was soon to have a practical side, too, as the CNRS was about to restructure its committee system. And a most controversial problem would arise: If pure and applied mathematics were reshuffled, where should computing go? Should it be integrated in electronics, or in mathematics? Or should it have an evaluation committee of its own, like a full-fledged science? This problem would agitate the scientific community for a long decade [37].

At the second IFIP congress (Munich, August 1962), a session was devoted to "Progress in the logical foundations of information processing" – a topic not adressed at the first IFIP congress in Paris (1959). John McCarthy hammered again the gospel he was preaching at Blaricum a year before; and engineer Heinz Gumin, from Siemens, explained why computer designers needed mathematical logic [38]. Among the French delegation (nearly 10% of the audience), at least a few listeners got the message.

Actually the message was already being spread in the French computing community through its learned society AFCAL. In late 1961, at the AFCAL seminar on symbolic languages, Louis Nolin, who had attended the Blaricum meeting, gave a programmatic lecture. He recommended designing computer languages according to the axiomatic method established in mathematics – Algol being exemplary of this approach. In order to build an algorithm, it was useful to determine first if the function was effectively computable. For this, "computer scientists would be well advised to learn about the solutions elaborated 30 years ago by logicians"[14]. This remark of Nolin, in a way, sums up my whole paper: After a long decade of tinkering, computer scientists in need of theoretical bases found them in the logicians' work of the 1930s.

Louis Nolin had become De Possel's assistant and chief programmer at Institut Blaise Pascal, and was thus in a good position to translate words into action. In the autumn of 1962, regular courses of "Logic for programmers", on the theories of computability and recursive functions, were introduced in the computer science curriculum of the Paris faculty of science at graduate level. J.-L. Destouches, assisted by Jean Porte, Daniel Lacombe and a third logician, Roland Fraïssé, organized a seminar. Meanwhile, Paulette Février published a translation of A. Grzegorczyk's classic treaty on recursive functions, and created within the Institut Blaise Pascal a collection of brochures explicitly titled "Logic for the calculator's use": Reprints of journal articles, seminar and course texts, doctoral dissertations in logic, were thus made available beyond the tiny circle of French logicians.

From 1963 on, logic was thus firmly established in the computer science curriculum at the University of Paris' *Institut de Programmation* and at the CNRS Institut Blaise Pascal. Beside its intellectual interest for programmers, outlined by Nolin and others, the adoption of logic had an institutional motivation: Computing teachers needed to set up course programs with more formal matters than Fortran training or the physical description of machines, and logic responded perfectly to this quest.

---

[14] See [39]. This paper gave a few major references: Gödel's definition (1934), its use by Kleene [40], Martin Davis' treaty [41], and Daniel Lacombe's overview on recursive functions [30]. Nolin had also read Rózsa Péter [62] in its original German version.

Figure 3. Graduate-level courses at the Institut de Programmation (1965). Courses specialized in formal grammars, algorithms, formal concepts of machine and program, seminars on mathematical logic or automata theory, manifested the convergence of various intellectual agenda around these boundary objects: Computers and software.

This introduction of logic in the "automatic computing" curriculum coincided with other topical evolutions. Now equipped with more powerful and reliable second-generation computers, researchers could address new "crucial problems" – problems likely to shape a scientific discipline: Language compilation, algorithmic complexity, computability, structures of information. Seeking theoretical models, they found them in logic, as well as in other branches of algebra and in formal linguistics. Reciprocally, logicians could use computers, for example to test demonstration procedures.

Other universities followed progressively. Grenoble was practically in phase with Paris, although at a smaller scale, as logic was taught by an astronomer turned linguist, Bernard Vauquois. Vauquois had defended a doctoral thesis in astrophysics, but devoted his *deuxième thèse* to "Arithmetization of logic and theory of machines", thus read works by Alan Turing and John Von Neumann regarding computability, logic and formal languages[15]. In 1959 he was put in charge of a laboratory for machine translation and became the first French member of the Algol committee. While Vauquois soon turned completely to machine translation, he still introduced basic notions of, and references to, mathematical logic in the Grenoble computer science curriculum. In particular, the cross-fertilization between various scientific fields in the mid-1960s in Grenoble is well exemplified by the prehistory of the Prolog language, as told by one of its participants [43]: The synergy between two projects – Algol compiling and natural language processing – led young researchers to absorb a wealth of recent international publications on syntax analysis, W-grammars, graph theory, recursive functions and lambda-calculus. This boiling exploration of new avenues geared itself with another soaring movement, artificial intelligence and automatic demonstration, and later led to Prolog and to a novel conception of algorithmics, directly based on mathematical logic[16].

Jean-Jacques Duby, whom we have seen at odds with Cartan at the Ecole Normale Supérieure, persevered for a while in logic. Lacombe gave him a paper just published by Hao Wang, who had written a computer program that mechanically proved mathematical logic theorems from Whitehead and Russell's *Principia Mathematica* [46], and Duby undertook the writing of programs in LISP to demonstrate the exercises of Alonzo Church's text book automatically. Using the big IBM 7090 at IBM France to this end, he caught the attention of Benoit Mandelbrot, who headed a scientific unit within IBM Corp. at Yorktown Heights, and soon joined IBM. Duby never completed his doctorate. Instead, he switched to programming languages and systems, and ended up heading a computer science laboratory jointly created by IBM and the University of Grenoble in 1967. He was the first French computer scientist trained in all branches of mathematics, except in numerical analysis [47][17].

Soon after Grenoble, other faculties where computing science remained firmly rooted in mathematics joined this convergence movement, particularly

---

[15] See [42]. The *deuxième thèse* was a requisite for the *doctorat*, to make sure that potential professors had a broad scientific culture beyond their major specialty.

[16] See [44]. The direct link from Church's lambda-calcul to Prolog was, in 1970, Colmerauer's reading of the paper written five years before in Princeton by John Alan Robinson [45].

[17] Author's interview with Jean-Jacques Duby, 24 March and 21 October 2009.

Nancy, Lille and Clermont, in conjunction with research on the Algol language [47].

In 1966, the Ministry of National Education defined a new, nationwide masters diploma, *Maîtrise d'Informatique*, including a certificate of "Algebra, mathematical logic, compiler and system theory" [48]. Logic thus switched status, shifting from a marginal intellectual topic to a recognized subdiscipline within the academic curriculum. This recognition, in turn, implied that the University train and hire logicians.

Boosted by this interaction with an expanding new discipline, mathematical logic flourished again in French universities at the end of the decade. Reciprocally, the alliance between logicians and computer practitioners was a decisive factor in the assertion of computing as a new science. This dynamism was further reinforced by the convergence with another emerging discipline: Linguistics.

II. From Machine Translation to Computational Linguistics

While advances in logic responded initially to fundamental queries, the machine translation projects which emerged in the 1950s were motivated mainly by practical concerns : How could scientists keep up with the growing flow of publications in different languages ? And, even more vital in the context of the Cold War, how could the West gather intelligence on scientific and technical efforts carried in the Soviet block (and vice-versa) ? Electronic brains might provide a solution, both as documentary systems and as fast translators. Starting with a few ideas and experiments on both sides of the Atlantic from 1946 on, research on machine translation came to mobilize by 1961 some thirty teams and 4 to 6 million dollars worldwide.

I will only give here a short, sketchy account of a story which is worth a book, and has been analyzed from a linguist's point of view by Jacqueline Léon[18]. We will also leave aside, for another paper, the research efforts devoted by pioneers of humanistic text processing (lexicography, etc.) and of other linguistic approaches.

French linguists in the 1950s were hardly more receptive to American structuralist explorations than mathematicians to computability theories[19]. The Société de Linguistique de Paris, largely dominated by marxist savants, was more influenced by the Russian school of mathematical linguistics. In the rare occasions when they paid attention to the emerging theories of formal linguistics, of Z.S. Harris' *Methods in Structural Linguistics*, and later of Chomsky's revolutionary approach, they either criticized them sharply or misunderstood them – or both. If the first collective book on *Machine Translation* published in the USA was reviewed in France in 1957, it was not by a linguist but by Jean Porte, the logician turned programmer with whom we made acquaintance in the previous section of this paper [52]. In other words, the method, purpose and stake of formal linguistics which developed across the Atlantic made little sense in the French linguists' intellectual landscape, and was clearly at odds with their scientific agenda [53].

In this context, research on Machine Translation was not initiated by linguists, but rather by (relative) outsiders in the late 1950s, when the

---

[18] For a worldwide, historical overview of Machine Translation, see [49, 50].

[19] On the different degrees of reception of American innovations, including cybernetics, by French computer experts, see [51].

establishment of De Gaulle's administration favoured long-term policies, R&D investments and collaborations between academic, military and industrial scientists. The initiator was Emile Delavenay, who as director of the Publication Service at UNESCO, was interested in machine translation and surveyed international advances in this field. In 1958 he created a working group, and soon an *Association pour la traduction automatique des langues* (ATALA). The founding congress of the International Federation of Information Processing Societies (IFIP), also held at UNESCO in June 1959 and where there was much talk of machine translation, contributed in opening French computer specialists to this field and to other non-numerical applications.

Members of ATALA were a mix of linguists, mathematicians, computer experts or logicians, including a few military engineers and officers – about a hundred members by 1960. Its journal *La Traduction automatique*, launched in 1960, produced jointly by Bull and IBM France, was a vehicle for the diffusion of American linguistics and of formal language studies in France. Topics ranged from machine translation to automatic documentation and applied linguistics. Simultaneously, ATALA created a seminar of quantitative linguistics at Institut Henri-Poincaré, which also hosted the first computer of Paris University and the seminar of the Association Française de Calcul.

A convergence of interests between ATALA, the Army and the Centre National de la Recherche Scientifique (CNRS) swiftly led to the creation of two laboratories in 1959, funded jointly by the Defense and the CNRS, under the common name of the *Centre d'Etudes de Traduction Automatique* (CETA). The first one was located near Paris within the Army's Central Laboratory under command of a military engineer, Aimé Sestier ; the other was created at the university of Grenoble, under direction of astronomer Bernard Vauquois. Both men, in addition to Delavenay, were soon appointed members of the CNRS' Linguistics Committee, a decision which confirmed the desire of the CNRS directors to shake up the little French linguistic sphere.

Both laboratories hired or trained computer engineers who would serve the machine, and specialists of languages – rather practitioners of Russian and other languages than academic linguists – who would elaborate translation methods. Both hoped to rapidly develop Russian-to-French translation techniques, in order to keep track of Soviet scientific and technical publications in real time, and to achieve operational results by 1965. The belief in a quick feasability of machine translation was based altogether on technoptimism, on the ignorance of computer engineers regarding linguistic constraints and pecularities, and on the certainty that the Soviets were more advanced than Westerners, both in machine translation and in missile technology.

However the two laboratories were soon to diverge.

At the university of Grenoble, we have already mentioned that the team's director, Bernard Vauquois, had by personal curiosity acquired a culture in mathematical logic, and had learned to program scientific calculations for his doctoral dissertation in astrophysics. His arrival as professor in Grenoble reinforced the university's computer science curriculum, where he introduced basic notions of mathematical logic and of theory and practice of formal languages, particularly of Algol.

While Vauquois did not do research in these fields, he supervised doctoral students who explored the crossroads between them and began to establish a discipline of programming, distinct from numerical analysis. He soon turned his own research completely toward machine translation. His approach was based on the development of a "swivel-language" (*langage-pivot*), which

would function as an intermediary between source and target languages. Note the similarity with the Universal Computer Oriented Language (UNCOL), projected in 1958 by a working group of SHARE and the American Association for Computing Machinery, which aimed to "translate" programs written in high-level languages into machine code [26, p. 60]. Vauquois worked on the hypothetical analogy between translation and compilation, a key issue in programming in the early 1960s.

Throughout the decade, Vauquois' team resisted the growing waves of criticism from linguists, particularly from the new generation of computational linguists, who objected that automatizing the translation process would require considerable basic research in linguistics before yielding any convincing practical result. The Army's support lasted until 1967, while CNRS funding would be maintained over the next decades, thus giving the *Centre d'études pour la traduction automatique* ample time to adjust its theoretical models and produce results justifying its survival.
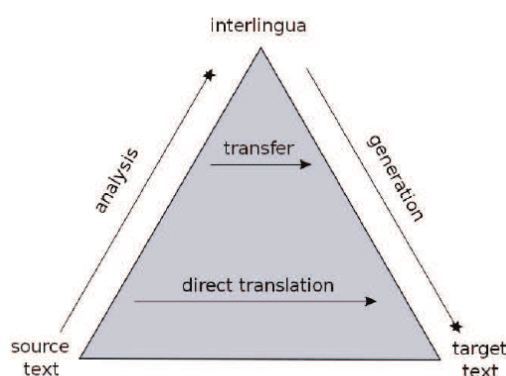

Figure 4. Bernard Vauquois' pyramid, showing his concept of machine translation through an intermediary "swivel-language".

The Paris team, installed at the Army's Central Laboratory (*Laboratoire Central de l'Armement*) in Montrouge, comprised a mixture of young military engineers trained at Ecole Polytechnique and academics and numerical analysts and "linguists" – or rather specialists of a given language, particularly Russian. It was headed by a military engineer, Aimé Sestier, who had pioneered scientific computing on the first stored-program computer developed in France, CUBA, and taken courses in numerical analysis and programming in Grenoble. His center hosted men who worked not only on ballistics and operations research, but also on coding, cryptography or literary analysis, and was ready to answer the Defense's need for machine translation.

The collaboration with the Grenoble team soon appeared to be limited, due altogether to an ill-conceived division of labor between the two centres (morphology / syntax), the difference of theoretical models, and the uncompatibility of their computers (an IBM 650 in Paris, a Bull Gamma ET in Grenoble).

Moreover, the naive technoptimism of the beginnings led to cruel disillusions. Failed tests of ideas on the computer, and criticism by linguists, revealed that human translation was a subtle, complex process, much more difficult to automate than expected if one wanted to do better than automatic dictionaries. In 1959-1960, the US National Science Foundation entrusted an assessment mission on machine translation experiments to Y. Bar-Hillel, a logician who stressed the theoretical fragility of these projects [54]. The Bar-Hillel report was a devastating evaluation, and triggered drastic cuts in

machine translation budgets in America and elsewhere – its conclusions being confirmed five years later by the ALPAC report.

In 1962, after three years of intense work, Sestier read the Bar-Hillel report and decided at once to terminate his machine translation enterprise, refocusing his laboratory on mainstream scientific computing. Most members of the team left for the University or the CNRS. For several polytechnicians interested in research, it was an opportunity to switch to an academic career, in particular at Institut Blaise-Pascal.

A young military engineer, Jacques Pitrat, took on the research program on artificial intelligence born from the speculations of Alan Turing and the Dartmouth meeting of 1956. He studied formal systems and ambitioned to invent a theorem demonstrator, an "artificial mathematician" in his words, a project to which he eventually devoted his doctorate and the rest of his scientific life[20]. Pitrat left the military R&D and joined the Institut Blaise Pascal as a CNRS researcher, bringing with him logic problems linked with automatic demonstration and artificial intelligence. He defended in 1966 a doctoral thesis on a demonstrator of theorems and meta-theorems, the first French doctorate in AI. Jacques Pitrat, Paul Braffort and others interacted on artificial intelligence research at Euratom in Brussels and in a "Leibniz" seminar at Ispra (Italy), leading to a book by Braffort – arguably the first book including "*artificial intelligence*" in its title[21].
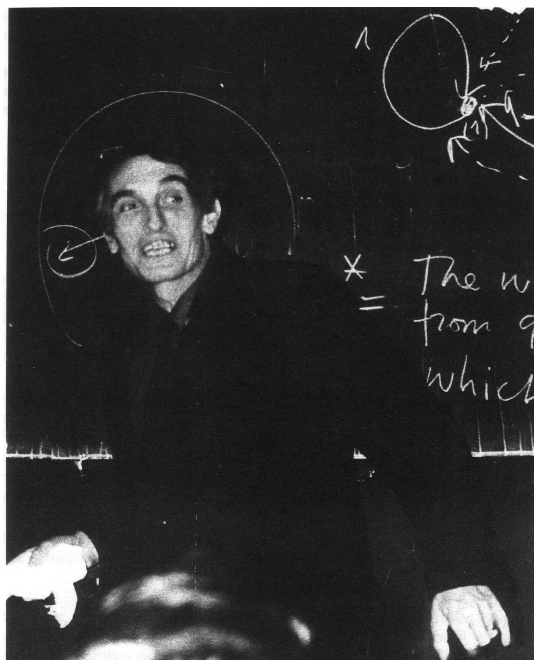
Figure 5. Marcel-Paul Schützenberger at the IFIP conference in New York (1965).

Another polytechnician, Maurice Gross, switched from mathematics to linguistics in 1961 when he went to the USA with a UNESCO grant to study at MIT. There, he followed the course of Noam Chomsky, and later obtained

[20] Author' seminar with J. Pitrat, Sorbonne University, 12 March 2008; and interview, 2 February 2011, Grenoble. Se also [55] and Pitrat's website: http://www-apa.lip6.fr/META/theses.html.

[21] This book predated the books published in the USA by Herbert Simon [57] and by John Slagel [58].

his PhD under Zellig Harris at the University of Pennsylvania. Maurice Gross had been the one to bring the Bar-Hillel report to his boss Sestier. Personal reflection and the Bar-Hillel report convinced both men that machine translation belonged to engineering and had to be separated from basic research in formal linguistics. Gross reinvested all his personal passion and his former training as a "hard scientist" into linguistics, a field in which he was soon recognized internationally. This move coincided with the creation of several academic teams of linguists interested in computational linguistics, particularly in Paris and in Nancy.

Back in Paris, Maurice Gross, now a CNRS researcher at Institut Blaise Pascal, met with three remarkable men with whom he established a long-lasting scientific friendship: Together they interweaved computing, algebra, logic and linguistics on the Parisian intellectual scene. Marcel-Paul Schützenberger, a biologist and mathematician who had written a seminal paper with Noam Chomsky [59], introduced automata theory in France and was a natural leader in the creation of a French school of theoretical computer science. Jean-Claude Gardin, a navy officer turned archeologist after the war, pursued two intellectual agendas: to formalize reasoning in social sciences, close to Pitrat's artificial intelligence projects, and to develop methods for automatic documentation, practically and theoretically. This search led Gardin to create two laboratories and develop a specific programming language, Snobol (automatic documentation and information retrieval were another research field motivated by practical concerns, which revealed new, fundamental problems). André Lentin was an algebraist interested in formal grammars, with whom Gross wrote a treatise which soon became a classic [60].

Figure 6. Maurice Gross and Maurice Nivat at the ICALP conference, 1972 (photo INRIA). The intellectual radiance of Schützenberger, combined with new problems emerging from computing, attracted men interested in algebra, linguistics and software like Gross, Lentin, Nolin, Nivat, who actively promoted theoretical computer science.

These new objects of knowledge, theories and problems circulated rapidly in the boiling intellectual atmosphere and academic expansion of the 1960s. By the mid-1960s they were introduced into the nascent curriculum in computer science, particularly at the Institut de Programmation of the Paris University. Simultaneously, these men all "lobbied" with the governmental agency, DGRST, which defined French science policy and awarded research contracts accordingly – so that about one fifth of the funding for computer science, hitherto mainly devoted to machine architecture, technology and numerical analysis, was reoriented to support research on programming languages, compilation, formal linguistics and automatic documentation. Beyond the Blaise Pascal and Henri Poincaré institutes, Schützenberger, Gardin, Gross and Lentin taught these matters in every institution which invited them — the chair of Numerical analysis at the Sorbonne, the Center for quantitative linguistics created at the Sorbonne by Prof. Jean Favard, the chair of Computer Science at the the University of Toulouse, the University of Pennsylvania, etc., and in the universities where M. Gross and his friends became professors: Aix-en-Provence, Paris-Vincennes, and finally Paris 7-Diderot where this invisible, but not unconspicuous college of theoretical computer scientists/linguists finally settled in the 1970s.
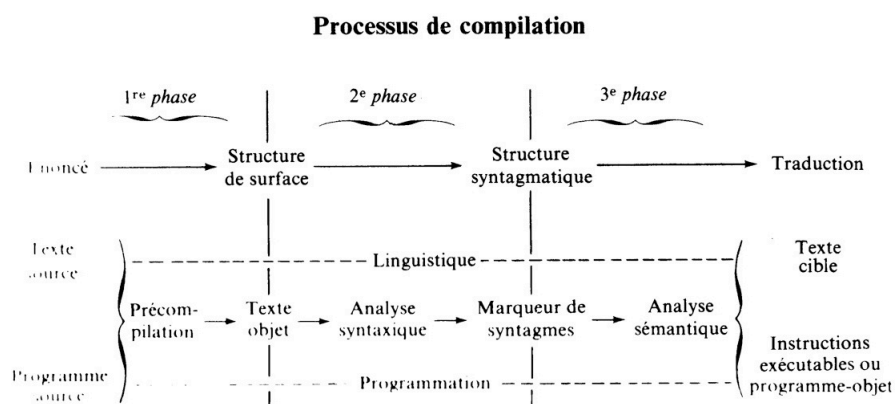

Figure 7. Convergence of linguistic theories and software issues.
Perceiving common structures between different phenomenons was a founding process of research in computer science. A typical example was the similarity between the translation of natural languages and the compilation of programming languages, as charted here by René Moreau, a military officer turned linguist and chief computer scientist at IBM [61, p. 45].

Conclusion

This story may be summed up in terms of timing and receptivity. During the fifteen years following the end of the war, French mathematicians and linguists pursued intellectual agendas where the theories of computability developed in other countries since the 1930s, or the algebraisation of linguistics, made little or no sense, and could thus not be integrated. It was not a case of "conservatism vs. progress", but a typical case of different professional groups being "differently progressive" (to use politically-correct jargon).

Then, within a short period in the early 1960s, sweeping changes occurred at an accelerated pace: Mathematical logic became a topic of several publications and doctoral dissertations by French scholars, and met with the

growing need of computer experts for theoretical models; at the same time, formal linguistics became paradigmatic for a fraction of linguists and for researchers interested in programming languages and information structures. This sudden receptivity was largely due to the general expansion of French research and higher education under the Gaullist regime, which favoured the arrival of a new generation of scientists on one hand (although men like Lentin and Schützenberger were already professors in the 1950s); and, on the other, to the progress of computing techniques and capabilities, which allowed for the broadening of applications, particularly non-numerical, while requiring a better understanding of what computing was.

However interesting these conceptual investigations were, it was only the political pressure from a strong socio-economic demand that supported their institutionalization and allowed them to participate in the construction of a new discipline. Reciprocally, they brought a formalized framework to computing techniques which, alone, would never have been able to rise to such an academic status.

Computing was not the first technology which developed long before it received a proper theory. Similar cases had happened in the past, particularly that of the steam engine which inspired thermodynamics, or of the electron tubes whose physical principles were only fully understood after the second world war in which they had served by the millions, at a time when transistors appeared to replace them. In Kuhnian terms, in each of these cases a technical revolution converged with radically new theories to build a paradigm, a disciplinary matrix ; yet, the emergence of computer science resulted from a convergence of intellectual agendas whose diversity was unprecedented.

References

[1] P. Mounier-Kuhn (2012), "Computer Science in French Universities: Early Entrants and Latecomers", *Information & Culture: A Journal of History*, 47(4), pp. 414-456
[2] P. Mounier-Kuhn (2010), *L'Informatique en France, de la seconde guerre mondiale au Plan Calcul. L'émergence d'une science*, Presses de l'Université Paris-Sorbonne
[3] H. Rogers Jr (1967), *Theory of Recursive Functions and Effective Computability*. McGraw Hill
[4] M. Minsky, Computation: Finite and Infinite Machines, Prentice Hall
[5] E. G. Daylight (2012), *The Dawn of Software Engineering. From Turing to Dijkstra*, Heverlee (Belgium), Lonely Scholar
[6] E. G. Daylight (2015) "Towards a Historical Notion of 'Turing, the Father of Computer Science'", *History and Philosophy of Logic*, 36(3).
[7] T. Haigh (2014), "Actually, Turing Didn't Invent the Computer", *Communications of the ACM*, 47(1), pp. 36-41.
[8] M.S. Mahoney (2011) "Computer Science: The Search for a Mathematical Theory", in Haigh, T. (ed.), *Histories of Computing*, Harvard University Press
[9] A. Rose (1959), "Un multiplicateur ultrarapide", *CRAS*, 248, pp. 2271-2272
[10] A. Rose (1962), "Remarque sur la machine universelle de Turing", *CRAS*, 255, pp. 2044-2045

[11] A.R.D. Mathias (2014), "Hilbert, Bourbaki and the scorning of logic", in C.T.Chong, Qi Feng, T.A.Slaman and W.Hugh Woodin (eds.), *Infinity and Truth*, World Scientific Lecture Notes, Institute for Mathematical Sciences

[12] R. Fraïssé (1994), "René de Possel. La période algéroise", in J. Sakarovitch (ed.), *De Bourbaki à la machine à lire. Journée d'hommage à René de Possel (1905-1974)*, Paris, Publications de l'Institut Blaise-Pascal, p. 56

[13] M. Guillaume (2009), "La logique mathématique en France entre les deux guerres mondiales : quelques repères", *Revue d'histoire des sciences*, 62(1), p. 177-220

[14] M. Bitbol (2001), "Jean-Louis Destouches, théories de la prévision et individualité", *Philosophia Scientiae*, 5(1), pp. 1-30

[15] A. B. & S. Feferman (2004), *Alfred Tarski: Life and Logic*. Cambridge University Press, p. 220- 230

[16] P. Destouches-Février (ed.) (1954), *Applications scientifiques de la logique mathématique: Actes du 2e colloque international de logique mathématique*, Paris, Aug. 1952, Institut Henri Poincaré. Centre d'études de logique symbolique, Collection de logique mathématique. Paris, Gauthier-Villars ; Louvain, E. Nauwelaerts.

[17] P. Février (1948), "Le calcul des constructions", *CRAS*, 227, p. 1192-1193

[18] A. Hodges (2012), *Alan Turing: The Enigma* (preface by Douglas Hofstadter), [New York, Simon and Schuster, 1983], London, Vintage

[19] F. M. Colebrook (1953), "Le modèle pilote du calculateur automatique électronique arithmétique (ACE) du NPL", *Les Machines à calculer et la pensée humaine*, Paris, Éditions du CNRS, p. 65

[20] D. Tamari (1951-1954), "Machines logiques et problèmes de mots. I : les machines de Turing (T.M.)". *Séminaire Bourbaki*, 2

[21] F.-H. Raymond, *Exposé sur la structure logique des grandes machines à calculer universelles*, 22 November 1949, SEA private archives

[22] C. Lepage, *Etude sur la logique générale des machines arithmétiques*, 15 November 1950, CL/DB, Internal Note SEA 2M12

[23] Cl. Lepage, *Quelques idées nouvelles en matière de machines arithmétiques : machines à microprogrammes, machines complètement arythmiques*, 19 June 1952, CL/DB, SEA Internal Note no. 9M12

[24] P. David (1985). "Clio and the Economics of QWERTY". *American Economic Review*, 75(2), pp. 332-337

[25] F.-H. Raymond (1955), "Les calculatrices numériques universelles", *Mémorial de l'Artillerie française*, n° 3 & 4

[26] D. Nofre, M. Priestley & G. Alberts (2014), "When Technology Became Language: the Origins of the Linguistic Conception of Computer Programming", *Technology and Culture*, 55(1), pp. 40-75

[27] L. Löfgren (1959), "Automata of High Complexity and Methods of Increasing their Reliability by Redundancy", *Actes du Congrès international de l'Automatique* (Paris, 18-24 June 1956), Bruxelles, Presses Académiques Européennes, pp. 34-42

[28] L. Löfgren (1958), "Automata of High Complexity and Methods of Increasing their Reliability by Redundancy", *Information and Control*, 1(2), pp. 127-147

[29] D. Lacombe (1955), "Extension de la notion de fonction récursive aux fonctions d'une ou plusieurs variables réelles I", *Comptes Rendus Académie des Sciences Paris*, 240, pp. 2478-2480

[30] D. Lacombe (1960), "La théorie des fonctions récursives et ses applications. (Exposé d'information générale)". *Bulletin de la Société Mathématique de France*, vol. 88, pp. 393-468

[31] J. Porte (1958), "Recherches sur les logiques modales", *Congrès international du CNRS sur Le Raisonnement en Mathématiques*, Paris, CNRS, pp. 117-126

[32] R. Feys (1955), "Un colloque sur le raisonnement scientifique (Paris - 26 septembre - 1er octobre 1955)". *Revue Philosophique de Louvain*, 53(40), pp. 636-639

[33] J. Porte (1958), "Une simplification de la théorie de Turing", in *Actes du 1er congrès international de Cybernétique* (Namur), Paris, Gauthier-Villars, pp. 251-280

[34] P. Braffort & D. Hirschberg (eds.) (1963), *Computer Programming and Formal Systems*, Amsterdam, North-Holland

[35] P. Braffort (1988), "Les digitales du mont Analogue", *Colloque sur l'Histoire de l'Informatique en France*, Grenoble, INPG

[36] *Annales de la Faculté des sciences de l'Université de Clermont*, Série Mathématiques, 1962, vol. 7-8

[37] P. Mounier-Kuhn (2012), "Comment l'informatique devint une science", *La Recherche*, n° 465, pp. 92-94

[38] H. Gumin (1962), "Digital Computers, Mathematical Logic and Principal Limitations of Computability", in C. M. Popplewell (ed.), *Information Processing 1962, Proceedings of IFIP Congress 62*, Munich, North-Holland

[39] L. Nolin (1963), "Quelques réflexions sur les langages de programmation", *Chiffres*, 6(1), p. 11-12)

[40] S. C. Kleene (1952), *Introduction to Metamathematics*, Amsterdam, North-Holland

[41] M. Davis (1958), *Computability and Insolvability*, New York, McGraw-Hill

[42] C. Boitet (1988), "L'apport de Bernard Vauquois à la traduction automatique des langues naturelles", *Colloque sur l'histoire de l'informatique en France*, Grenoble, INPG, vol. 2, p. 63-80

[43] J. Cohen (1988), "A View of the Origins and Development of Prolog", *Communications of the ACM*, 31(1), pp. 26-36

[44] A. Colmerauer (2011), *Du traitement de la langue naturelle à Prolog*, Grenoble, Ensimag

[45] J. A. Robinson (1965), "A Machine-Oriented Logic Based on the Resolution Principle", *Communications of the ACM*, 5:23–41

[46] H. Wang (1960), "Toward Mechanical Mathematics", *IBM Journal of Research and Development*, 4(1), pp. 2-22

[47] P. Mounier-Kuhn (2014), "Algol in France: From Universal Project to Embedded Culture", *IEEE Annals of the History of Computing*, 36(4), p. 6-25

[48] A. Lentin, "Projet de réforme de l'enseignement à l'Institut de programmation", Institut Blaise Pascal archives, s.d. [certainly early 1966]

[49] W. J. Hutchins (ed.) (2000), *Early Years in Machine Translation: Memoirs and Biographies of Pioneers*. Amsterdam, John Benjamins.

[50] J. Léon (2015), *Histoire de l'automatisation des sciences du langage*, Lyon, ENS Editions

[51] P. Mounier-Kuhn (2002), "L'informatique française : une résistible 'américanisation' (1946-1970)", in D. Barjot & C. Réveillard (eds.), *L'Américanisation de l'Europe occidentale au XXe siècle,* Presses de l'Université Paris-Sorbonne, pp. 207-226

[52] W.N. Locke & A.D. Booth (eds.) (1955), *Machine Translation of Languages*, 14 essays, New York, MIT and John Wiley & son, pp. 208-226

[53] J. Léon (2010), "Automatisation-mathématisation de la linguistique en France dans les années 1960. Un cas de réception externe", in F. Neveu *et alii* (eds.), *Congrès Mondial de Linguistique Française*, Paris, Institut de Linguistique Française, pp. 825-838

[54] Y. Bar-Hillel (1960), « The Present Status of Automatic Translation of Languages », *Advances in Computers*, vol. 1, p. 91-163

[55] J. Pitrat (1966), *Réalisation de programmes de démonstration de théorèmes par des méthodes heuristiques*, faculté des Sciences de Paris

[56] P. Braffort (1968), *L'Intelligence artificielle*, Paris, PUF, coll. "La Science vivante"

[57] H. Simon (1969), *The Sciences of the Artificial*, Cambridge (MA), The MIT Press

[58] J. Slagel (1971) *Artificial Intelligence: The Heuristic Programming Approach*, New York, McGraw-Hill

[59] N. Chomsky and M.-P. Schützenberger (1963), "The algebraic theory of context-free languages", in P. Braffort et D. Hirschberg (éd.), *Computer Programming and Formal Systems*, Amsterdam, North Holland, p. 118-161

[60] M. Gross and A. Lentin (1967), *Notions sur les grammaires formelles*, preface by Noam Chomsky, Paris, Gauthier-Villars

[61] R. Moreau (1975), *Introduction à la théorie des langages*, Paris, Hachette

[62] R. Péter (1951), *Rekursive Funktionen*, Budapest, Akadémina Kiadó

[63] M. Tedre, *The Science of Computing: Shaping a Discipline*, New York, CRC Press, 2014.