in "Chemins croisés entre les mathématiques, la biologie et l'epistémologie", M. Montévil, B. Bravi, J.A. Perez-Escobar, A. Angelini (Eds.), Éditions Spartacus-Idh (to appear, 2024).

The early years of Italian Theoretical Computer Science, in Pisa

Simone Martini*

Dipartimento di Informatica-Scienza e Ingegneria
University of Bologna, Italy
February 29, 2024

Abstract

We discuss the transition from computing practice to computer science in Italy during the late 1950s and early 1960s. It highlights the role of two key academic figures, Corrado Böhm (PhD ETH Zurich, 1954) and Alfonso Caracciolo di Forino (MS Turin, 1952), in shaping the development of computer science in the country, in the context of the construction of the CEP, the first computer built in Italy. Böhm and Caracciolo's teaching approaches, rooted in the mathematical theory of computation, laid the foundations for computer science education in Italy. The abstract also touches on Giuseppe Longo's early steps from student in Pisa to his contributions in theoretical computer science.

To Giuseppe Longo, on his 75th birthday

1 Introduction

The late 1950s and early 1960s are the period of transition from the mere practice of computing to the new discipline of computer science. Compared to some Western countries, computing machines arrived in Italy rather late. On the other hand, the development of computing as a science, and thus the break with "its confusing past"[1], is a process that began early, in the mid-1950s. This was due in no small part to the qualities of some of the academic players, especially those involved in the construction of the first Italian computing machine in Pisa.

Before telling this story, however, we should dispel the myth that computer science was born directly out of logic, out of Turing's work, just as Athena was

*Partially supported by: SERICS (PE00000014) under the Italian MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU; and by INdAM GNSAGA.

[1] "No scientific discipline exists without first inventing a visual and written language which allows it to break with its confusing past", as Bruno Latour (1986) says about chemistry.

born as an armour-wearing adult from the head of Zeus. While this narrative might make sense in an idealistic "history of ideas" and might sometimes be useful as a teaching trick, it does not correspond to historical fact. Over the last decade, several authors, e.g. Haigh (2014); Daylight (2015); Bullynck et al. (2015), have challenged the narrative of "Turing the father of computer science", showing that Turing's work had little influence on actual computers[2], and that even John von Neumann, who certainly knew Turing's papers[3], was not *directly* influenced or inspired by them in his work on the design of EDVAC.

Obviously, someone knew. In addition to von Neumann, Herman Goldstine, Haskell Curry, Paul Bernays, Hans Hermes, Saul Gorn and others realised early on that the notion of Turing machine could be used to give an abstract model of a computer, assuming that the computer had potentially infinite computational resources (computing time, memory, etc.). This was particularly clear to the logic school in Münster—Hermes (1954) shows in detail that programmable computers are universal[4]. But it is only in the 1950s that "the (universal) Turing machine starts to become an accepted model in relation to actual computers and is used as a tool to reflect on the limits and potentials of general-purpose computers by both engineers, mathematicians and logicians" (De Mol, 2019). And still later, a mathematical theory of computation, based on logic and Turing's work, arrives as a result of an explicit agenda at the end of the 1950s, when some prominent players (e.g., John McCarthy (1963)) tried to ground programming and computation into mathematics (and in particular mathematical logic) in order to give computing an entry ticket to science (Martini, 2020).

2 Corrado Böhm

Among the select few who knew, was Corrado Böhm, one of the main characters in the story that we are going to tell in this paper. He was born in 1923 in Milan[5], and graduated in Electrical Engineering in 1946 at the University of Lausanne, before becoming a graduate student and research assistant at ETH in Zürich. In this position, he was sent to Konrad Zuse's laboratories, to as-

[2] After the war, Turing designed a computer, the ACE, but it was not the ACE that was the forerunner of the first British computer built, the Manchester Mark I. "Historians agree that the first wave of modern computers [...] during the late 1940s were all inspired by a single conceptual design" (Haigh, 2014), described in von Neumann (1945). Also the actual impact of Turing on computer *science* is not granted, in those early days; for a balanced review see De Mol (2019, Sect 5)—the entry on Turing machines for the Stanford Encyclopedia of Philosophy—from which we quote the following: "Recent historical research shows also that one should treat the impact of Turing machines with great care and that one should be careful in retrofitting the past into the present."

[3] An argument on the equivalence of Turing machines and McCulloch and Pitts's neuron nets "supplied with an infinite blank tape," can be found in von Neumann (1949). For some of the relationships between Turing and von Neumann, see Stanley Frankel's letter to Brian Randell, quoted in Randell (1972).

[4] Universal "in the sense that one can construct a programmable computer such that for every computing machine its computational power is included in the computational power of that computer" (Börger and Glaschick, 2022).

[5] For a biography, see `corradobohm.eu`.

sess the possibility of ETH renting Zuse's Z4. He had two supervisors: Eduard Stiefel, the director of the Institute for Applied Mathematics and designer of the *Elektronische Rechenmaschine der ETH* (ERMETH), and Paul Bernays, Hilbert's former assistant and close collaborator. The Ph.D. thesis was submitted in 1952[6] (in French) and later defended and published (Böhm, 1954). The main contribution of the thesis is a universal programming language based on the single notion of assignment, for which a translation procedure (*codification automatique*) into a lower-level language is given. The translation procedure is described as a program in the same language to be translated, thus being the first example of a metacircular[7] compiler (Knuth and Pardo, 1980). The thesis is remarkable also for several other reasons. The first of these is the explicit reference to Turing machines as a formal model for *any* general-purpose computing machine:

> *Nous voulons admettre – ce qui est assez plausible – que les calculatrices les plus évoluées sont universelles, au sens spécifié par M. Turing.*

We don't have any archival evidence on the sources for this bold statement, but we can reasonably assume that Bernays was no stranger to this reasoning[8]. A second striking remark, again from the introductory material, is the following:

> *Le « programme » est susceptible, par rapport aux calculatrices universelles, d'une double interprétation. La première est : « Description d'un comportement de la calculatrice ». La deuxième : « Description d'une méthode numérique de calcul ».*

In contrast to the crystal clear remark on Turing, this passage is denser and more cryptic. We read it as one of the first examples of the duality between an operational description, the instructions to the computer that specify the machine's behaviour, on the one hand, and the numerical function that results from that sequence of operations, on the other. The function is represented by an intensional description of it (a particular algorithm–*une méthode numérique*– for computing it). Programs are no longer (only) technology to instruct an electronic device—they are algorithms for functions, abstract objects on which mathematics can be done.

After submitting his thesis, Böhm left Zürich and, in 1953, was appointed as a permanent researcher at Istituto Nazionale per le Applicazioni del Calcolo (INAC) of the CNR in Rome. It will be in Italy that, some years later, he will get in touch with Alfonso Caracciolo, the other main character of our story. They will meet in the context of the project for the Calcolatrice Elettronica Pisana (CEP).

[6] The thesis was completed shortly after David Wheeler's August 1951 Cambridge dissertation, thus making it the second Ph.D. thesis in computer science ever presented.

[7] Metacircular, or self-hosting, compilers are programming language translators written in the same language as the source programs.

[8] From the biographical sketch in `corradobohm.eu`: "As Corrado himself likes to recall, it was really Paul Bernays who stimulated his interest in Turing machines and universal computing machines."

3 Alfonso Caracciolo

At the beginning of the 1950s, several Italian research institutes realised that they should have access to an electronic computer. Some of them (such as Politecnico di Milano and INAC-CNR in Rome) decided to buy a machine from one of the few manufacturers at the time. Instead, at the suggestion of the physicist Enrico Fermi, the University of Pisa decided to build such a machine from scratch, developing all the necessary hardware and software technology in-house, without any real prior know-how[9]. The project officially started in March 1955, with the creation of CSCE (Centro di Studi sulle Calcolatrici Elettroniche), organized into two main sections: Engineering (with the goal of designing and realizing the hardware of CEP) and Mathematical Logic (with the goal of designing the CEP instruction set and, then, developing all the necessary software and programming techniques). The head of the Mathematical Logic Section was Alfonso Caracciolo, a thirty-year-old physicist who had graduated from the University of Turin only three years earlier.

Alfonso Caracciolo di Forino was born in Penne[10] in 1925, into a noble family with an ancient tradition and grew up in Rome. At university in 1943 he enrolled in Physics in Rome, where, after a good start, he began to feel unsatisfied[11]. He was interested in foundational themes and language use in Physics, but he could not find any mentor on these subjects in Rome's albeit excellent physics institute. Even accounting for the war period, in 1950 (Physics was a four-year program) he was still struggling with a couple of exams and wrote to Ludovico Geymonat, the most prominent Italian epistemologist of the time. They began to correspond and Alfonso found a mentor who, despite their age difference, would become a friend over time. In 1951 Caracciolo wrote to Vittorio Somenzi, another epistemologist and family friend, asking about "computing machines", the first archival mention of the topic that will later become his main scientific interest. He finally moved to the University of Turin, where Geymonat was professor, graduating in 1952, and entering into Geymonat's circle. It was there that he began to develop his interest in technical languages (Caracciolo di Forino, 1953) and logical problems. During his trial period at CSCE he wrote an internal report (Caracciolo, 1954), where we read:

> *mentre nelle questioni di dettaglio le macchine finiscono per essere notevolmente diverse fra loro, nelle linee generali funzionano tutte secondo gli stessi principi. [...] si può ben dire che quasi tutte le macchine sono 'universali' nel senso che ciascuna di esse è in grado di risolvere qualunque problema che possa essere risolto da ogni altra*

[9] See De Marco et al. (1999); Bonfanti (2012) for accounts at the national level, and Cignoni and Gadducci (2012, 2020) for specific aspects of the CEP project and its impact on Olivetti, the Italian private firm which invested on this project for developing its own electronic computer. Of the CEP project, Auerbach (1961) says: "The computer development work is among the most advanced observed in Europe."

[10] Penne is a small town in the Abruzzo region, where the Caracciolo family owned a *palazzo*.

[11] For these biographical notes I have used material from the archives of the University of Turin, the Geymonat archive in Milan, and the private archive of the Caracciolo family.

calcolatrice elettronica a cifre.[12]

The statement lacks an explicit mention of Turing, but the reference to "universal" and the generality of the claim ("any problem", "any machine") leaves no doubt on Caracciolo's understanding and on the fact that, of course, he is referring to an abstract model where unlimited resources are assumed.

In 1954, both Böhm and Caracciolo had clear that there is more to computing than just technology. It is difficult for us to appreciate the depth, the significance of their observations, shared, as we have already observed, by a fortunate few. We are always tempted to project into the past what is obvious now[13]. Instead, we should remember that programming in those years was essentially a technical matter, closely linked to the specific machine to be programmed. The general adoption of a linguistic conception of programming was not yet complete, as Nofre et al. (2014) argue. This "linguistic turn", and even more the realisation that computing machines are universal, are by no means obvious, they require a distance from physical reality—computer programming used rudimentary tools that were difficult to recognise as "languages"[14], and no actual machine is universal, being finite and using finite resources. The assumption of what Martini (2020) calls the "standard model" of programming languages (i.e. the tacit assumption that programs should be understood as instructions for a machine with true arithmetic and in principle unlimited resources) is similar to what frictionless motion is to mechanics. And like frictionless motion, it is an abstraction with a formidable impact on the possibility of constructing a science around it.

4 Teaching Computer Science

This awareness of the *scientific* nature of computing[15] informs both Böhm's and Caracciolo's teaching. The first prototype of CEP (the *Macchina Ridotta*, MR) was completed in July 1957. At that point, it was clear that CSCE needed also to prepare people to fruitfully *use* the computer. A first technical course for a dozen of graduating engineering students was held in 1956 (Fabri, 1958). But the most important decision was to intervene directly in the education of

[12] "while in matters of detail the machines end up being considerably different from each other, in general, they all work according to the same principles. [...] it may well be said that almost all machines are 'universal' in the sense that each of them is capable of solving any problem that can be solved by any other electronic digit calculator."

[13] The "retrospective illusion of truth" of Bergson (1934): "*Par le seul fait de s'accomplir, la réalité projette derrière elle son ombre dans le passé indéfiniment lointain ; elle paraît ainsi avoir préexisté, sous forme de possible, à sa propre réalisation.*"

[14] Böhm (1954) uses only once the words "*langage formel*" to refer to his "*symbolisme*".

[15] As opposed to vocational. Samuel D. Conte, first Head of the first US Department of Computer Science (Purdue University, 1962) recalls in a 1999 Computerworld magazine interview: "Most scientists thought that using a computer was simply programming—that it didn't involve any deep scientific thought and that anyone could learn to program. So why have a degree? They thought computers were vocational vs. scientific in nature" (quoted in Conte's obituary at Purdue University, 2002).

university students by introducing programming topics into the standard mathematics and physics curricula. At that time, the courses that could be taken for a particular degree were determined by national law[16] and clearly, there was no course directly available for "computer programming". For the academic year 1958-1959, University of Pisa decided to give to Corrado Böhm, by contract, the course of *Calcoli numerici e grafici*[17], an old title for what a few years later would be called Numerical Analysis. Gadducci and Cignoni (2013) published the detailed lesson plan[18] for this course. It is a fascinating journey, where Böhm "naturally" embeds standard numerical topics into a mathematical theory of computation. Not citing numerical analysis subjects, we find: Moore automata[19], Turing machines (TMs) and universal TMs, partial and total computable functions, the existence of uncomputable functions[20], the logical structure of a digital computer, and techniques for programming such a digital computer. This approach, while most other universities focused on the details of the computer architecture they possessed, set the standard for computer science education in Pisa forever. Most importantly, it set that computing was to be understood inside mathematical logic and computability, years before Computer Science re-discovered Turing as its father.

This rooting of programming in the theory of computation is what we find also in Caracciolo's teaching. The archives of the University of Pisa have the lesson plans of Caracciolo's course on "Cybernetics" for three years, starting from 1961-1962. Besides TMs, Post systems and their equivalence to TMs, we find an almost complete course on programming languages, including syntax, semantics, metalanguage, rule-based systems, and an introduction to the ALGOL and FORTRAN programming languages.

The part on programming languages reflects Caracciolo's own research interests. At the end of the 1950s, CSCE was in dramatic need of people proficient in programming techniques, especially system programming, and translators (compilers). They formalised their collaboration with Böhm, hired a young programmer (Florence Ragusa) for two years from the Courant Institute in New York (via Peter Lax), but it was clear that without an active local research community they could not get the right momentum. Caracciolo's interest in languages and their foundations, that was clear since his university years, had finally the possibility to become his research field. In 1963 he published his first paper on the Communications of the ACM (Caracciolo di Forino, 1963)—the most prestigious venue at the time for papers of this kind—on contextual constraints in the definition of programming languages. It was followed by several others in the proceedings of important IFIP conferences (Caracciolo di Forino, 1966, 1965), and again Caracciolo (1966) in the Communications, on "formal

[16] And they remained fixed by law until 1999, leaving universities no freedom but to choose which course to activate among those fixed by the national regulation.

[17] Computing by numerical and graphical means.

[18] *Registro delle lezioni*, preserved in the archives of the university.

[19] Moore (1956) had been published just two years before Böhm's course was given.

[20] While Böhm could have read Hermes (1954), it is highly unlikely that he would have had access to Davis (1958), the textbook which, together with Hermes (1961), would be the standard reference on computability theory for at least a decade.

pragmatics", an entirely new and original topic (which, however, will not have much of a following), before the series on PANON (Caracciolo di Forino et al., 1966), a novel rule-based language that anticipated analogous languages used in artificial intelligence.

With this scientific production, he secured for himself and for CSCE a first-row participation in the research community in programming languages, testified by the organisation in 1965, in Pisa, of a NATO Summer School on Programming Languages, with some of the stars of the field[21]. Some years later, although he was not a member of IFIP WG 2.1[22], Caracciolo hosted the June 1968 WG's meeting in Tirrenia, near Pisa, where van Wijngaarden's controversial proposal that would lead to ALGOL 68 was discussed.

It was during these years that the huge financial and cultural investments made by the University of Pisa since 1955 began to pay off. Not so much for the availability of the CEP[23], but in the skills and know-how that have been developed. When the Rector of Pisa obtained from the Italian Parliament the law establishing the first degree in Computer Science, the heads of the two sections of the CSCE (G.B. Gerace for the Engineering Section[24] and A. Caracciolo for the Mathematical Logic Section) were among the pillars on which the degree rested. Rector Faedo had planned ahead, recruiting at the university and at the local CNR institute some young physicists, mathematicians

[21] Besides Caracciolo himself (Special programming languages), speakers were S. Ginsburg (Theory of context-free languages), P. Landin (λ-calculus and its applications), P. Naur (The systematic design of effective compilers), A. van Wijngaarten (Formal definition of syntax and semantics of programming languages) (IEI-CNR archive, Pisa). Among the attendees, G. Ausiello, J. de Bakker, A. Grasselli, F. Luccio. In Ausiello (2018)'s recollection, "the lecturers were some of the most prominent computer scientists in the world." Caracciolo's lectures presented the special purpose language he was developing, for programming machine tools. He presented its syntax by using BNF, the formal method used for ALGOL, adapting to programming languages the formalisms developed for Chomsky's generative grammars.

[22] WG 2.1 is the working group on *Algorithmic Languages and Calculi* of the International Federation for Information Processing (IFIP). It was formed in 1962 for the support and maintenance of the programming language ALGOL. See `https://ifipwg21wiki.cs.kuleuven.be/IFIP21/Profile` for membership over the years.

[23] Although the CEP was actively used for both research and teaching, it soon became obsolete compared to the commercial computers of the time. Pisa, thanks to the initiative of Alessandro Faedo, mathematician and Rector of the University, secured first a 70/90 as a gift from IBM in 1963, followed by the acquisition of an IBM System/360 in 1969, with a 75% discount (which meant the still astronomical price of 1 billion lire, to be paid in four yearly instalments).

[24] This is not the place to present and discuss the research behind the architecture and electronic design of the CEP. As Cignoni and Gadducci (2012) argue, the MR (the preliminary model available in 1957) "adopted state of the art solutions that it was not easy to find all together on others machines of that period." Among these state-of-the-art solutions, we find a fully microprogrammed control, almost at the same time as the British EDSAC 2 project, which is credited as the first microprogrammed machine. A key element that made microprogramming possible was the MR's small instruction set (only 32 instructions), for which the Mathematical Logic Section was responsible. As with programming and programming languages (see also *infra* for Montanari's course), these architectural issues found their way into the classroom, in the computer architecture courses that Gerace and his group will be teaching for years to come.

and engineers to help design[25] and then implement this new degree in Computer Science[26]. The inaugural lecture was held on 17 November 1969, with 388 students enrolled, and on the same day the second year was also opened to students wishing to transfer from other degrees. The general design was clear from the start: courses with general titles, to cope with the rapid development of the discipline[27]; a strong physical and mathematical background; and—most importantly for our story—a strong emphasis on "science", i.e. methods and general principles, more than technologies. When the fourth year began in 1971, Ugo Montanari gave his first course on *Metodi per il trattamento dell'informazione*, setting the standard and the syllabus for more than twenty years. Turini (2008) recalls that "Montanari had planned to teach the most important theoretical foundations of computer science: formal languages, computability theory, and the semantics of programming languages." The early visionary approach of Böhm and Caracciolo—teaching programming (for Böhm) and programming languages (for Caracciolo) in the context of a general mathematical theory of computation—lived on, ten years later, in a course that would become a fixture of the Computer Science degree (in Pisa and elsewhere in Italy.)

Caracciolo at that time had a prominent international role, also in the organisation and politics of the new science. He was a member of the CNR Informatics Commission, he was trying to establish a European software institute inside the European Community, with the major computer manufacturers of that time in Europe, and he was one of the major players, together with Maurice Nivat, for the creation of the European Association for Theoretical Computer Science (EATCS) (Nivat, 2015). He led the "Programming systems" group at IEI-CNR[28], whose goal was to give formal definition of the semantics of real programming languages using Markov algorithms (Ausiello, 2018). And he continued to give his courses in the new degree, before becoming Professor of *Teoria e tecnica della programmazione* at the Università dell'Aquila and then moving to the LUISS University in Rome, in the first half of the seventies (and switching his research interests to the applications of computing to society.)

5 Conclusions

We can finally come to Giuseppe Longo and tie together the various strands of our story. Giuseppe was a mathematics student in Pisa at the end of the sixties. He attended Caracciolo's lectures, which were stimulating and rich, but

[25] See, for instance, Grasselli (1967) which reviews all the university courses on electronic computers given in the western world at that time.

[26] *Scienze dell'informazione*, which more literally translates into Information sciences.

[27] Remember that a national law was needed to change these titles. Therefore, among the compulsory courses we find *Teoria e applicazione delle macchine calcolatrici* (whose standard content was introduction to programming), *Sistemi per l'elaborazione dell'informazione* (Computer architecture, Operating systems), *Metodi per il trattamento dell'informazione* (Mathematical theory of computation), *Elaborazione dell'informazione non numerica* (AI), etc.

[28] Istituto di Elaborazione dell'Informazione (IEI) was the new name given to CSCE, when it became a CNR institute.

also somewhat chaotic and excessive[29], discovering his own interest in the mathematical theory of computation. He asked for guidance to Antonio Grasselli, the informal chair of the CS degree, who presented him to Giorgio Ausiello, who graduated in Physics in Rome in 1966 with Böhm as supervisor (Ausiello, 2018). Giorgio was a student at the 1965 Summer School on Programming Languages that Caracciolo organised, and was just starting to collaborate with the IEI-CNR in Pisa. Ausiello proposed Longo to work on abstract computational complexity. The thesis—*Studio assiomatico del consumo di risorsa per il calcolo delle funzioni ricorsive*—was defended in 1971, formally signed by Antonio Grasselli. Some years later, Longo had the responsability of the course of *Metodi per il trattamento dell'informazione*, keeping Montanari's syllabus, albeit adapting it according to his own expertise and sensitivity (hence a more formal treatment of recursion theory, and some lambda-calculus.)

It is the beginning of his work on that (contended[30]) border between mathematical logic and theoretical computer science that remained his main field for more than thirty years, before his "philosophical turn".

Acknowledgments

I would like to thank Fabio Gadducci and Giorgio Ausiello for their help and generous comments. And to Isabella Caracciolo for sharing documents from her family archive and for answering my many questions about her father.

References

Auerbach, I. L. (1961). European electronic data processing–A report on the industry and the state-of-the-art. *Proceedings of the IRE*, 49:330–348.

Ausiello, G. (2018). *The Making of a New Science. A Personal Journey Through the Early Years of Theoretical Computer Science*. Springer, Cham.

Bergson, H. (1934). *La pensée et le mouvant*. Félix Alcan, Paris.

Böhm, C. (1954). Calculatrices digitales. Du déchiffrage des formules logico-mathématiques par la machine même dans la conception du programme. *Annali di matematica pura e applicata*, IV-37(1):1–51.

Bonfanti, C. (2012). Information technology in Italy: The origins and the early years (1954 - 1965). In Tatnall, A., editor, *Reflections on the History of Computing: Preserving Memories and Sharing Stories*, pages 320–347. Springer, Berlin, Heidelberg.

[29] Vincenzo Manca, personal communication, 2018.

[30] A proper account of the relations in Italy between mathematical logic and computer science in the early days will be the subject of a forthcoming paper.

Börger, E. and Glaschick, R. (2022). Logic and machines: Turing tradition at the logic school of Münster. *FACS-FACT*, 2022-1:69–129.

Bullynck, M., Daylight, E. G., and De Mol, L. (2015). Why did computer science make a hero out of Turing? *Commun. ACM*, 58(3):37–39.

Caracciolo, A. (1954). Rapporto sulle moderne calcolatrici elettroniche. Technical Report NI-28-1954 (prima serie), CSCE, Pisa.

Caracciolo, A. (1966). Some preliminary remarks on theoretical pragmatics. *Commun. ACM*, 9(3):226–227.

Caracciolo di Forino, A. (1953). Sur la construction du langage de la physique. In *Actes du XIème Congrès International de Philosophie – Philosophie et méthodologie des sciences de la nature*, volume VI, pages 113–118, Amsterdam. North-Holland.

Caracciolo di Forino, A. (1963). Some remarks on the syntax of symbolic programming languages. *Commun. ACM*, 6(8):456–460.

Caracciolo di Forino, A. (1965). Linguistic problems in programming theory. In *Information Processing. IFIP Congress*, pages 223 – 228, New York. Spartan.

Caracciolo di Forino, A. (1966). On the concept of formal linguistic systems. In Steel, T. J., editor, *Formal Language Description Languages for Computer Programming (Vienna, Austria; Sept. 15-18, 1964)*, pages 37–51, Amsterdam. North-Holland.

Caracciolo di Forino, A., Spanedda, L., and Wolkenstein, N. (1966). Panon-1b: A programming language for symbol manipulation. In *Proceedings of the First ACM Symposium on Symbolic and Algebraic Manipulation*, SYMSAC '66, pages 0601–0615, New York, NY, USA. Association for Computing Machinery.

Cignoni, G. A. and Gadducci, F. (2012). Rediscovering the very first Italian digital computer. In *2012 Third IEEE HISTory of ELectro-technology CONference (HISTELCON)*, pages 1–6.

Cignoni, G. A. and Gadducci, F. (2020). Pisa, 1954-1961: Assessing key stages of a seminal Italian project. *IEEE Annals of the History of Computing*, 42(2):6–19.

Davis, M. (1958). *Computability and Unsolvability*. McGraw Hill, New York.

Daylight, E. (2015). Towards a historical notion of 'Turing — the father of computer science'. *History and Philosophy of Logic*, 36(3):205–228.

De Marco, G., Mainetto, G., Pisani, S., and Savino, P. (1999). The early computers of Italy. *IEEE Ann. Hist. Comput.*, 21(4):28–36.

De Mol, L. (2019). Turing Machines. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2019 edition.

Fabri, E. (1958). Appunti delle lezioni di "Introduzione alla programmazione di una calcolatrice elettronica". Technical Report 35, CSCE.

Gadducci, F. and Cignoni, G. A. (2013). A syllabus for the fifties. Teaching computer science on the first Italian computers. In *HaPoC 2013 : 2nd International Conference on the History and Philosophy of Computing*.

Grasselli, A. (1967). Insegnamento universitario nel campo dei calcolatori elettronici. Technical Report 2.61, CSCE-CNR, Pisa.

Haigh, T. (2014). Actually, Turing did not invent the computer. *CACM*, 57(1):36–41.

Hermes, H. (1954). Die Universalität programmgesteuerter Rechenmaschinen. *Mathematisch-Physikalische Semesterberichte (Göttingen)*, pages 42–53.

Hermes, H. (1961). *Aufzählbarkeit - Entscheidbarkeit - Berechenbarkeit. Einführung in die Theorie der rekursiven Funktionen.* Springer-Verlag, Heidelberg.

Knuth, D. E. and Pardo, L. T. (1980). The early development of programming languages. In Metropolis, N., Howlett, J., and Rota, G.-C., editors, *A History of Computing in the Twentieth Century*, pages 197–273. Academic Press, New York, NY, USA.

Latour, B. (1986). Visualisation and cognition: Thinking with eyes and hands. In Kuklick, H., editor, *Knowledge and Society Studies in the Sociology of Culture Past and Present*, volume 6, pages 1–40. Jai Press.

Martini, S. (2020). The standard model for programming languages: The birth of a mathematical theory of computation. In de Boer, F. S. and Mauro, J., editors, *Recent Developments in the Design and Implementation of Programming Languages*, volume 86 of *OpenAccess Series in Informatics (OASIcs)*, pages 8:1–8:13, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

McCarthy, J. (1963). A basis for a mathematical theory of computation. In Braffort, P. and Hirschberg, D., editors, *Computer Programming and Formal Systems*, volume 35 of *Studies in Logic and the Foundations of Mathematics*, pages 33 – 70. Elsevier. A preliminary version presented at the Western Joint IRE-AIEE-ACM 1961 Computer Conference, pp. 225–238. ACM, New York, NY, USA (1961).

Moore, E. F. (1956). Gedanken-experiments on sequential machines. In Shannon, C. E. and McCarthy, J., editors, *Automata Studies. (AM-34), Volume 34*, pages 129–154. Princeton University Press, Princeton.

Nivat, M. (2015). The true story of TCS. *Theoretical Computer Science*, 591:1–2.

Nofre, D., Priestley, M., and Alberts, G. (2014). When technology became language: The origins of the linguistic conception of computer programming, 1950–1960. *Technology and Culture*, 55:40–75.

Randell, B. (1972). On Alan Turing and the origins of digital computers. *Machine Intelligence*, pages 3–20.

Turini, F. (2008). The semantics of Ugo Montanari. In Degano, P., De Nicola, R., and Meseguer, J., editors, *Concurrency, Graphs and Models: Essays Dedicated to Ugo Montanari on the Occasion of His 65th Birthday*, pages 804–805. Springer, Berlin, Heidelberg.

von Neumann, J. (1945). First first draft of a report on the EDVAC. Technical Report Contract no. W-670-ORD-4926, Moore School of Electrical Engineering, University of Pennsylvania.

von Neumann, J. (1949). Rigorous theories of control and information. Published in von Neumann (1966), pages 42-56.

von Neumann, J. (1966). *Theory of self-reproducing automata*, volume edited by A. W. Burks. University of Illinois Press, Urbana and London.