Kalmár's Argument for the Independence of Computer Science*

Máté Szabó[1,2][0000−0001−7721−1103]

[1] Archives Henri-Poincaré, UMR 7117, Université de Lorraine, Nancy, France
mate.szabo@univ-lorraine.fr
https://idea-udl.academia.edu/MateSzabo
[2] IHPST, UMR 8590, Université Paris 1 Panthéon-Sorbonne, Paris, France

Abstract. Computer Science is a rather young discipline, and as usual with new disciplines, in its early stage there were important discussions about its aim, scope and methodology. Throughout these debates, it was claimed at different times that computer science belongs to the natural sciences, mathematics, or engineering. Questions about the organization of the field were raised as well: is there a need for computer science departments, or for separate computer science majors at the university level? The history of these debates has been documented rather well in recent years. However, the literature focuses mostly on sources from the US and Western Europe. The aim of this paper is to include the stance of eminent Hungarian logician and computer scientist László Kalmár in the history of this discussion. Kalmár's view is reconstructed based on recently found, formerly unpublished archival materials from 1970-1971: a conference abstract and his correspondence about Hungarian computer science education. In this paper, I will also situate Kalmár's view among the positions of other prominent scholars in these debates.

Keywords: László Kalmár · history of computer science · Hungary.

1 Introduction

Computer Science is a rather young discipline, and as usual with new disciplines, in its early stage there were important discussions about its aim, scope and methodology. Many people argued that computer science is an independent branch of science worthy of academic examination on its own. However, throughout these debates it was also claimed at different times that computer science belongs to the natural sciences, (applied) mathematics, or engineering. In many cases the arguments were based on the backgrounds or scientific interests of those who put them forward; researchers of artificial intelligence argued for the natural science interpretation, while mathematicians invested in the field of computing emphasized its mathematical aspects. Besides its methodology, the scope of computer science was called into question as well. Is it, to name a few options, the study of machines and related questions (as the name of its first

* I would like to thank Kendra Chilson for her help in writing this paper.

and largest association, the Association of Computing Machinery, indicates), of information and data processing or of algorithms? Again, scholars were usually arguing for one or another view based on their own research interests. During these early, identity-forming years even the name of the field was called into question and generated debates.

These questions about the identity of computer science were not merely intellectual or philosophical questions–they also had practical consequences. For example, if computer science is an independent science, then it should have independent institutions within academia, such as departments at universities. Although today we take the independence of computer science for granted, it was not obvious in the beginning. Indeed, even in 1966, President of the ACM Anthony Oettinger stated, "I personally believe, and still believe that I am right, that departments of computer science have no place in the eternal scheme of things. [...] I am forced to split my mind and say that I believe that it is an intellectual mistake to have departments of computer science, while I believe there is no real tactical alternative to having them" ([16], pp. 27-28). He believed that computer science is not part of either mathematics or engineering, but that it is anchored to both. As a consequence, he was worried that separate computer science departments might become isolated within universities. Another practical question was whether there was a need for separate computer science majors at the university level: if computer science is simply viewed as a tool for natural sciences and engineering, then a couple courses should suffice for the experts of those fields, possibly even only on the graduate level. These questions dominated the discussions about computer science education throughout the 1960s.

Scholars have taken an interest in the history of computer science's quest for its identity as an independent scientific discipline. As early as 1976, Wegner wrote about the different research paradigms in the field [22]. Many of these debates were thematized and further analyzed more recently in [3] and [6]. The most complete historical overview of these debates can be found in Tedre's excellent book [21] from 2014. However, the literature covers almost exclusively Western sources (mainly for language reasons), even though computer science as a discipline clearly had to go through a similar process to gain independence, acceptance, and prestige outside the US and Western Europe.

The aim of this paper is to include a scholar in this discourse from the Eastern Block as well.[1] The short argument presented below comes from prominent Hungarian logician and later computer scientist László Kalmár[2] from 1970-1971. He was at the vanguard of Hungarian research in computer science and automata theory as well as building computing devices. He was also indispensable in the start of computer science education in Hungary ([20]; [18]). As a consequence, he was involved in many similar discussions about computer science as a discipline,

[1] See [7], Section 6 and especially page 183, for examples depicting similar struggles in the Soviet Union.

[2] For bibliographical information and description of his work in the field of computer science see ([20], Section 3) and [14].

and faced many challenges while fighting for its institutional independence in Hungary.

Kalmár, although coming from a mathematical background, argued for the independence of computer science from mathematics based on methodological differences. He used his argument to support certain institutional changes in the Hungarian academic world of computing. I will also situate Kalmár's view among the positions of other prominent scholars in these debates. However, due to lack of space only those with similar views will be indicated.

Sources. While looking through the correspondence between Kalmár and Patrick Suppes[3] in the Kalmár Nachlass at the Klebelsberg Library at the University of Szeged, I accidentally found an interesting acceptance letter from Suppes. The letter, dated the 3rd of May 1971, announces that Kalmár's contributed paper, entitled *Is Computing Science an Independent Science?*, is accepted for presentation at the Fourth International Congress on Logic, Methodology and Philosophy of Science.[4] It seems, however, that Kalmár never did deliver his talk. The proceedings [19] do not mention Kalmár, nor does the list of his official travels [9] mention this congress, and the list of his collected papers in [1] does not contain anything similar. Fortunately, the one-page long abstract can be found in the Nachlass under 'Folder 311' [10]. In addition, again by pure accident, I stumbled upon another exposition of the same argument by Kalmár. Folder 'Lev-12' [11] contains a 24-page long letter from April 10th, 1971 detailing his comments and recommendations about the national computer science education for Hungary's unified computer science initiative. The letter was sent to György Aczél, the secretary for cultural affairs of the Central Committee of the Hungarian Socialist Workers' Party, upon Aczél's request. In this letter, Kalmár gives detailed recommendations for computer science education from elementary school through high school to university, and even postgraduate courses and trainings. The context in which the question of the independence of computer science comes up is the education and training of future academic scholars. I will use these two sources to present Kalmár's argument that computer science is independent from mathematics and the implications he thinks this has for the organization of academic institutions.

Remark. Before turning to Kalmár's writings, I must explain his choice of words. His abstract is entitled *Is Computing Science an Independent Science?*: he uses 'comput*ing* science' instead of the now customary 'computer science.' First it should be remarked that the field itself did not yet have a generally accepted, singular term for the discipline of computing (for examples see Chapter 7 of [21] and p. 324 of [13]). Furthermore, it appears to be a deliberate choice on

[3] Between 1963 and 1965, both Kalmár and Suppes served in the governance of the DLMPS, Kalmár as Vice-President and Suppes as Secretary General, and as members of the Committee on the Teaching of Logic and Philosophy of Science from 1964 until 1968 as well.

[4] The Congress took place in Bucharest, Romania from August 29 to September 4 in the same year.

Kalmár's part, as the typewritten abstract I found originally used the expression "computer scientist," which was later changed to "computing scientist" by hand. His letter [11] written in Hungarian provides some clarification about his choice of words. There (pp. 16-17), Kalmár makes a distinction between two commonly used Hungarian terms, 'számítástechnika' and 'számítástudomány,' which can be translated as 'computing technology (or technique)' and 'computing science,' respectively. Kalmár briefly indicates that he uses these different phrases such that 'computing technology' covers hardware-related issues, while questions concerning software design and engineering belong under 'computing science.' For the remaining part of this paper, I will use 'computing science' wherever I discuss Kalmár's view. The reader should keep in mind that Kalmár understands it to mean what we would today call software design and engineering, and that it does not cover the entirety of computer science, broadly understood. Thus, in Kalmár's terminology 'computing science' is part of computer science.[5]

2 Kalmár on the Independence of Computing Science

As mentioned above, there are two sources that contain Kalmár's argument for the independence of computing science. First, I will use his letter [11] on the unified computer science education initiative to provide the context in which Kalmár used the argument, then display his conference abstract that contains the argument in its entirety [10], and finally, explain some of his points in detail and position him among the opinions of others at the time.

Section 6 of Kalmár's letter ([11], pp. 16-19) is devoted to the question of the "education of academic scholars" in computer science, that is, those who received scientific degrees, engaged in research and possibly stayed in academia. They also clearly were to have a serious impact on the education of the subject, as they would be the ones to teach it at the university level. The unified computer science initiative contained a directive of funding ten computer science departments in the five year period between 1971 and 1975.[6] Kalmár emphasized that, in order to provide quality education, computer science departments needed highly trained faculty conducting research in both hardware- and software-related issues. However, as Kalmár pointed out, there were only two PhD[7] holders in the field of software engineering at the time in Hungary, and "even the number

[5] To make things precise, but possibly even worse, computer science departments in Hungary are usually called 'számítástechnika' departments, thus the word Kalmár uses for hardware-related issues was also used in Hungary as an umbrella term that can be translated as 'computer science' broadly understood.

[6] This period coincides with the Fourth Five Year Plan of Hungary. (Five year plans were overarching, nationwide centralized economic plans in the socialist countries.)

[7] In Hungary, and many other countries in the Eastern Block, this scientific degree was called 'candidate of sciences' ('kandidátusi fokozat' in Hungarian). As it is a PhD-equivalent degree, I decided to use 'PhD' throughout the paper to avoid confusion and cumbersome phrasing. (Indeed, many 'candidate of sciences' degrees were actually converted to PhDs in the 1990s, after the collapse of the Eastern Block).

of those PhDs is rather low that were defended in computing science broadly understood" (p. 17).[8] This obviously posed a problem for the founding of new departments, as the faculty of university departments had to hold a certain number of scientific degrees in order to be accredited.[9] Kalmár's explanation of the low number of software engineering and design PhDs, and his recommended solution, were tied to his argument for the independence of computing science. To understand the context appropriately, we have to go into more detail about the process of obtaining a PhD in Hungary during this period.

Hungary, among other Eastern Block countries, adopted many features of the Soviet academic system. These changes were put into effect in Hungary in 1949, including the creation of the 'Scientific Qualification Committee'[10] of the Hungarian Academy of Sciences. This committee was a centralized organization responsible for the selection of PhD candidates and approval of dissertation topics, as well as approval of faculty members as supervisors, etc. (these decisions were, in many cases, also not free from political considerations). Thus, in this new system, departments and universities lost their freedom and autonomy to award scientific degrees [15].[11]

Although the Scientific Qualification Committee claimed to assign high priority to software-related topics, the number of applicants remained quite low. Kalmár explained the low number of software-related PhDs by the organizational structure of the committee and its approved dissertation categories. Software-related dissertation topics fit only under the *Mathematical machines and programming* category offered by the Mathematical subcommittee. However, according to Kalmár, the subcommittee contained only mathematicians, who understood "programming" as it was customary in operation research at the time, i.e. as linear programming, convex programming, etc.[12] As a consequence, applicants with software-related research interests were often either rejected, as their

[8] It is well known that the Eastern Block lagged behind the West in computing technologies in general. The gap was even larger in the case of software development and maintenance than in the case of hardware ([4] pp. 98-100, and [8]).

[9] In addition, according to Kalmár, most of the PhD holders had already reached well-paid, high ranks in the industry and were unlikely to leave their jobs for academia.

[10] 'Tudományos Minősítő Bizottság' in Hungarian.

[11] For the sake of completeness, it has to be mentioned that from the 1950s, universities were allowed to award a title, colloquially referred to as 'little doctorate' ('kisdoktori' in Hungarian), but it did not count as a scientific degree and in most cases they were not allowed to be converted into PhD degrees in the 1990s.

[12] On p. 17 Kalmár makes a claim, the accuracy of which it is hard to judge today, that this understanding was facilitated by a typo. According to Kalmár, the category was supposed to be called 'Matematikai gépek és programozásuk' which translates as 'Mathematical machines and their programming'. However, the official description read 'Matematikai gépek és programozások,' which differs only in one letter (the second from last), and means 'Mathematical machines and programming,' where programming is actually in plural (which is grammatically correct in Hungarian). Thus, programming wasn't necessarily linked to the mathematical machines anymore, and required multiple kinds of programming, leading to the preference of operation research themed dissertation topics.

topic was "not mathematical enough," or directed towards operation research. In addition, Kalmár noted that some members of the committee "even a couple years ago during committee meetings openly proclaimed their opinion that programmers are trained at universities but no one should apply for a PhD with such a topic, as 'programming is not a scientific research topic'." This attitude kept many worthy candidates from even applying.

Kalmár saw these issues as part of a "natural process" in which new branches have to fight for their acceptance and approval. Thus, that computing science faced these difficulties was not surprising–quite the opposite, it was to be expected. He even mentioned operation research itself and probability theory as recent examples of new branches that had to fight for their acceptance as legitimate branches of academic mathematical research.

However, argued Kalmár further, the case of computing science was somewhat different from the acceptance of those branches. While he acknowledged that computing science had its origins in mathematics, he also argued that its methodology was so different from mathematics that it should be considered "an independent science", i.e. independent from mathematics. This difference in methodology explained, according to Kalmár, the rejected dissertation topics as well, since mathematicians did not understand what counts as an (intellectual) achievement in software design and engineering, and thus could not judge which topic was worthy of a PhD degree. The solution Kalmár proposed was, of course, to create a new computing science subcommittee within the Scientific Qualification Committee where the members were computing experts instead of mathematicians. In January of 1970, he submitted a request for such a subcommittee to be created.

Although today it is widely accepted without much argumentation that computing science should be considered independent from mathematics, it was not so at the time.[13] This is why, at this point in the letter, Kalmár put forward his argument for the independence of computing science based on its different methodology from mathematics. The same argument was accepted (without the aforementioned context) to the Fourth International Congress on Logic, Methodology and Philosophy of Science of 1971. As the argument provided in the Hungarian letter [11] for computing science being an independent science is very similar to the English abstract [10], I display the entire abstract below to show Kalmár's argumentation in his own phrasing, instead of providing a summary of it. This is a formerly unpublished abstract of a presentation that was most likely never delivered. To retain its original appearance, I used a typewriter font and kept its original typesetting. However, I silently corrected typos and clear grammatical mistakes, and changed the parentheses from "/" and "/" to "(" and ")". The two references listed in the Bibliography are not referred to in the abstract text by Kalmár. The three footnotes (14, 15 and 16) are added by me.

[13] For example Knuth in the preface of his [12] from 1968 wrote that "computers are widely regarded as belonging to the domain of 'applied mathematics"' (p. ix). Interestingly, Knuth uses the term 'computer,' not even '(theoretical) computer science' belonging to applied mathematics.

Is Computing Science an independent science?

Computing Science obviously has its origin in Mathematics. The question is, whether it is a branch of Mathematics or it can be considered as an independent science.

Beside its special subject-field, Computing Science diverges from Mathematics by its method. Indeed, while Mathematics is a proof-oriented science, Computing Science is more algorithm-oriented. In any case, a computing scientist puts generally as much ingenuity into his algorithms as a mathematician into the proofs of his theorems.

True enough, algorithms play some role in Mathematics as well. However, even the most sophisticated mathematical algorithms (e.g. Kronecker's algorithm for decision of the reducibility of a polynomial in the field of rationals, say, or Galois' algorithm, using the latter, for decision of solvability, by means of radicals, of an algebraic equation, with rational coefficients, say) are very short relative to a compiling algorithm or to an operational system.

Also, the computing scientist has to prove his propositions, e.g. the correctness of his programs. However, in most cases, the proof has a verificative character. The name "debugging" given to such verifications shows that the computing scientist does not esteem this activity, though important, so high as the mathematician his proofs. In most cases, the errors found in the course of debugging are easily corrected (at least if the programming idea is sound), while errors in mathematical proofs are in general fatal.[14]

A mathematical problem, asking if some statement is true or not, is finally solved by a proof (or disproof) of the statement in question. On the contrary, if one has a computational algorithm for the solution of a given problem of Computing Science, the problem is not yet finally settled, for one is asking for a better algorithm for the same goal (from the point of view of computing time or memory place).[15] Well, a mathematician can also look for a simpler proof of some theorem. However, to find one is not as great an achievement as to find the first proof. On the other hand, the improvement of a computational algorithm is sometimes as (or more) valuable as producing the first algorithm for the same purpose.

These arguments show that Computing Science requires a way of thinking that is different from that of a traditional mathematician. Hence, Computing Science is appropriately considered an independent science rather than a branch of Mathematics.

László Kalmár

[14] This comparison of Kalmár's is not clear without further arguments. For, if the idea behind a mathematical proof is sound, it can be "easily corrected" as well. What he might have meant is that judging an idea to be sound in programming is easier than in mathematics.

[15] On a similar note in the letter (p. 18), Kalmár remarks that a proof of the optimality of a particular algorithm belongs to mathematics.

Bibliography

C.B. Jones, P. Lucas: Proving correctness of implementation techniques, IBM Laboratory Vienna, Technical Report TR 25.110, 12 august 1970.[16]

C.D. Allen: The application of formal logic to programs and programming, IBM Systems Journal, 10:1, 1971.

We see that in arguing for the independence of computing science, Kalmár tried to differentiate it only from mathematics, and not from engineering. Most likely he did not address its relation to engineering for two reasons. First, and most importantly, arguments for computer science and programming being an engineering discipline (i.e. "software engineering") became widespread only later, from the 1970s onward. Second, in Hungary technical universities offered programming majors rather late and did not dominate the field. ([18], [20])

Kalmár emphasized the differing methodologies of computing science and mathematics to set computing science aside from applied mathematics. Indeed, it was customary at the time to categorize computing science as applied mathematics. Of course, the subjects of applied mathematics differ from pure mathematics, but it is still considered to be a branch of mathematics. Thus, Kalmár had to argue that the difference between mathematics and computing science is not a mere difference in their subjects, but a difference in their methodologies.

At the beginning of the abstract, Kalmár declared computing science to be an "algorithm-oriented" science. The most famous advocate of this point of view is most likely Donald Knuth, who was originally trained as a mathematician just like Kalmár. Indeed, in his [13], Knuth wrote that his "favorite way to describe computer science is to stay that it is the study of *algorithms* [...] because they are really the central core of the subject, the common denominator which underlies and unifies the different branches." (pp. 323-324) However, Knuth emphasized the mathematical aspect of algorithms and compared programming to creating mathematical proofs: "The construction of a computer program from a set of basic instructions is very similar to the construction of a mathematical proof from a set of axioms." ([12], p. ix) He did so to stress the strong interconnectedness of programming and mathematics, not only with applied, but with pure mathematics as well. Kalmár, on the contrary, downplayed the role of algorithms in mathematics in order to separate it from computing science.

Furthermore, Kalmár distanced the notion of mathematical proof from the social practice of "proving" programs to be correct, i.e. from "debugging." In addition to pointing out the different practices in "verification" in these fields, he also claimed that verification is not considered to be the intellectually challenging part of programming, quite in contrast to the appreciation of proofs in

[16] Also published as Jones C.B., Lucas P. (1971) Proving correctness of implementation techniques. In: Engeler E. (ed) Symposium on Semantics of Algorithmic Languages. Lecture Notes in Mathematics, vol 188. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/BFb0059698.

mathematics. This part of the argument can be considered Kalmár's response to the so-called "verificationists."[17] This is the view that programs (algorithms) are mathematical entities, and if all their specifications are fully formally described, their correctness could and should be verified by formal mathematical proofs instead of "debugging." ([21], [3]) According to Tedre, "Although the formal verification movement was, from its start in the early 1960s, light years away from the reality of actual programming practice in the industry, many believed in its intellectual superiority." ([21], p. 60) This "intellectual superiority" is inherited from the practices of formal mathematics, which is an accepted and well respected science. Clearly, Kalmár was advocating instead for the examination and acceptance of the practices being used in computing science, such as debugging.

In the last step Kalmár compared where the intellectual effort was invested in these fields. He claimed that the intellectual effort on display in computing science was on par with mathematicians' efforts to provide proofs, but it was used to design ever more sophisticated and complex algorithms. As a consequence, connecting back his argument to the academic and institutional context, mathematicians should not be the ones assessing the intellectual merits of achievements in computing science, simply because they are not acquainted with its methodology and practice.

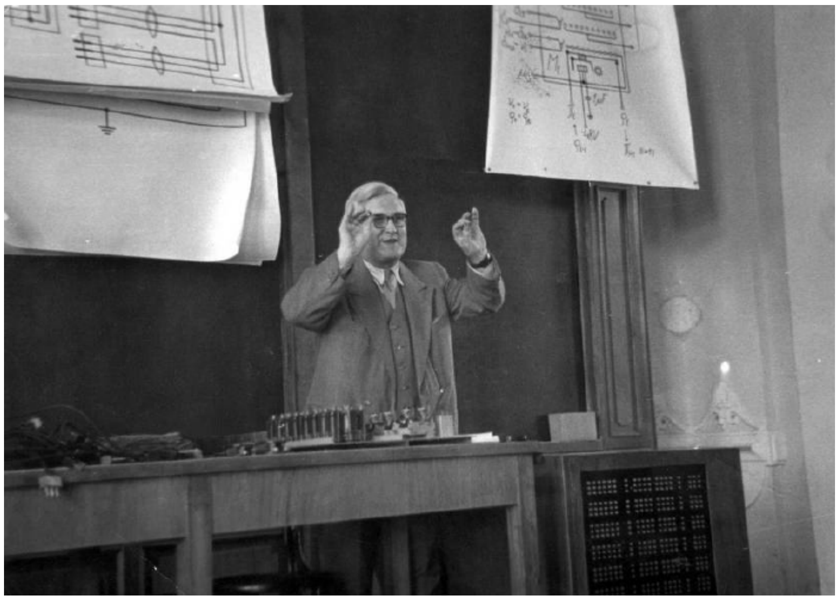

Fig. 1. Kalmár during a lecture. (Picture is from [2]).

[17] Indeed, the two entries in the Bibliography attached to the abstract are proponents of the verificationist view.

3 A Strong Parallel: Kalmár and Perlis

As a final thought for this paper, I would like to put Kalmár's argument and the context in which he gave it in parallel with Alan Perlis' *Computer Science is Neither Mathematics nor Electrical Engineering* [17] from 1968. Although today this similarity might not appear to be surprising, as their stance turned out to be the well accepted one, I find it quite striking just how much their views, background in mathematics and even their positions in the academic institutions lined up with each other despite being on two opposite sides of the world.[18]

Perlis, at Carnegie Mellon University at the time, was instrumental in starting a program in computer science during the mid 1960s, which led to the funding of a separate computer science department which he was the first head of [5]. Similarly, Kalmár started the first university-level training in computer science and programming in Hungary at the University of Szeged in 1957 and was the head of a separate computer science department from 1967 [20].[19] Just as Kalmár expressed frustration over mathematicians assessing computing scientists in the Scientific Qualification Committee, Perlis began his paper by describing how, in the US, the allocation of federal funding for computer science research is decided by various mathematics and applied mathematics committees. Perlis believed this was because "Computer Science is, unfortunately a bit too large to be ignored, and yet too new to be properly treated. As a result, computer science is in danger of being mishandled and misinterpreted" (p. 69).

Similarly to Kalmár, Perlis downplayed the importance of algorithms in mathematics: "Before the advent of the computer, algorithms were encountered, but they were rare, simple, and always consigned to the support and background of other investigations." (p. 70) Then he pointed out features in the practice of computer science that are not shared with mathematics: "Still, there are aspects of computer science's preoccupation with algorithms which are less directly related to mathematics. This is true, for example, of computer programming. The algorithms of computer programming are enormously complex and more specialized than it is the custom of mathematics to treat." (p. 71) Finally, he claimed that since computer science is "preoccupied with design and process" while "mathematics is oriented to abstract analysis" (p. 71), they have different methodological approaches, and thus computer science should be institutionally independent from mathematics.[20]

[18] Again, for lack of space, no one else holding this general position is mentioned from among the many. As just one example, see George Forsythe's position as described by Tedre ([21], pp. 37-38). Still, I believe, Kalmár and Perlis' positions show a striking resemblance.

[19] The department was called *Foundations of Mathematics and Computer Technology Department* until 1971, when it morphed into the *Computer Science Department*, still headed by Kalmár until his retirement in 1975.

[20] Interestingly, even though Perlis mentions "engineering" in the title explicitly, he does not provide arguments for the independence of computer science from it, just as Kalmár did not.

Thus, Kalmár and Perlis described the methodological differences between computer science and mathematics slight differently. Perlis named the "abstractness" of mathematics and the "design" focus of computer science as distinguishing features, and Kalmár pointed to the different approaches of their verification processes. Nevertheless, their academic pasts and positions, the context in which they argued for an independent computer (or computing) science, and their arguments themselves are astonishingly similar.

References

1. Ádám, András and Pál Dömösi. 1986. "Kalmár László." In Pénzes István (ed) *Műszaki nagyjaink. Vol. 6.* Budapest: Gépipari Tudományos Egyesület. 47–89.
2. Bohus, Mihály, Dániel Muszka and Péter Gábor Szabó. 2005. "A szegedi informatikai gyűjtemény (The Computer Collection in Szeged)." Conference slides accessed online on the 18th of March, 2019 at https://www.yumpu.com/hu/document/read/29881933/a-szegedi-informatikai-gyujtemeny-in-memoriam-kalmar-laszlo
3. Colburn, Timothy. 2000. *Philosophy and Computer Science.* Armonk: M. E. Sharpe.
4. Davis, Norman C. and Seymour Goodman. 1978. "The Soviet Bloc's Unified System of Computers". *ACM Computing Surveys*, Vol. 10, No. 2: 93–122.
5. Denning, Peter J. 1990. "Alan J. Perlis, 1922-1990. A Founding Father of Computer Science as a Separate Discipline." *Communications of the ACM*, Vol. 33, No. 5: 604–605.
6. Eden, Amnon. 2007. "Three Paradigms of Computer Science." *Minds and Machines*, Vol. 17, No. 2: 135–167.
7. Ershov, Andrei P. and Mikhail R. Shura-Bura. 1980. "The Early Development of Programming in the USSR." In Metropolis, Nicholas, Jack Howlett and Gian-Carlo Rota (eds) *A History of Computing in the Twntieth Century.* New York: Academic Press. 137–196.
8. Goodman, Seymour E. 1979. "Software in the Soviet Union: Progress and Problems". In Yovits, Marshall C. (ed) *Advances in Computers*, Vol. 18: 231–287.
9. Kalmár, László. 1928–1975. *The Official Travels of Professor Kalmár.* A document assembled by Kalmr Nachlass, Klebelsberg Library, University of Szeged.
10. Kalmár, László. 1970–1971. *Is Computing Science and Independent Science? Abstract.* The abstract is not dated, most likely it was prepared during the previous year to the Congress. In 'Folder 311', Kalmár Nachlass, Klebelsberg Library, University of Szeged.
11. Kalmár, László. 1971. *Számítástechnikai programunk megvalósítása (The Implementation of Our Computer Science Initiative).* In 'Folder Lev-12' (containing Kalmár's correspondence related to the programming major 1957–1974), Kalmár Nachlass, Klebelsberg Library, University of Szeged.
12. Knuth, Donald E. 1968. *The Art of Computer Programming. Vol. 1: Fundamental Algorithms.* Reading: Addison–Wesley Publishing.
13. Knuth, Donald E. 1974. "Computer Science and Its Relation to Mathematics". *The American Mathematical Monthly*, Vol. 81, No. 4: 323–343.
14. Makay, Árpád. 2007. "The Activities of László Kalmár in the World of Information Technology." *Acta Cybernetica*, Vol. 18: 9–14.

15. Nagy, Péter Tibor. 2011. “A tudományos továbbképzés Kádár-korszakbeli társadalomtörténetéhez. (On the Social History of the Postgradual Scientific Education in the Kádár Era)” *Kultúra és közösség*, Vol. 2., No. 15: 23–34.
16. Oettinger, Anthony. 1966. “President’s Letter to the ACM Membership.” *Communication of the ACM*, Vol. 9, No. 12: 838–839.
17. Perlis, Alan. 1968. “Computer Science is Neither Mathematic Nor Electrical Engineering.” In Finerman, Aaron (ed) *University Education in Computing Science*. New York: Academic Press. 69–79.
18. Sántáné-Tóth, Edit. 2017. “Computer Oriented Higher Education in Hungary.” *Studia Universitatis Babes-Bolyai Digitalia*, Vol. 62, No. 2: 35–62.
19. Suppes, Patrick, Leon Henkin, Athanase Joja and Grigore Constantin Moisil (eds). 1973. *Proceedings of the Fourth International Congress for Logic, Methodology and Philosophy of Science IV, Bucharest, Romania, 1971*. Amsterdam: North-Holland Publishing Company and Warszawa: PWN – Polish Scientific Publishers.
20. Szabó, Máté. 2019. “László Kalmár and the First University Level Programming and Computer Science Training in Hungary.” Forthcoming in Leslie, Chris (ed) *Proceedings of the IFIP World Computer Congress, WG 9.7, Poznan, Poland*. 30 pages.
21. Tedre, Matti. 2014. *The Science of Computing: Shaping a Discipline*. CRC Press.
22. Wegner, Peter. 1976. “Research Paradigms in Computer Science.” In *ICSE ‘76: Proceedings of the 2nd International Conference on Software Engineering*. Los Alamitos: IEEE Computer Society Press 322–330.