Was Algol 60 the First Algorithmic Language?

Helena Durnová
Masaryk University, Brno

Gerard Alberts
University of Amsterdam

Editor: Nathan Ensmenger

The phrase "algorithmic language" is conspicuously associated with Algol, the acronym first used to name the programming language Algol 60, which originated through a cooperation between the ACM and Association for Applied Mathematics and Mechanics (GaMM) groups of programming specialists. One crucial meeting was the first joint meeting of the two groups, held in Zurich, 27 May to 2 June 1958. The report from this meeting, known as the Zurich Report, was made available to a wide audience through the *Communications of the ACM* in December 1958 as the "Preliminary Report–International Algebraic Language"[1] and through the new German-based journal *Numerische Mathematik* as the "Report on the Algorithmic Language Algol by the ACM Committee on Programming Languages and the GaMM Committee on Programming"[2] in December 1959, a year later. The two articles are basically identical, but their titles are not. Why the shift from "algebraic" (IAL) to "algorithmic" (Algol) in 1958 or 1959? Clearly, the community was searching for a word. Just like "procedure," "information," "code," or "program," the notion of an "algorithm" was one of the qualifications of choice to characterize the quintessence of computer science, at the time when Hartree's notion of "numerical analysis" no longer served the purpose.

References to algorithms were commonplace in mathematics, even if they weren't particularly prominent. In fact, the term vaguely refers to the man from Khwarizm, al-Khwarizmi, who working in the house of wisdom in Bagdad in the early ninth century, wrote a textbook on how to complete and bring into balance numerical expressions, Hisab al-Jabr w' al-Muqabala. The man's name, Mohammad ibn Musa, was almost forgotten; as his learnings on how to manipulate numbers were handed down, they were identified by his nickname or by his novel technique of completion, al-Jabr. *Algebra* (the name for the mathematical field of solving equations) and *algorithm* (a word for a procedure of doing so) have stayed with the mathematicians since. However, by 1950, algebra had moved far away from practical arithmetic. As to algorithm, from the advent of heavy computing in the early 20th century, it came to mean an abstract mathematical idea rather than a procedure. In the 1920s computing real things, solving messy equations, came to be called practical mathematics or numerical analysis, and the techniques to do so were called procedures, schemes, or *Verfahren* (in German). This venerable notion of algorithm allowed computer users in the 1950s reflecting on developing computing procedures and transferring these to automatic machinery, to describe what they thought they were actually doing. Wasn't this the new concept they were looking for?

Indeed, a major strand in the emerging community of computer scientists embraced the notion of algorithms as quintessential to their field.[3] In his *Art of Computer Programming* published in 1968, a decade after the Zurich meeting, Donald Knuth wrote about the notion of algorithm: "The word 'algorithm' itself is quite interesting: at first glance it may look as though someone intended to write 'logarithm' but jumbled up the first four letters."[4] Knuth continued his search for roots a couple of years later in his paper "Ancient Babylonian Algorithms."[5]

Knuth was not the only one to fall in love with the notion of algorithm. The Dutch computer pioneer Aad van Wijngaarden devoted his career specifically not just to algorithms, but to algorithmic languages. His valedictory symposium in 1981 was called, accordingly, "Algorithmic Languages."[6] Even if his last piece of research, "Languageless Programming," departed from that devotion, it was presented at a conference entitled "Algorithms in Modern Mathematics and Computer Science," which was staged by his close colleagues Andrei Ershov and Donald Knuth who had "thought of making a scientific pilgrimage to the birthplace of al-Khwarizmi."[7]

To celebrate and acknowledge the role of Muhammad ibn Musa in the development of computer science, a conference was organized in Urgench, in the Khorezm Province of Uzbekistan.[8] Ample time was devoted to al'Khwarizmi's life. Heinz Zemanek, the Vienna based designer of the Mailüfterl computer, gave a full lecture about al'Khwarizmi's times, life, and works and even

continued on p. 102

 Published by the IEEE Computer Society 1058-6180/14/$31.00

continued from p. 104

proposed to vote on the year of his birth: "It is generally assumed that he was born around 780. We could take a vote that it was in 779 and celebrate his 1200th birthday today." With a selection of the world's leading computer scientists present in Urgench, the conference was the epitome of celebrating the heritage of al'Khwarizmi and constructing the algorithm genealogy of computer science.

One early adopter of the notion of algorithm when reflecting on automated computing in the early 1950s was the Swiss Heinz Rutishauser, who worked closely with Friedrich Bauer and Klaus Samelson in Munich. At the 1955 Darmstadt congress, he pled for a unified notation for the writing of programs, or *einheitliche algorithmische Schreibweise* (unified algorithmic notation): "And just like there is only a single symbol for integration in the whole world, we could finally arrive to a unified algorithmic notation."[9] In other words, he suggested that algorithmic notation should be developed in the same way as mathematical notation had arisen before, into a single notation, which would become a mathematical tool of the same kind as matrix algebra and integration.

As a result of the Darmstadt gathering, GaMM installed a committee including Bauer, Samelson, and Rutishauser on the unification of program notation. In 1957, that committee reported on an algorithmic notation (for formula translation).[10] Even if in the same years the notion of language, in the sense of programming language, was generally adopted,[11] Rutishauser did not combine the two notions. In 1957, Bauer went to the United States on a trip to meet the ACM community and collaborate on the future development of a unified programming language. He was accompanied by Hermann Bottenbruch, a pupil of Alwin Walther from Darmstadt and member of the same committee. Early in 1958, Bottenbruch finished his dissertation, "The Translation of Algorithmic Formulae Languages to Program-Languages for Computing Machinery."[12] Explicitly and amply referring to the work of Rutishauser in his plea for algorithmic notation and algorithmic thinking, Bottenbruch caught the two ideas floating in the air and combined them:

> I prefer the expression "algorithmic language" over the notion "algorithmic notation" introduced by Rutishauser in [12], [13]. In Rutishauser's work algorithmic notation is conceived as a specific algorithmic language, which includes algebraic formulae next to other expressions. Above all, I am indebted to publication [12] by Rutishauser, because it instigated my focus on "algorithmic thinking". Earlier attempts to develop a program out of a static system of equations failed. Yet, developing better algorithmic languages will not free us from the goal of developing programs directly from the static relations between given and sought quantities. In fact, one could speak of "automatic programming" in an actual sense only if the process were performed by a machine.[13]

Submitted for publication in March 1958, Bottenbruch's dissertation swiftly found its way to the press, but it does not seem to have been noted much. References to it are scarce. Yet, once phrased, the expression "algorithmic language" was immediately adopted by the European colleagues. The GAMM proposal for the Zurich meeting in May–June 1958, dated 9 May, opens with this phrase: "In the following, the first stage of an algorithmic language representing the basis of the formula translation project Zurich-Munchen-Mainz-Darmstadt is developed."[14] The GAMM proposal aims high, calling for a "universal language for the description of computing processes," while the ACM proposal remains practical and only aims at designing a programming language.[15] In Zurich in May–June 1958, the two initiatives merged into an "international standard for a common algebraic language for digital computers."[16] It remained an "international algebraic language" in the ACM, but in Europe, there was a definite motion toward an algorithmic language. Bauer recalls that his German colleagues adopted the term in an evening session and joked about "Algol, the star in the constellation Perseus, and ALGOL—algorithmic language." Interestingly, the abbreviation does not appear in Bottenbruch's dissertation.[17]

Meanwhile, a new force joined the European side of the initiative: Peter Naur from Regnecentralen in Copenhagen. Following up on a conference in Copenhagen in February 1959, Naur initiated a newsletter, the *Algol Bulletin.* Both the phrase and the acronym were adopted for the *Algol Bulletin* from the start. We assume that Samelson found the novelty clarifying and important enough to revise the title of the rendering of the Zurich report in *Numerische Mathematik,* which had originally been submitted in October 1958.

Algol was equally smoothly, although perhaps rather reluctantly, adopted by the English speaking part of the community: R.W. Bemer speaks of the first occurrence of Algol in the United States in August 1959.[18]

Thus, by the publication date of the Zurich report in *Numerische Mathematik* (December 1959), IAL had been changed to Algol and the algorithmic language per se was born. And the stars were positioned for "algorithm" to become a core notion of computer science.

References and Notes

1. A.J. Perlis and K. Samelson, "Preliminary Report–International Algebraic Language," *Comm. ACM*, vol. 1, no. 12, 1958, pp. 8–22.
2. A.J. Perlis and K. Samelson, "Report on the Algorithmic Language Algol by the ACM Committee on Programming Languages and the GAMM Committee on Programming," *Numerische Mathematik*, vol. 1, Dec. 1959, pp. 41–60.
3. N. Ensmenger, "The Rise of Computer Science," *The Computer Boys Take Over*, MIT Press, 2010, chap. 5, pp. 111–136.
4. D.E. Knuth, *The Art of Computer Programming*, vol. 1, Addison-Wesley, 1968, p. 1.
5. D.E. Knuth, "Ancient Babylonian Algorithms," *Comm. ACM*, vol. 15, no. 7, 1972, pp. 671–677.
6. J.W. de Bakker and J.C. van Vliet, eds., *Algorithmic Languages*, North Holland Publishing Company, 1981.
7. A.P. Ershov and D.E. Knuth, eds., *Proc. Algorithms in Modern Mathematics and Computer Science*, Springer, 1981.
8. H. Zemanek, "DIXIT ALGORIZMI. His background, His Personality, His Work, and His Influence," *Proc. Algorithms in Modern Mathematics and Computer Science*, A.P. Ershov and D.E. Knuth, eds., Springer, 1981, p. 24.
9. A. Walther and W. Hoffmann, eds., "Elektronische Rechenmaschinen und Informationsverarbeitung" [Electronic Digital Computers and Information Processing], *Nachrichtentechnische Fachberichte*, supplement of NTZ 4, Friedr. Vieweg & Sohn, Braunschweig, 1956, p. 143. Translation ours.
10. GAMM Committee on Programming, "Vorschläge für eine algorithmische Schreibweise zur Formelübersetzung" [Proposals for an Algorithmic Notation for Formula Translation], Oct. 1957.
11. D. Nofre, M. Priestley, and G. Alberts, "When Technology Became Language: The Origins of the Linguistic Conception of Computer Programming, 1950–1960," *Technology and Culture*, vol. 55, no. 1, 2014, pp. 40–75.
12. H. Bottenbruch, "Übersetzung von algorithmischen Formelsprachen in die Programmsprachen von Rechenmaschinen" [The Translation of Algorithmic Formulae Languages to Program-Languages of Computing Machinery], *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, vol. 4, 1958, pp. 180–221.
13. Bottenbruch, "Übersetzung von algorithmischen Formelsprachen in die Programmsprachen von Rechenmaschinen," p. 180. Translation ours. [12] refers to H. Rutishauser, "Automatische Rechenplanfertigung bei programmgesteuerten Rechenmaschinen" [Automatic Preparation of Computing Plans for Program Controlled Computing Machinery], Mitteilungen Nr. 3 aus dem Institut der angewandte Mathematik der ETH Zürich. Verlag Birkhäuser, Basel, 1952; and [13] to H. Rutishauser, "Maßnahmen zur Vereinfachung des Programmierens" [Measures to Simplify Programming], *Nachrichtentechnische Fachberichte*, supplement of NTZ 4, Friedr. Vieweg & Sohn, Braunschweig, 1956, pp. 26–31.
14. F.L. Bauer et al., "Proposal for a Universal Language for the Description of Computing Processes," *Computer Programming and Artificial Intelligence*, J.W. Carr, ed., Univ. of Michigan Summer School, 1958, pp. 355–373.
15. J.W. Backus et al., "Proposal for a Programming Language," ACM Ad Hoc Committee on Languages, 1958; www.softwarepreservation.org/projects/ALGOL/report/ACM_ALGOL_Proposal_1958.pdf.
16. A.J. Perlis and K. Samelson, "Report on a Proposed International Standard for a Common Algebraic Language for Digital Computers," *Computer Programming and Artificial Intelligence*, J.W. Carr, ed., Univ. of Michigan Summer School, 1958, pp. 375–401.
17. R. Wexelblatt, ed., *History of Programming Languages*, Academic Press, 1981, p. 128.
18. R.W. Bemer, "A Politico-Social History of Algol (with a Chronology in the Form of a Log Book)," *Ann. Rev. Automatic Programming*, M. I. Halpern and C.J. Shaw, eds., Pergamon Press, p. 166.

Helena Durnová *teaches mathematics and the history of mathematics at the Faculty of Education of Masaryk University, Brno, Czech Republic. Contact her at hdurnova@ped.muni.cz.*

Gerard Alberts *is an associate professor of the history of mathematics and computing in the Korteweg-de Vries Institute for Mathematics at the University of Amsterdam. Contact him at g.alberts@uva.nl.*