7. Curriculum '78—Is Computer Science Really that Unmathematical?

Anthony Ralston and Mary Shaw

In 1979, Ralston was investigating curricula for discrete mathematics [95] and Shaw was participating in evaluations of Curriculum '78 and the role of mathematics in undergraduate computer science. They combined their notes to form a criticism of the mathematical content of Curriculum '78 that appeared in Communications of the ACM [94]. Some comments on the paper appeared a few months later [69].

If computer science had not yet developed—significantly—as a science in the ten years between Curriculum '68 [1] and Curriculum '78 [3], then perhaps all those people who wondered if computer science was really a discipline would have been correct. In 1968 computer science was searching for but had not yet found much in the way of the principles and theoretical underpinnings which characterize a (mature) science. Ten years later, there is nothing laughable about calling computer science a science. This decade has seen major advances in the theory of computation and in the utility of theoretical results in practical settings. The rapid growth of the field of computational complexity has greatly increased our ability to analyze algorithms. And perhaps most significantly, we have finally started to make real progress in developing principles and theories for the design and verification of algorithms and programs.

Are these changes evident in Curriculum '78? Sadly, no. That curriculum only lends support to the (incorrect) equation

\[ \text{Computer Science} = \text{Programming} \]

that is mistakenly believed by so many outside the discipline. In the "Objectives of the Core Curriculum" [3] only the second objective—"be able to determine whether or not they have written a reasonably efficient and well-organized program"—recognizes that good programming requires more than just mastery of the syntax and semantics of a programming language. And even here the reference to principles and theory is, to be charitable, vague.

The principles and theories of any science give it structure and make it systematic. They should set the shape of the curriculum for that science, for

- only in that way can they provide a framework for the mastery of facts, and
- only in that way will they become the tools of the practicing scientist.

This is as true for computer science as it is for mathematics, for the physical
sciences, and for any engineering curriculum. Inevitably, for any science or any engineering discipline, the fundamental principles and theories can only be understood through the medium of mathematics. In the following sections we focus on the place of mathematics in the computer science curriculum and try to show how badly Curriculum '78 fails in this respect.

But first we note one matter of crucial importance which makes an emphasis on principles and theory even more important in computer science than in other disciplines. Computer science is an evolving field. Specific skills learned today will rapidly become obsolete. The principles that underlie these skills, however, will continue to be relevant. Only by giving the student a firm grounding in these principles can he or she be protected from galloping obsolescence. Even a student who aspires only to be a programmer needs more than just programming skills. He or she needs to understand issues of design, of the capability and potential of software, hardware, and theory, and of algorithms and information organization in general.

(Curriculum '68)  
M1 Introductory calculus  
M2 Mathematical analysis I  
M2P Probability  
M3 Linear algebra  
B3 Introduction to discrete structures  
B4 Numerical calculus  

plus two of  
M4 Mathematical analysis II  
M5 Advanced multivariate calculus  
M6 Algebraic structures  
M7 Probability and statistics  

(Curriculum '78)  
MA1 Introductory calculus  
MA2 Mathematical analysis I  
MA2A Probability  
MA3 Linear algebra  
MA4 Discrete structures  

(Required for some students)  
MA5 Mathematical analysis II  
MA6 Probability and statistics  

Figure 7-1: Required Mathematics Courses in ACM Curricula

7.1. Curriculum '78 and Mathematics

A comparison between the mathematics content of Curriculum '78 and that of Curriculum '68 is instructive. It reveals that

1. Whereas Curriculum '68 required the student to take eight (8) mathematics courses (see Figure 7-1), Curriculum '78 requires only five (5) mathematics courses.

2. The mathematics courses in Curriculum '68 formed an integral part of its prerequisite structure (see Figure 7-2). Note, in particular, for how many courses the discrete structures course (B3) is a prerequisite. In Curriculum '78, however, there is no mathematics prerequisite for any undergraduate computer science course with the exception of three advanced and clearly quite mathematical courses (only one of which