With a suitable setting of the parameters, we can even imitate atypewriter with its fixed width letters, as shown in Figure 13e. Thereis also a provision to slant the letters as in Figure 13f; here the penposition is varied, but the actual shape of the pen is not being slanted, so circular dots remain circles.

Finally, Figure 13i illustrates the variations you can get by giving weirder settings to the parameters. **In other words, there is a unique tangent at every point of the curve.** *This chapter and the next are for readers who share my fascination with original source documents.* ***The draft report I wrote during the first third of May was entitled TEXDR.AFT***

Several years ago when I began to look at the problem of designing suit-able alphabets for use with modern printing equipment, I found that25 of the letters were comparatively easy to deal with. The other letterwas S. For three days and nights I had a terrible time trying to under-stand how a proper S could really be defined. The solution I finally cameup with turned out to involve some interesting mathematics, and I be-lieve that students of calculus and analytic geometry may enjoy lookinginto the question as I did. The purpose of this paper is to explain whatI now consider to be the “right” mathematics underlying printed S’s,and also to give an example of the METAFONT language I have recentlybeen developing.* (A complete description of METAFONT, which is acomputer system and language intended to aid in the design of lettershapes, appears in [3, part 3].)

First part

Before getting into a technical discussion, I should probably mentionwhy I started worrying about such things in the first place. The centralreason is that today’s printing technology is essentially based on discretemathematics and computer science, not on the properties of metals orof movable type. The task of making a plate for a printed page is nowessentially that of constructing a gigantic matrix of 0s and 1s, where the0s specify white space and the 1s specify ink. I wanted the second editionof one of my books to look like the first edition, although the first editionhad been typeset with the old hot-lead technology; and when I realizedthat this problem could be solved by using appropriate techniques ofdiscrete mathematics and computer science, I couldn’t resist trying tofind my own solution.

Second part

After looking at these Renaissance constructions and a lot of mod-ern S shapes, I came to the conclusion that the main stroke of thegeneral S curve I sought would be analogous to the curve in Figure 6,each boundary curve was to be an ellipse followed by a straight linefollowed by another ellipse. This led me to pose the following problem:What ellipse has its topmost point at (xt, yt) and its leftmost point at(xl, yl) for some yl, and is tangent to the straight line of slope − thatpasses through (xc, yc), given the values of xt, yt, xl, , xc, and yc?(The ellipse in question is supposed to have the coordinate axes as itsmajor and minor axes; in other words, it should have left-right sym-metry. See Figure 7 on the next page.) The reason for my posing thisproblem should be fairly clear from our previous discussions: We know apoint that is supposed to be the top of the S curve, and we also know howfar the curve should extend to the left; furthermore we have a straightline in mind that will form the middle link of the stroke.

Sub-sect

By studying this example we can get some idea of the problems involved in specifying a proper S shape. However, I was actually seekingthe solution to a more general problem than the one Torniello faced:Instead of specifying only one particular S, I needed many different variations, including bold face S's that are much darker than the normal text. I discussed this recently with Alan Perlis, who pointed out that a central issue arising when we try to automate something properly is what he calls “the art of making constants variable.” In the case of letter design, we don’t merely want to take a particular drawing and come up with some mathematics to describe it; we really want to find the principles underlying the drawing, so that we can generate infinitely many drawings (including the given one) as a function of appropriate parameters. My goal was to create entire alphabets that would dependon a dozen or two parameters in such a way that all the letters wouldvary in a compatible manner as the parameters would change.

The problem stated in the preceding paragraph is interesting to me for several reasons. In the first place, it has an answer (as we will see) and in the second place, the answer does in fact lead to a satisfactory S curve. In the third place, the answer isn’t completely trivial; during a period of two years I came across this problem four different times and each time I was unable to find my notes about how to solve it, so I spent several hours rederiving the formulas whenever I needed them. Finally I decided to write this paper so that I wouldn’t have to derive them again.