The executioner’s argument was, that you couldn’t cut off a head unless there was a body to cut it off from: that he had never had to do such a thing before, and he wasn’t going to begin at his time of life.

The King’s argument was, that anything that had a head could be beheaded, and that you weren’t to talk nonsense.

The Queen’s argument was, that if something wasn’t done about it in less than no time, she’d have everybody executed, all round.

Alice’s Adventures in Wonderland

In this final chapter we’ll look at an example to show how the interpreter executes a command. I’ll also write down the full code, and describe a short interactive session using the interpreter. But first, let me state in more detail the subset of ABC that the interpreter can handle.
What the Interpreter will Recognize

The data structures recognized by the interpreter are tables, integer variables and integer constants. The statement syntax included is shown below:

PUT {} IN FRED
PUT whatever IN FRED [“Bill”]
PUT numeric value IN BERT
PUT x,y,z IN z,y,x

PUT algebra IN BERT
WRITE FRED [“Bill”]
WRITE FRED
WRITE keys FRED
WRITE x,y,z
WRITE 25+33*99/(4-5)

WHILE x < 100 : PUT x + 1 IN x
IF x = 25 : PUT y + 72 IN x

Plus three that are not standard ABC:

QUIT 

Template Matching

As we saw in the previous chapter, the syntax analysis must detect the statement keywords and create the corresponding Lisp expressions for subsequent evaluation. Many practical interpreters use a template matching technique which uses the keyword to select the template and then substitutes appropriately for the specific language statement used. The present interpreter uses this technique; I shall discuss its application by example rather than by exhaustive description.

Consider the ABC statement:

WHILE f < 100 : PUT f + y , f - y IN f , y

The syntax analysis (or parsing) of this statement starts at the left, and the keyword WHILE is detected. The statement structure for a WHILE is:

WHILE condition true EXECUTE the remaining ABC statements.

An equivalent in Common Lisp is:

(loop (if condition (progn remaining-statements) (return)))