```
print "QUANTUM!"
                                                                                                                 5.times do
                                                                                                                                                 end
                                                                                                                                  VS.
                                                                       pushq %rpp

.cfi_def_cfa_offset 16

.cfi_offset 6, -16

movq %rsp, %rbp

.cfi_def_cfa_register 6

subq $16, %rsp

movl $0, -4(%rbp)
                               main, @function
                                                                                                                                                                                                    $1, -4(%rbp)
                                                                                                                                                                     $.LCO, %edi
$0, %eax
printf
                                                                                                                                                                                                                            $4, -4(%rbp)
.string "QUANTUM!"
                                                                                                                                                                                                                                                         .cfi_def_cfa 7, 8
                                                              .cfi_startproc
                    .globl main
                                                                                                                                                                                                                                                                                 .cfi_endproc
                                                                                                                                                  jmp .L2
                                                                                                                                                                                                                                       jle .L3
                                .type
           .text
                                                                                                                                                                                                                                                   leave
                                                                                                                                                                        movl
                                                                                                                                                                                  movl
                                                                                                                                                                                                      addl
                                                                                                                                                                                                                             cmpl
                                                     .LFBO:
                                          main:
                                                                                                                                                              .I.3:
```

Figure 1.6 Contrasting a low-level and a high-level language for computer programs. The programs on the left and right perform the same task, but one is written in the low-level x86 assembly language and one in the high-level language Ruby.

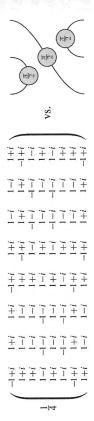


Figure 1.7 Contrasting a low-level and a high-level language for quantum processes, just like we contrasted the low-level and a high-level representation for digital data in Fig. 1.5 and a low-level and a high-level programming language in Fig. 1.6.

that by embracing the diagrammatic language for quantum theory, features like quantum teleportation are pretty much staring you in the face!

Although it goes beyond the scope of this book, it is worth mentioning that the diagrammatic language we use has found applications in other areas as well, such as modelling meaning in natural language (Fig. 1.8), doing proofs in formal logic, control theory, and modelling electrical circuits.

Diagrams are also becoming increasingly important in some fancy research areas of pure mathematics, such as knot theory, representation theory, and algebraic topology. By using diagrams we eliminate a huge amount of redundant syntactic garbage in representing