Appendix B

An exercise in Turing programming

B.1 Introduction

Recall that we defined two codes $\langle n \rangle$ and $[n]$ for $\mathbb{N}$ in Chapter 2.

As an exercise, we construct a machine T such that

$$T(\langle n \rangle) = [n].$$

Note that we say nothing of $T(p)$ for strings p *not* of the form $\langle n \rangle$. However, as we have seen, $\{\langle n \rangle : n \in \mathbb{N}\}$ is a maximal prefix-free subset of T. It follows that for any machine T satisfying our condition $T(p)$ must in fact be undefined for all p *not* of the form $\langle n \rangle$.

Our strategy in constructing T is to divide the task into 3 phases:

Reading First we read the input string $\langle n \rangle$ onto the tape.

Conversion Next we convert the string $\langle n \rangle$ on the tape into $B(n)$, by repeatedly dividing n by 2, retaining the quotient and remainder at each stage.

Suppose

$$n = 2^i m_i + r_i \quad (0 \leq r_i < 2^i).$$

Let $B_i(n)$ denote the last i bits of $B(n)$. Numerically, $B_i(n) = r_i$; but we have ensured that we have a string of length i by padding the beginning with 0's if necessary.

For example, if $n = 5$, $i = 2$ then

$$m_2 = 1,\ r_2 = 1,$$

while

$$B(n) = 101,\ B_2(n) = 01.$$

After the ith iteration, the tape will contains the string

$$\langle m_i \rangle 0 \cdots 0 \langle B_i n \rangle,$$

where there are a number of 0's (at least one) separating the left component $\langle m_i \rangle$ from the right component $\langle B_i(n) \rangle$.

Now

$$m_i = \begin{cases} 2m_{i+1} \text{ if } m_i \text{ is even,} \\ 2m_{i+1} + 1 \text{ if } m_i \text{ is odd,} \end{cases}$$

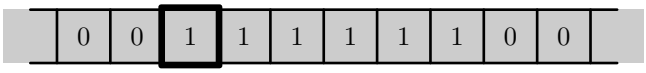
while

$$\langle B_{i+1} \rangle = \begin{cases} 10\langle B_i \rangle \text{ if } m_i \text{ is even,} \\ 11\langle B_i \rangle \text{ if } m_i \text{ is odd.} \end{cases}$$

In effect, m is halving at each iteration, while 10 or 11 is prepended to $\langle B_i(n) \rangle$ according as m is even or odd.

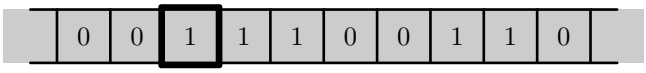
The iteration ends when m vanishes, leaving $\langle B(n) \rangle$ on the tape. We keep the first bit of $\langle m \rangle$ in the same position on the tape; so $\langle m \rangle$ is shrinking on the right, while $\langle B_i(n) \rangle 1$ is growing on its left by 2 bits at each iteration.

For example, suppose $n = 6$, so that $B(n) = 110$. Initially we have

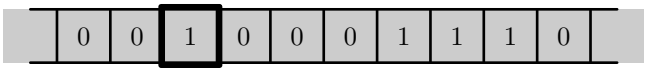


on the tape.

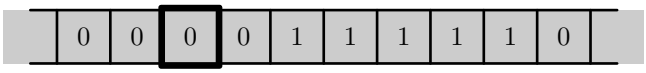
After the first step, $m = 3, r = 0, B_1(n) = 1$, and the tape reads



After the second step, $m = 1, r = 2, B_2(n) = 10$, and the tape reads



Finally, after the third step, $m = 0, r = 6, B_3(n) = B(n) = 110$, and the tape reads



We see that the tape contains the desired string $\langle B(n) \rangle$, although the scanner may be to the left of this.

Writing To conclude, we write out the tape contents $[n]$.

B.2 The reading phase

It is convenient to represent the rule

$$(q_i, b) \mapsto (a, q_j)$$

by the quadruple

$$(i, b, a, j).$$

With this convention, the following rules read the input string $\langle n \rangle$ onto the tape, with a 1 as a marker (whose purpose will shortly become apparent) to the right of $\langle m \rangle$.

$$(0, 0, \texttt{noop}, 1)$$
$$(1, 0, \texttt{swap}, 1)$$
$$(1, 1, \longleftarrow, 2)$$
$$(2, 0, \texttt{read}, 3)$$
$$(3, 1, \longleftarrow, 2)$$
$$(3, 0, \longrightarrow, 4)$$

Note that the fourth and fifth of these rules define a kind of loop; as long as we are reading 1's we will remain in the loop, leaving it only when we read a 0. At this point there will be $\langle n \rangle 1$ on the tape, and we will be looking at the 0 to the left of $\langle n \rangle$. The last rule brings the scanner back to the first bit of $\langle n \rangle$, with the machine in state q_4.

If $n = 0$ the tape will be blank (all 0's). Let us deal with this case at once. We must output 0 and halt:

$$(4, 0, \texttt{write}, 0)$$

(Recall that the machine halts on re-entering state q_0.)

B.3 The conversion phase

We are in state q_4, and the scanner is over the leftmost 1 of $\langle n \rangle$ on the tape. To avoid confusion, let us move to state q_5:

$$(4, 1, \texttt{noop}, 5)$$

This is the beginning of our big loop. Initially the tape holds $\langle n \rangle$. But at the start of subsequent loops the tape holds

$$\langle m \rangle 0 \cdots 0 \langle B_i(n) \rangle$$

with one or more 0's separating the two components.

We run through $\langle m \rangle$ two bits at a time.

$$(5, 1, \longrightarrow, 6)$$
$$(6, 1, \longrightarrow, 5)$$

We follow two different paths for a while, according as m is even or odd.

If m is even then we end in state q_5, with the scanner over what was the 0 at the end of $\langle m \rangle$. We want to insert 10 before $B(m)$. But first we must check if this is the first time round the loop, in which case the marker will serve.

$$(5, 0, \longrightarrow, 10)$$
$$(10, 1, \texttt{noop}, 30)$$

If this is not the first circuit, we must move to the beginning of $\langle B(m) \rangle$.

$$(10, 0, \texttt{noop}, 15)$$
$$(15, 0, \longrightarrow, 15)$$
$$(15, 1, \longleftarrow, 16)$$
$$(16, 0, \longleftarrow, 17)$$
$$(17, 0, \texttt{swap}, 30)$$

If m is odd then we end in state q_6, with the scanner over the 0 at the end of $\langle m \rangle$. We want to insert 11 before $B(m)$
. First we must delete the last 1 of $\langle m \rangle$. Then we must check if this is the

first time round the loop, in which case we insert a 1 after the marker.

$$(6, 0, \longleftarrow, 20)$$
$$(20, 1, \texttt{swap}, 20)$$
$$(20, 0, \longrightarrow, 21)$$
$$(21, 0, \longrightarrow, 22)$$
$$(22, 1, \longrightarrow, 23)$$
$$(23, 0, \texttt{swap}, 23)$$
$$(23, 1, \longleftarrow, 30)$$

If this is not the first circuit, we must move to the beginning of $\langle B(m)\rangle$

$$(22, 0, \texttt{noop}, 25)$$
$$(25, 0, \longrightarrow, 25)$$
$$(25, 1, \longleftarrow, 26)$$
$$(26, 0, \texttt{swap}, 26)$$
$$(26, 1, \longleftarrow, 27)$$
$$(27, 0, \texttt{swap}, 30)$$

Note that in all cases we are over the first bit of the newly installed 10 or 11. The two paths merge, and we return to the start of $\langle m\rangle$.

$$(30, 1, \longleftarrow, 31)$$
$$(31, 0, \longleftarrow, 31)$$
$$(31, 1, \texttt{noop}, 32)$$
$$(32, 1, \longleftarrow, 32)$$
$$(32, 0, \longrightarrow, 35)$$

We must check if $m = 1$, in which case our task is over; we only need to delete $\langle m\rangle$ and insert 11 before $\langle B(m)\rangle$.

$$(35, 1, \longrightarrow, 36)$$
$$(36, 0, \longleftarrow, 37)$$
$$(37, 1, \texttt{swap}, 37)$$
$$(37, 0, \texttt{noop}, 38)$$
$$(38, 0, \longrightarrow, 38)$$
$$(38, 1, \longleftarrow, 39)$$
$$(39, 0, \texttt{swap}, 39)$$
$$(39, 1, \longleftarrow, 40)$$
$$(40, 0, \texttt{swap}, 70)$$

If $m \neq 1$ we must replace $\langle m \rangle$ by $\langle \lfloor m/2 \rfloor \rangle$. First we replace each 11 by 10, deleting the extra 1 if m is odd.

$$(36, 1, \longleftarrow, 45)$$
$$(45, 1, \longrightarrow, 46)$$
$$(46, 1, \texttt{swap}, 47)$$
$$(47, 0, \longrightarrow, 45)$$
$$(46, 0, \longleftarrow, 48)$$
$$(48, 1, \texttt{swap}, 48)$$
$$(48, 0, \longleftarrow, 49)$$
$$(49, 0, \longleftarrow, 50)$$
$$(45, 0, \longleftarrow, 50)$$

Now we have converted $\langle m \rangle$ to a sequence of 10's, and are on the 1 of the last 10. We are going to repeatedly move this 1 to the leftmost 'hole', ie 0, available. After repeating this, we will have

$$11 \cdots 1010 \cdots 10$$

on the tape, with an even number of 1's at the beginning, followed by a sequence of 10's. But first we find the 'hole', by going backward over 10's Note that the procedure is slightly different according as this is the first or a subsequent repeat. In the first case we meet a 0 when expecting a 1. In the second case we meet a 1 when expecting a 0.

$$(50, 1, \longleftarrow, 51)$$
$$(51, 0, \longleftarrow, 50)$$
$$(50, 0, \texttt{noop}, 52)$$
$$(52, 0, \longrightarrow, 52)$$
$$(52, 1, \texttt{noop}, 55)$$
$$(51, 1, \texttt{noop}, 53)$$
$$(53, 1, \longrightarrow, 53)$$
$$(53, 0, \texttt{noop}, 55)$$

We check if we have finished this phase, ie there are no more 10's to the

right, in which case we go back to the start of our bigloop.

$$(55, 0, \longrightarrow, 56)$$
$$(56, 0, \texttt{noop}, 57) \qquad (57, 0, \longleftarrow, 57)$$
$$(57, 1, \texttt{noop}, 58)$$
$$(58, 1, \longleftarrow, 58) \qquad (58, 0, \longrightarrow, 5)$$

If this phase is not finished, we 'fill' the hole, go to the last 10 and delete the 1, and then return to the first 10. Again there are two paths, according as this tape component starts with 10 or 1110.

$$(56, 1, \longleftarrow, 60)$$
$$(60, 0, \texttt{swap}, 61)$$
$$(61, 1, \longrightarrow, 62)$$
$$(62, 0, \longrightarrow, 61)$$
$$(61, 0, \longleftarrow, 63)$$
$$(63, 1, \texttt{swap}, 63)$$
$$(63, 0, \longleftarrow, 64)$$
$$(64, 0, \longleftarrow, 65)$$
$$(65, 1, \longleftarrow, 64)$$
$$(64, 1, \longrightarrow, 55)$$
$$(65, 0, \texttt{noop}, 66)$$
$$(66, 0, \longrightarrow, 66)$$
$$(66, 1, \texttt{noop}, 5)$$

B.4 The writing phase

It only remains to write out the binary number on the tape. We are in state q_{70}, and the scanner is over the first 1 of $\langle B(n)\rangle$. We read through and print two bits at a time

$$(70, 1, \texttt{write}, 71)$$
$$(71, 1, \longrightarrow, 72) \qquad (72, 0, \texttt{write}, 71)$$
$$(72, 1, \texttt{write}, 71)$$
$$(70, 0, \texttt{write}, 0)$$