Analysis of Skelet's machine 17

March 5, 2023

See the transition table of the machine at `https://bbchallenge.org/1365166`. We will use the typical macro-machine notation where the Turing machine head lives in between the tape cells and can point either to the left or to the right.

Rules. We can reduce the analysis of behavior to two states only. When the machine is going right:

$$\texttt{B> 0} \to \texttt{<C 0}$$
$$\texttt{B> 10} \to \texttt{10 B>}$$
$$\texttt{B> 110} \to \texttt{101 B>}$$
$$\texttt{B> 111} \to \texttt{Halt}.$$

And when it is going left:

$$\texttt{1 <C} \to \texttt{<C 1}$$
$$\texttt{00 <C} \to \texttt{11 B>}$$
$$\texttt{010 <C} \to \texttt{<C 010}$$
$$\texttt{110 <C} \to \texttt{101 B>}.$$

In what follows we abbreviate `B>` by `>`, and `<C` by `<`.

For a nonnegative integer n we write $(10)^n$ to mean $1010\dots10$ where 10 is repeated n times. We define $(10)^0$ to be an empty word. When the machine head is to the right of the tape, the tape will have the form

$$(10)^{n_1}1(10)^{n_2}\dots1(10)^{n_k}$$

for some nonnegative $n_1, \dots, n_k$, as will be shown later. For example, after several simulation steps the tape and the machine head arrive at `11101010 B>` which can be written as $(10)^0 1(10)^0 1(10)^3 >$.

We have the following rules; here a period indicates the left end of the tape and n is assumed to be positive:

$$> (10)^n \to (10)^n >$$
$$> 1(10)^n \to (10)^1 1(10)^{n-1} >$$
$$1 < \to < 1$$
$$(10)^{2n} < \to < (10)^{2n}$$
$$1(10)^{2n+1} < \to (10)^1 1(10)^{2n} >$$
$$.(10)^{2n+1} < \to .11(10)^1 1(10)^{2n+1} >$$
$$. < \to .11 >$$

Observe the following corollary of the first two rules:

$$> (10)^a 1(10)^b \rightarrow> (10)^{a+1} 1(10)^{b-1},$$

and more generally

$$> (10)^{n_1} 1(10)^{n_2} \dots 1(10)^{n_{k-1}} 1(10)^{n_k} \rightarrow (10)^{n_1+1} 1(10)^{n_2} \dots 1(10)^{n_{k-1}} 1(10)^{n_k-1} >$$

In other words, when going to the right, the leftmost number is incremented and the rightmost number is decremented.

When going to the left, the machine passes 1s and even powers of 10 without any changes, but reverses the direction when it hits an odd power (see the fifth rule). The last two rules show what happens when the machine hits the left border of the tape.

It is easily seen from the rules above that if the tape and machine head had the form

$$.(10)^{n_1} 1(10)^{n_2} \dots 1(10)^{n_k} >$$

Then after one step the machine turns to the left:

$$.(10)^{n_1} 1(10)^{n_2} \dots 1(10)^{n_k} <$$

The head goes to the left until it hits an odd number or the tape border, and then returns back to the rightmost tape edge:

$$.(10)^{n'_1} 1(10)^{n'_2} \dots 1(10)^{n'_s} >$$

While the number of powers of 10 and respective exponents have changed, the tape still has the required form (or perhaps the machine halted). Since the head always returns to the right border, we will omit it from our formulae and the arrow $\rightarrow$ will denote one cycle of this left-and-right movement.

We introduce one last bit of formalism, the tape $.(10)^{n_1} 1(10)^{n_2} \dots 1(10)^{n_k}$ will be abbreviated to the list of numbers $n_1, n_2, \dots, n_k$. Now we show how the tape transforms after one machine cycle, each transformation is written both in tape and list format, and a short commentary is given afterwards. We start with short tapes, n is assumed to be positive.

$$\begin{gathered} .(10)^{2n} \rightarrow .11(10)^{2n} \\ 2n \rightarrow 0, 0, 2n \end{gathered} \tag{S1}$$

One even number adds two leading zeros.

$$\begin{gathered} .1(10)^{2n} \rightarrow .11(10)^1(10)^{2n-1} \\ 0, 2n \rightarrow 0, 0, 1, 2n-1 \end{gathered} \tag{S2}$$

One is moved to the left and two leading zeros are added.

$$\begin{gathered} .11(10)^{2n} \rightarrow \texttt{Halt} \\ 0, 0, 2n \rightarrow \texttt{Halt} \end{gathered} \tag{S3}$$

Two leading zeros and an even number results in halting; recall that the machine halts when `B> 111` occurs.

$$\begin{gathered} .(10)^{2n+1} \rightarrow .11(10)^1(10)^{2n} \\ 2n+1 \rightarrow 0, 0, 1, 2n \end{gathered} \tag{S4}$$

One is moved to the left and two leading zeros are added.

$$.1(10)^{2n+1} \to .(10)^1(10)^{2n} \qquad \text{(S5)}$$
$$0, 2n+1 \to 1, 2n$$

One is moved to the left.

$$.11(10)^{2n+1} \to .1(10)^1(10)^{2n} \qquad \text{(S6)}$$
$$0, 0, 2n+1 \to 0, 1, 2n$$

One is moved to the left.

We proceed with longer tapes. As with shorter tapes, the behavior mostly depends on the position of the odd number in the list. Here all the variables $n_1, n_2, \ldots$ are assumed to be positive.

$$.(10)^{2n_1}1(10)^{2n_2}\ldots 1(10)^{2n_k} \to .11(10)^{2n_1+1}1(10)^{2n_2}\ldots 1(10)^{2n_k-1} \qquad \text{(E1)}$$
$$2n_1, 2n_2, \ldots, 2n_k \to 0, 0, 2n_1+1, 2n_2, \ldots, 2n_k-1$$

One is moved from the rightmost number to the leftmost and two leading zeros are added.

$$.1(10)^{2n_1}1(10)^{2n_2}\ldots 1(10)^{2n_k} \to .11(10)^1 1(10)^{2n_1}1(10)^{2n_2}\ldots 1(10)^{2n_k-1} \qquad \text{(E2)}$$
$$0, 2n_1, 2n_2, \ldots, 2n_k \to 0, 0, 1, 2n_1, 2n_2, \ldots, 2n_k-1$$

One is moved from the rightmost number to the leftmost (which happens to be 0) and two leading zeros are added.

$$.11(10)^{2n_1}1(10)^{2n_2}\ldots 1(10)^{2n_k} \to \texttt{Halt} \qquad \text{(E3)}$$
$$0, 0, 2n_1, 2n_2, \ldots, 2n_k \to \texttt{Halt}$$

Two leading zeros and all even numbers result in halting, just as in the case of short tapes we considered earlier.

Now we consider what happens when there is an odd number in the list. Here expressions $2n_i$ are assumed to be nonzero, while expressions $2n_i+1$ may equal 1.

$$.(10)^{2n_1+1}1(10)^{2n_2}\ldots 1(10)^{2n_k} \to .11(10)^1 1(10)^{2n_1+2}1(10)^{2n_2}\ldots 1(10)^{2n_k-1} \qquad \text{(O1)}$$
$$2n_1+1, 2n_2, \ldots, 2n_k \to 0, 0, 1, 2n_1+2, 2n_2, \ldots, 2n_k-1$$

One is moved from the rightmost number to the leftmost and two leading zeros with one are added.

$$.1(10)^{2n_1+1}1(10)^{2n_2}\ldots 1(10)^{2n_k} \to .(10)^1 1(10)^{2n_1+1}1(10)^{2n_2}\ldots 1(10)^{2n_k-1} \qquad \text{(O2)}$$
$$0, 2n_1+1, 2n_2, \ldots, 2n_k \to 1, 2n_1+1, 2n_2, \ldots, 2n_k-1$$

One is moved from the rightmost number to the leftmost (which happens to be 0).

$$.11(10)^{2n_1+1}1(10)^{2n_2}\ldots 1(10)^{2n_k} \to .1(10)^1 1(10)^{2n_1+1}1(10)^{2n_2}\ldots 1(10)^{2n_k-1} \qquad \text{(O3)}$$
$$0, 0, 2n_1+1, 2n_2, \ldots, 2n_k \to 0, 1, 2n_1+1, 2n_2, \ldots, 2n_k-1$$

Variants of the previous rule with leading zeros are the same, one is moved from the rightmost number to the position to the left of the odd number.

Now, more generally, consider the rightmost odd number in the list (here m is positive):

$$\begin{gathered}\dots(10)^m 1(10)^{2n_1+1}1(10)^{2n_2}\dots 1(10)^{2n_k} \to \dots(10)^{m+1}1(10)^{2n_1+1}1(10)^{2n_2}\dots 1(10)^{2n_k-1} \\ \dots, m, 2n_1+1, 2n_2, \dots, 2n_k \to \dots, m+1, 2n_1+1, 2n_2, \dots, 2n_k-1\end{gathered} \tag{O4}$$

One is moved from the rightmost number to the position to the left of the rightmost odd number. In our case the rightmost odd number is $2n_1+1$, so one is added to m resulting in $m+1$, and one is subtracted from $2n_k$. In the case when the rightmost odd number is on the border, the computation is the same:

$$\begin{gathered}\dots(10)^m 1(10)^{2n_1+1} \to \dots(10)^{m+1}1(10)^{2n_1} \\ \dots m, 2n_1+1 \to \dots m+1, 2n_1\end{gathered} \tag{O5}$$

One sees that in general our rules decrease the rightmost number. When it hits zero, all numbers are shifted to the right but with a twist:

$$\text{if } n_1, \dots, n_k \to n'_1, \dots, n'_s \text{ then } n_1, \dots, n_k, 0 \to n'_1, \dots, n'_s + 1. \tag{O6}$$

The derivation of this is as follows:

$$\begin{aligned} .(10)^{n_1}1\dots 1(10)^{n_k}1(10)^0 &> \\ .(10)^{n_1}1\dots 1(10)^{n_k} &< 1 \\ .(10)^{n'_1}1\dots 1(10)^{n'_s} &> 1 \\ .(10)^{n'_1}1\dots 1(10)^{n'_s}10 &> \\ .(10)^{n'_1}1\dots 1(10)^{n'_s+1} &> \end{aligned}$$

Rules discussion. There is some redundancy in the rules, for example, (S1)–(S4) can be seen as special cases of (E1)–(E3) and (O1), respectively. Rules (O2), (O3) and (O5) are a special case of (O4). The redundant rules are provided to clarify some edge cases.

Note that the sum of numbers is not changed by all the rules for the exception of the "overflow rule" (O1), where the sum is increased by 1, and the "counter reset" rule (O6), where the sum may increase by 1 or 2. Indeed, before applying the rule (O6), we had the tape $n_1, \dots, n_k, 0$, where $n_k \geq 1$ as we applied rule (O5) before that. We arrive at the tape $n'_1, \dots, n'_s + 1$, where $n_1, \dots, n_k \to n'_1, \dots, n'_s$ may increase the sum by at most 1, since the rule (O6) does not apply to $n_1, \dots, n_k$. Therefore the sum $n'_1 + \dots + (n'_s + 1)$ is larger than $n_1 + \dots + n_k + 0$ by 1 or 2.

Also note that in almost every rule the rightmost number decreases by 1, for the exception of the halting rules, rule (S1) where it does not change, and the rule (O6) where it may decrease or increase.

Examples. After several steps the machine tape has the form `1110 B>` or $(10)^0 1(10)^0 1(10)^1 >$. In the list format that will be $0, 0, 1$, and now we show how to simulate the machine for a few cycles; on each line we write the tape and the relevant rule.

$0, 0, 1$	(S6)
$0, 1, 0$	(O6)

To apply (O6), note that $0, 1 \to 1, 0$ by the rule (S5):

$$\begin{array}{ll} 1,1 & (O4) \\ 2,0 & (O6) \end{array}$$

Again, note that $2 \to 0, 0, 2$ by (S1) hence:

$$\begin{array}{ll} 0,0,3 & (O4) \\ 0,1,2 & (O4) \\ 1,1,1 & (O4) \\ 1,2,0 & (O6) \end{array}$$

We counted down from 3 to 0, it is time to apply the reset rule (O6). Observe that $1, 2 \to 0, 0, 1, 2, 1$ by the overflow rule (O1), so:

$$\begin{array}{ll} 0,0,1,2,2 & (O4) \\ 0,1,1,2,1 & (O4) \\ 0,1,1,3,0 & (O6) \end{array}$$

And we reset the counter again. Let us fast forward the simulation and consider the tape configuration $(10)^1 1(10)^1 1(10)^5 1(10)^5$ which appears later:

$$\begin{array}{ll} 1,1,5,5 & (O4) \\ 1,1,6,4 & (O4) \\ 2,1,6,3 & (O4) \\ 2,1,7,2 & (O4) \\ 2,2,7,1 & (O4) \\ 2,3,7,0 & (O6) \end{array}$$

The rightmost number is counting down from 5 to 0, while the other numbers increase. The behavior of the machine is driven by parity changes of these numbers and they are not random, as we will quickly see.

It may appear that when the rightmost number is zero, the rest of the numbers are in nondecreasing order. Indeed, since the parity of the rightmost number is alternating after each rule, we increment the number next to it half of the time and it may seem that a similar reasoning applies to other positions. That is not the case, after 1016 cycles we have the tape

$$0, 0, 3, 2, 9, 16, 32, 0.$$

It may also seem that after each reset rule the counter variable is larger than it was after the previous reset rule. This is not true, the following tape appearing in the machine history

$$0, 1, 2, 4, 5, 14$$

counts down to

$$0, 2, 4, 8, 12, 0$$

which transforms to $0, 0, 1, 2, 4, 8, 12$ by rule (O6). The new counter 12 is less than 14. Empirically, the counter variable does seem to increase over time, though not monotonically.

Gray code. Recall that a Gray code is a way of ordering binary numbers such that every number differs from the previous one in one position only. For example, for numbers with three digits an example of a Gray code would be

$$\begin{array}{c} 000 \\ 001 \\ 011 \\ 010 \\ 110 \\ 111 \\ 101 \\ 100 \end{array}$$

One way of incrementing numbers in a Gray code is as follows: one keeps a special parity bit p which is flipped after every increment. If the parity bit p is 0, we flip the last digit of the number, if p is 1 we find the rightmost set digit of the number and flip the bit to the left of it. For example, if $p = 0$ then 011 increments to 010 and p is set to 1. To increment 010 we find the rightmost set bit (which is at position 2) and we flip the bit to the left of it (position 1) producing 110. Note that if we reverse the function of the parity bit, i.e. flip the last digit when $p = 1$ and find the rightmost set digit when $p = 0$, then we will decrement the number. For example, for 011 the rightmost bit is on the third position, so we flip the second bit and get 001 which the predecessor of 011 in our Gray code.

This is exactly what happens in our Turing machine. The rightmost number serves as a count-down variable and as a parity bit (depending on whether the number is odd or even), and all the other numbers serve as digits where odd numbers correspond to Boolean 1 and even numbers to 0. Rule (O4) finds the rightmost odd number (i.e. set bit) and flips the number to the left of it, the rule (O1) ensures that if the set bit overflows then we will extend the tape with zeros. In the case when the rightmost number (i.e. parity bit) is odd itself, we increment the number to the left of it, which corresponds to flipping the last digit of a binary number. For example, consider the following counting phase, here odd numbers are in bold:

$$\begin{array}{r} \mathbf{1}, 2, \mathbf{5}, 8 \\ \mathbf{1}, \mathbf{3}, \mathbf{5}, \mathbf{7} \\ \mathbf{1}, \mathbf{3}, 6, 6 \\ 2, \mathbf{3}, 6, \mathbf{5} \\ 2, \mathbf{3}, \mathbf{7}, 4 \\ 2, 4, \mathbf{7}, \mathbf{3} \\ 2, 4, 8, 2 \\ 0, 0, \mathbf{3}, 4, 8, \mathbf{1} \\ 0, 0, \mathbf{3}, 4, \mathbf{9}, 0 \end{array}$$

Here the machine is counting in the Gray code backwards moving from 101 to 111, then to 110 etc. This is because in the starting position the parity bit is 0 (as 8 is even) and the parity of the starting number is also even (as $1+0+1$ is even). In the opposite situation the machine will be counting in the Gray code forward, as in the following tape:

$$
\begin{gathered}
0,0,\mathbf{3},2,8 \\
0,\mathbf{1},\mathbf{3},2,\mathbf{7} \\
0,\mathbf{1},\mathbf{3},\mathbf{3},6 \\
0,\mathbf{1},4,\mathbf{3},\mathbf{5} \\
0,\mathbf{1},4,4,4 \\
\mathbf{1},\mathbf{1},4,4,\mathbf{3} \\
\mathbf{1},\mathbf{1},4,\mathbf{5},2 \\
\mathbf{1},\mathbf{1},\mathbf{5},\mathbf{5},\mathbf{1} \\
\mathbf{1},\mathbf{1},\mathbf{5},6,0
\end{gathered}
$$

Halting. The only halting rule is (E3), which says that the machine halts if it has two leading zeros and all the other numbers are even. If there is one leading zero, then no overflow can possibly happen and the number of leading zeros will not increase. We may obtain two leading zeros from rules (E1), (E2) or (O1), but in both cases the tape will have the form

$$0,0,1,2n_1,\dots,2n_{k-1},2n_k+1,$$

so the parity number is odd and exactly one digit is odd. That means that the machine will start counting down in the Gray code, and it seems crucial to prove that the countdown variable $2n_k+1$ should run out before the machine counts down to the all-zero number. Perhaps one way would be to provide a strong enough upper bound on the rightmost number in terms of the number of digits.

Another way to obtain two leading zeros is the reset rule (O6), for example, as in the following tapes:

$$
\begin{aligned}
2,2,4,8,16,0 &\quad \text{(O6)} \\
0,0,3,2,4,8,16 &
\end{aligned}
$$

When we apply (O6) to $n_1,\dots,n_k,0$, we may get two leading zeros only if $n_1,\dots,n_k \to n'_1,\dots,n'_s$ gives us two leading zeros. Since in the end the rightmost number will be n'_s+1, in this situation the parity bit will be different from the parity of $n'_1+\cdots+n'_{s-1}$, so the machine will start counting forward in the Gray code. This will result in the application of the rule (O3), so one leading zero disappears immediately and hence this case does not raise any problems.