Question:

Let

$$f(x) = \begin{cases} 3x+1 & \text{for odd } x \\ x/2 & \text{for even } x \end{cases}$$

for any natural number x. If you start with an integer x and iterate f, you obtain a sequence, x, f(x), f(f(x)), ... Stop if you ever hit 1. For example, if x = 17, you get the sequence 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1. Extensive computer tests have shown that every starting point between 1 and a large positive integer gives a sequence that ends in 1. But the question of whether all positive starting points end up at 1 is unsolved; it is called the $3_{x\ +\ 1}$ problem.

Suppose that A_{TM} were decidable by a TM H. Use H to describe a TM that is guaranteed to state the answer to the 3x + 1 problem.

Answer:

----SETP1----

The solution can be built in stages. Later stages will use Turing machines that are constructed in earlier stages. The first step is to construct a Turing machine M_{query} that takes an input x and accepts if iterating f starting from x eventually yields 1 and loops forever otherwise.

M_{query}. On input $\langle x \rangle$:

1. If $x = 1$, then accept.
2. If x is odd, update $x \leftarrow 3x+1$. If x is even, update $x \leftarrow \frac{x}{2}$.
3. Go to step 1.

Our next step is to construct a Turing machine M_{loop} which iterates over all positive integers, looking for a counter-example to the $3x+1$ conjecture. That is, M_{loop} searches for some x such that iterating f starting from x never reaches 1 and accepts if it finds such an x. To do this, it seems tempting to have M_{loop} simulated M_{query} first on $x=1$, next on $x=2$, and so forth. Whenever iterating f on x yields 1, the simulation of M_{query} will eventually end and M_{loop} would then proceed to the next number $x+1$. But what if M_{loop} actually finds a counter-example x? In this case, the simulation of M_{query} on x will never terminate, and M_{loop} will be in the unfortunate situation that it has found what it is looking for, but it doesn't know it has found it!

To get around this, use H. Instead of simulating M_{query} on x, It has M_{loop} check whether or not M_{query} would accept x by passing $\langle M_{query}, x \rangle$, to H.

M_{loop}. On input $\langle w \rangle$:

1. Ignore the input w.
2. For each natural number $y = 1, 2, 3, \ldots$:
3. Run H on $\langle M_{\text{query}}, y \rangle$.
4. If H rejects, then accept. Otherwise, continue the loop.

Finally, in order to solve the $3x+1$ problem, it is required to know whether or not M_{loop} finds a counter-example. Again, it might be tempted to simulate M_{loop} and see if it ever finds a counter-example and accepts.

The problem is that there may not be any counter-example, in which case M_{loop} will loop forever and our simulation will not terminate. The trick is to use H again to see if M_{loop} finds a counter example.

$M_{3x+1.}$ On input $\langle w \rangle$:

1. Ignore the input w.
2. Run H on $\langle M_{\text{loop}}, \varepsilon \rangle$.
3. If H accepts, then print ="There is a counter-example to the $3x+1$ conjecture."