

Turing machine example - equality

In this example, we design a Turing machine to check whether or not a string is of the form $w = w$. For instance, $1011 = 1011$ is to be accepted but $1011 = 1111$ is to be rejected.

Here is an algorithm that will work for any input, however large, by matching one character at a time on either side of the $=$ and marking what has been matched:

1. Note what is under the head, and mark it (as done) with an X.
2. Sweep right until the $=$ symbol, “remembering” whether a 0 or a 1 was noted.
3. Continue sweeping right until the first non-marked character, “remembering” what was noted. If that does not match what was noted, halt and reject. If that does match, mark that character as done with an X.
4. Sweep left until the $=$ symbol.
5. Continue sweeping until the first marked character, move right so that now the head is on the first character that has not yet been tested, and begin at step 1.

Each step above corresponds with exactly one way to react to what is under the head. They each, therefore, correspond with states.

1. Denote the start state by q_0 . No matter what is under the head, it marks it with an X and moves right. Now if the character under the head is a 0, it must switch to a different state, q_1 , than it would if it were 1 (q_3), otherwise this difference will not be noted.
2. q_1 and q_3 sweep right until $=$. This means that if the character is not $=$, both states retain the same character by replacing it with itself and moving right while remaining in the same state. However, if one is in q_1 it means that a 0 was marked, otherwise a 1 was marked. When it comes across $=$, q_1 will switch to q_2 while q_3 will switch to q_4 .
3. q_2 and q_4 sweep right until it finds a non-marked symbol. If that is a blank, it means that every character matched and was marked, so it will accept. Otherwise, on reaching a non-marked non blank character, it checks if what is now under the head matches with what was seen before. This is possible because if it is in q_2 it means that a 0 was marked, otherwise a 1 was marked. So if it is in q_2 rather than q_4 (which means that it needs to match a 0 rather than a 1) and finds the first non-marked character as 1,

it will reject. Otherwise, it will try to compare the next character. Now it no longer matters if a 0 or a 1 was under the head so both states switch to a state, q_5

4. q_5 turns left on every character that is not = and remains in the same state until it finds an =. When it finds =, it switches to q_6 . q_5 never changes any characters.
5. q_6 turns left for every non X character. When it reaches an X it turns right and switches to q_0 .

Tip: If you are viewing this on a smartphone, view the table and output below in landscape mode.

This is the transition table.

	0	1	X	=	—
q_0	(q_1, X, R)	(q_3, X, R)	(q_0, X, R)	$(\text{Accept}, =, R)$	$(\text{Reject}, =, L)$
q_1	$(q_1, 0, R)$	$(q_1, 1, R)$	$(q_1, 1, R)$	$(q_2, =, R)$	$(\text{Reject}, =, L)$
q_2	(q_5, X, R)	$(\text{Reject}, 1, R)$	(q_2, X, R)	$(\text{Reject}, =, R)$	$(\text{Reject}, =, L)$
q_3	$(q_3, 0, R)$	$(q_3, 1, R)$	(q_3, X, R)	$(q_4, =, R)$	$(\text{Reject}, =, L)$
q_4	$(\text{Reject}, 0, R)$	(q_5, X, R)	(q_4, X, R)	$(\text{Reject}, =, R)$	$(\text{Reject}, =, L)$
q_5	$(q_5, 0, L)$	$(q_5, 1, L)$	(q_5, X, L)	$(q_6, =, L)$	(Accept, E, R)
q_6	$(q_6, 0, L)$	$(q_6, 1, L)$	(q_0, X, R)	$(\text{Reject}, =, R)$	$(\text{Accept}, =, R)$

Here you can see the steps of the Turing machine in two examples; one which is rejected, and the other, which is accepted. Each row shows one snapshots of the tape along with the specific input and output of the transition function δ .

input: 1011=1111

```

: |1| 0 1 1 = 1 1 1 1 ...  $\delta:(q_0, 1) \rightarrow (q_3, X, R)$ 
: X |0| 1 1 = 1 1 1 1 ...  $\delta:(q_3, 0) \rightarrow (q_3, 0, R)$ 
: X 0 |1| 1 = 1 1 1 1 ...  $\delta:(q_3, 1) \rightarrow (q_3, 1, R)$ 
: X 0 1 |1| = 1 1 1 1 ...  $\delta:(q_3, 1) \rightarrow (q_3, 1, R)$ 
: X 0 1 1 |=| 1 1 1 1 ...  $\delta:(q_3, =) \rightarrow (q_4, =, R)$ 
: X 0 1 1 = |1| 1 1 1 ...  $\delta:(q_4, 1) \rightarrow (q_5, X, R)$ 
: X 0 1 1 = X |1| 1 1 ...  $\delta:(q_5, 1) \rightarrow (q_5, 1, L)$ 
: X 0 1 1 = |X| 1 1 1 ...  $\delta:(q_5, X) \rightarrow (q_5, X, L)$ 
: X 0 1 1 |=| X 1 1 1 ...  $\delta:(q_5, =) \rightarrow (q_6, =, L)$ 
: X 0 1 |1| = X 1 1 1 ...  $\delta:(q_6, 1) \rightarrow (q_6, 1, L)$ 
: X 0 |1| 1 = X 1 1 1 ...  $\delta:(q_6, 1) \rightarrow (q_6, 1, L)$ 
: X |0| 1 1 = X 1 1 1 ...  $\delta:(q_6, 0) \rightarrow (q_6, 0, L)$ 
: |X| 0 1 1 = X 1 1 1 ...  $\delta:(q_6, X) \rightarrow (q_0, X, R)$ 
: X |0| 1 1 = X 1 1 1 ...  $\delta:(q_0, 0) \rightarrow (q_1, X, R)$ 
: X X |1| 1 = X 1 1 1 ...  $\delta:(q_1, 1) \rightarrow (q_1, 1, R)$ 
: X X 1 |1| = X 1 1 1 ...  $\delta:(q_1, 1) \rightarrow (q_1, 1, R)$ 
: X X 1 1 |=| X 1 1 1 ...  $\delta:(q_1, =) \rightarrow (q_2, =, R)$ 

```

input: 1011=1011

3

```

:  X  X  X  X  =  |X|  X  X  1  . . .  δ : (q4, X) → (q4, X, R)
:  X  X  X  X  =  X  |X|  X  1  . . .  δ : (q4, X) → (q4, X, R)
:  X  X  X  X  =  X  X  |X|  1  . . .  δ : (q4, X) → (q4, X, R)
:  X  X  X  X  =  X  X  X  |1| . . .  δ : (q4, 1) → (q5, X, R)
:  X  X  X  X  =  X  X  X  X  |  | . . .  δ : (q5, _) → (Accept, E, R)
:  X  X  X  X  =  X  X  X  X  E  |  | . . .  Accept

```