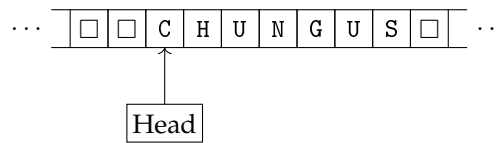


Things to remember

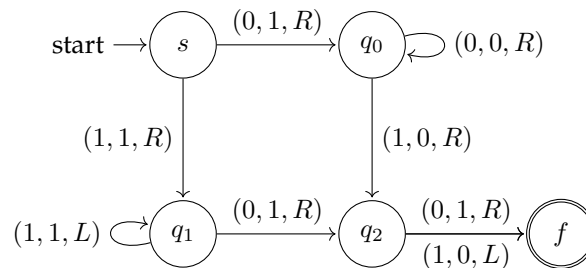
- Each Turing Machine must designate a **blank character**. In our course, the blank character is usually denoted '□'.
- Turing Machines can receive string inputs. On **input** w (w being some string), place w on the TM's tape (at the read/write head) before executing, and fill the rest of the infinite tape with □. For example, the tape should look as follows after accepting the input 'CHUNGUS':



- Each Turing Machine has a designated **finish state**. For us, the finishing state is called f . Upon reaching the finishing state, execution stops, and whatever string remains on the tape (excluding □ characters) is the **output**.

How to execute Turing machines

Each Turing machine has a DFA that governs its execution logic:



To execute the Turing Machine on a given input string:

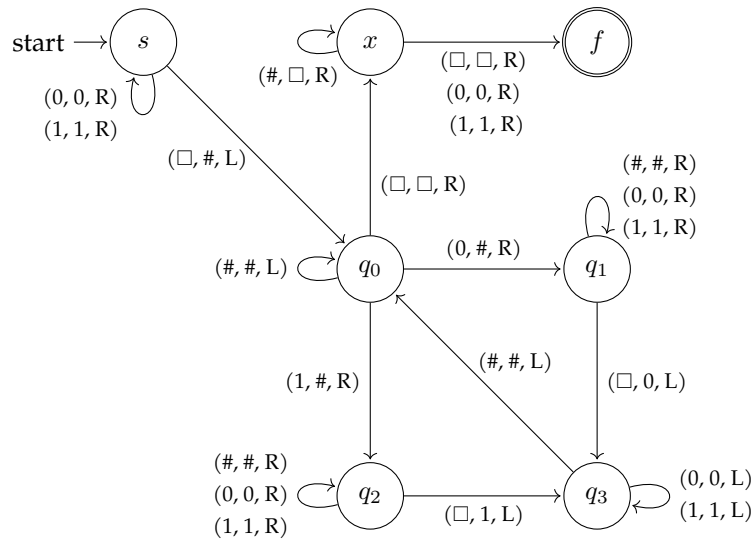
1. Place the input on the tape. Begin in the starting state (s in the above diagram).
2. Read one character from the read/write head.
3. Each arrow leaving the current state is labelled with a 3-tuple " (x, y, d) ". Find the arrow that has a label where the first entry x is the character you've just read in Step 2. Then:
 - Transition to the state that the arrow is pointing to.
 - Write the second character y on the label back to the tape.
 - Move the read/write head left or right based on the value of d on the label.
4. If the current state is not the finishing state, jump back to Step 2.

FAQ

- **Q:** After reading a character, can the TM choose to not write anything back to the tape?
A: No, but the TM can write the same character back to the tape, which has the same effect as not writing anything.
- **Q:** After reading a character, can the TM choose to not move the read/write head, as opposed to moving the head left or right?
A: Unfortunately, no. The TM must move the read/write head.
- **Q:** What happens if the TM never reaches the finish state?
A: If the TM never finishes execution, the TM is said to **loop**. There is no output in this case.
- **Q:** What's '#'?
A: This is just a character, like '0' or '1'.
- **Q:** What happens if the TM reads a character such as '#', and the current state doesn't have any transition that handles the character '#'?
A: The Turing Machine halts and catches on fire.

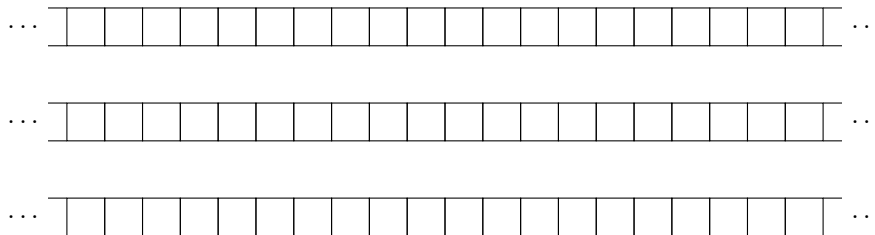
Jokes aside, this situation should never happen! When designing a Turing machine, we may be lazy and leave out transitions for characters that aren't expected in the given state. For example, in Exercise 1, while the TM is in the state s , there is no outgoing transition handling '#'; this is because we don't expect to see the character '#' while in state s .

Exercise 1. Consider the following Turing machine (with starting state s and finishing state f).



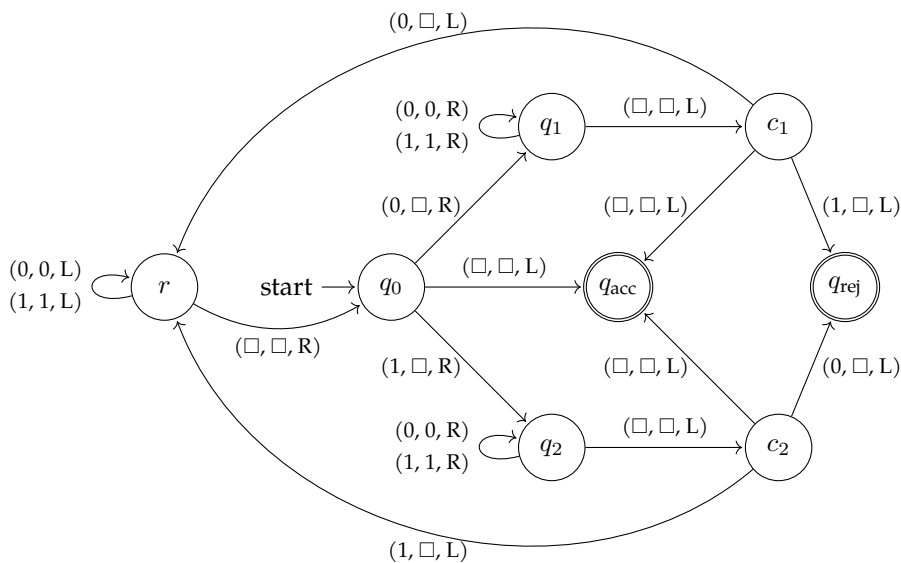
- (a) Execute the Turing machine on the following inputs, and determine the output:
- (i) "001".
 - (ii) "1101".
 - (iii) ϵ (the empty string).

Three blank tapes are provided below for your convenience.



- (b) Given a binary string w as input, what does the Turing machine output?

Exercise 2. Consider the following Turing machine (with starting state s , accepting state q_{acc} , and rejecting state q_{rej}).



Which input binary strings does the Turing machine accept?