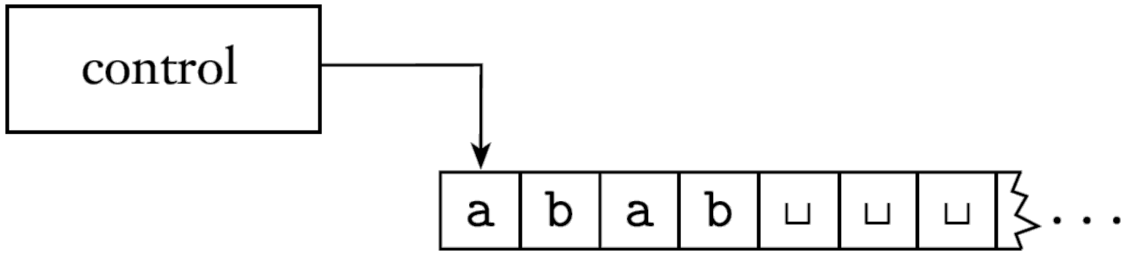
Turing Machines

Turing Machine (TM)

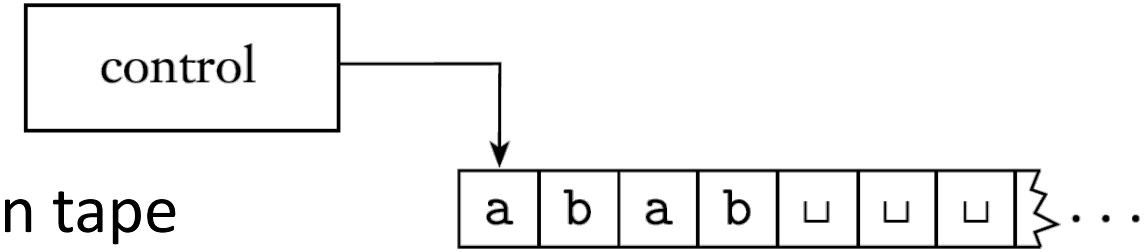
- Proposed by Alan Turing in 1936
- Similar to finite automaton
  - A finite state machine but with a tape which is used as unlimited and unrestricted memory
- Model of a general-purpose computer
  - Can do anything a real computer can do



- Memory is represented by an infinite tape
  - A controller moves along the tape to read and write symbols

How a Turing Machine Works

- Initially the tape contains only the input string
  - Blank everywhere else



- Machine can store information anywhere on tape
- The TM continues computing until it decides to produce an output
  - Has accepting and rejecting states
  - Continues forever, never halting, if it never reaches an accepting or rejecting state
- 3 Outcomes of TM
  - **Accept**, **Reject**, or **Infinite Loop**

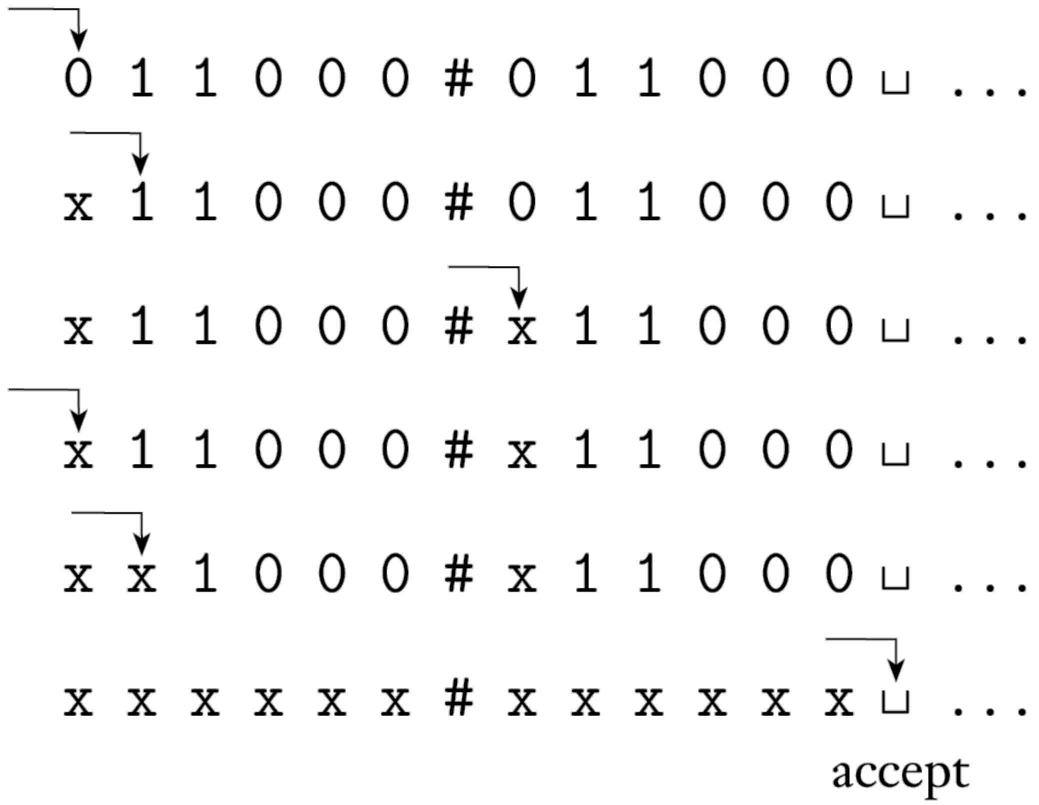
TM vs Finite Automata

1. TM can both write and read the tape
2. Controller can move both left and right along the tape
3. Tape is infinite
4. Rejecting and accepting states happen immediately

Example 1

- Given a TM that accepts a string in B = {w#w | w ∈ {0,1}*}

- Too long to remember all states
  - May move back and forth on tape
  - Compare relative position of inputs using # as reference
  - Matches crossed off with an x symbol
    - If mismatch, enter reject state
    - If all match, enter accept state



- TM computing on input
  - 011000#011000

Formal Definition for TM

- A **Turing machine** is a 7-tuple, $(Q, \Sigma, \Gamma, \delta, q_0, qac_{cept}, qre_{ject})$, where Q, Σ, Γ are all finite sets and
  - Q is the set of states,
  - Σ is the input alphabet not containing the **blank symbol** $\sqcup$,
  - Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$,
  - $\delta: Q \text{ x } \Gamma \rightarrow Q \text{ x } \Gamma \text{ x } \{\mathrm{L}, \mathrm{R}\}$ is the transition function,
  - $q_0 \in Q$ is the start state,
  - $q_{accept} \in Q$ is the accept state, and
  - $q_{reject} \in Q$ is the reject state, where $q_{accept} \neq q_{reject}$.

- Transition notation
  - δ(q,a) = (r,b,D)
    - q = current state
    - a = current symbol
    - b = symbol which overwrites a
    - r = new state
    - D = direction write head moves after rewriting symbol. (L or R)

- While formal definition is useful, it may be too large to describe.

Turing Machine Computation

- Given a TM $M = (Q, \Sigma, \Gamma, \delta, q_0, qac_{cept}, qre_{ject})$

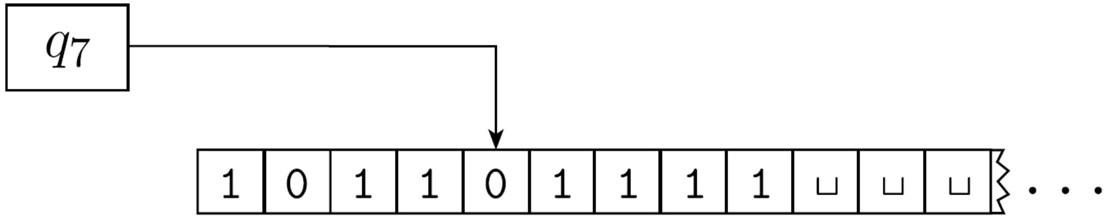
- Initially, M receives input w of length n and stores w on the n leftmost squares of tape
  - The rest of the tape is filled with blank spaces
    - 1st blank marks the end of input w
  - Head starts on the leftmost space on tape

- Transitions are described by transition function, δ, and will more either R or L
  - If head tries to move left off tape, it stays at 1st position
  - Continues until accept or reject state

Configurations

- As a TM computes, it changes 3 things over time:
  1. Current state
  2. Current tape contents
  3. Current head location

- The combination of the above 3 is called a **Configuration**
  - Notation: uqv
    - q = current state
    - uv = current stored tape string
      - The head location is the 1st symbol of v
    - Ex: uqv = $1011q_7 01111$
      - Current state = q_7
      - Tape contents = 101101111
      - Head location = 5th position (2nd zero in uv)



Configuration Transitions

- Configuration C_1, **yields** configuration C_2 if the TM goes from C_1 to C_2 in a single step

- Examples
  - $\delta(q_i,b) = (q_j,c,L)$
    - uaq_ibv **yields** uq_jacv
    - Head pointing to b, replace with c, move left, change state
  - $\delta(q_i,b) = (q_j,c,R)$
    - uaq_ibv **yields** $uacq_jv$
    - Head pointing to b, replace with c, move right, change state

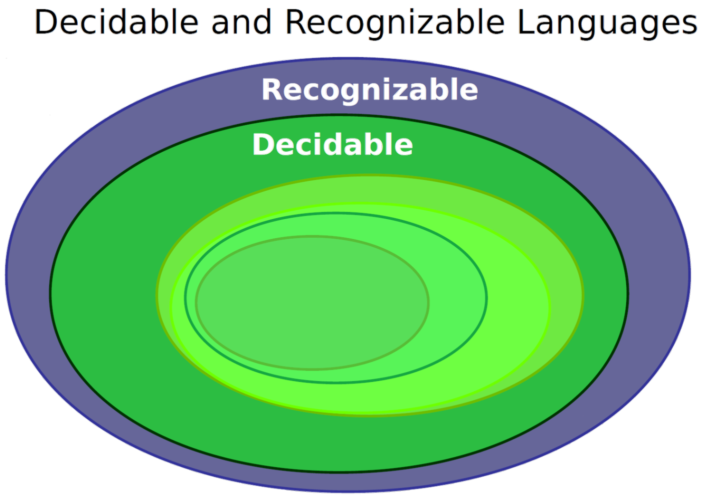
- Special cases at either end of the tape contents
  - If the head is in the left most position, $\delta(q_i,b) = (q_j,c,L)$
    - q_ibv **yields** q_jcv, head does not move
  - If the head is in the right most position, $\delta(q_i,b) = (q_j,c,R)$
    - uaq_i **yields** $uacq_j$, b was a blank space

Configuration Terms

- Start Configuration
  - Q_0w
- Accepting configuration
  - Configuration with the accepting state as current state
- Rejecting configuration
  - Configuration with the rejecting state as current state
- Halting configuration
  - Either accepting or rejecting configuration which halts TM

Recognizable vs Decidable

- A language is **Turing-recognizable** if some Turing machine recognizes it
  - A collection of strings that TM, M, accepts is the **language of M**
    - The language recognized by M, L(M)



- Difficult to tell if a TM is loop indefinitely
  or just taking long to reach a halting configuration
  - Useful to have a TM that will never loop
    - **Decider**

- A language is **Turing-decidable** if it is recognized by some TM that is a decider
  - When a decider recognizes a language it is said it **decides** that language

- Every Turing-decidable language is Turing-recognizable