Theory of Formal Languages and Automata

Lecture 16

Mahdi Dolati

Sharif University of Technology

Fall 2023

November 27, 2023

Turing Machine

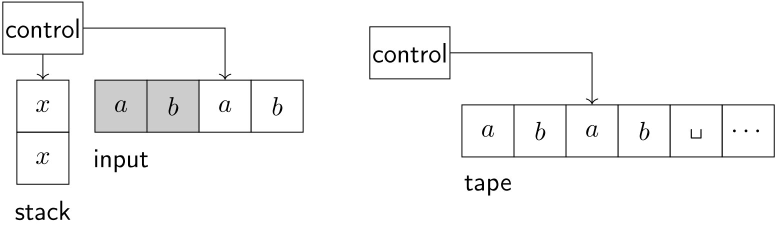
Models of computation:

- DFA: Deterministic, No temporary memory,
- NFA: Nondeterministic, No temporary memory,
- PDA: Nondeterministic, Infinite stack,
- Turing machine (TM): Deterministic, One-way infinite read-write tape.
  - Models of general purpose computers.
  - First proposed by Alan Turing in 1936.

Turing Machine

Turing machines: Power equivalent to a computer,

- Finite set of states,
- Initial state,
- Two special states:
  - Accept,
  - Reject.
- A tape of one-way infinite cells: Initially contains only the input string and is blank everywhere else.
- A tape head that can move to left and right.



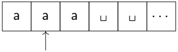
Turing Machine

Differences between finite automata and Turing machines:

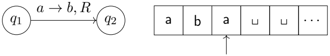
- A Turing machine can both write on the tape and read from it.
- The read-write head can move both to the left and to the right.
- The tape is infinite.
- The special states for rejecting and accepting take effect immediately.

Turing Machine

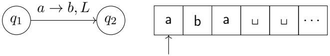
Transitions:



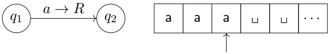
- Replace a with b and move right:



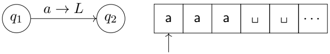
- Replace a with b and move left:



- On a move right:



- On a move left:

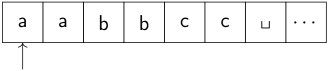


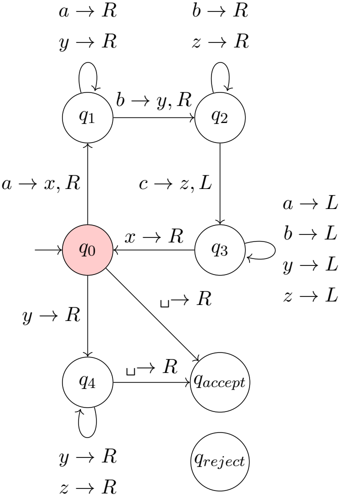
- Determine whether an input is a member of $A = \{a^n b^n c^n \mid n \geq 0\}$,
- The input is too long for you to remember it all, but you are allowed to move back and forth over the input and make marks on it.
- On each pass it matches one a with one b and one c. To keep track of which symbols have been checked already, the machine replaces them with other symbols (x, y, and z) as they are examined. If it replaces all the symbols, that means that everything matched successfully, and the machine goes into an accept state. If it discovers a mismatch, it enters a reject state.

Turing Machine

$A = \{a^n b^n c^n \mid n \geq 0\}$

w=aabbcc

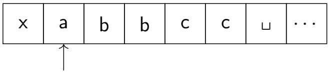


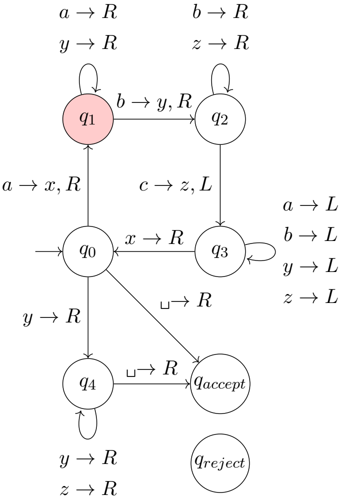


Turing Machine

$A = \{a^n b^n c^n \mid n \geq 0\}$

w=aabbcc

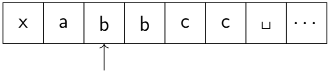


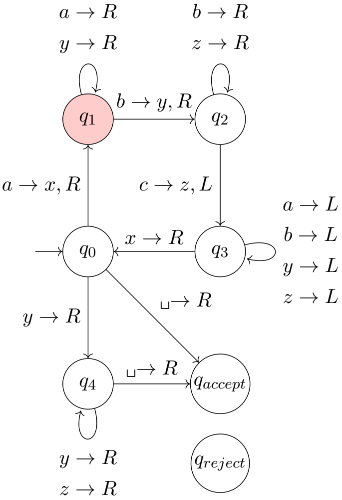


Turing Machine

$A = \{a^n b^n c^n \mid n \geq 0\}$

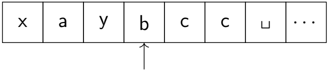
w=aabbcc

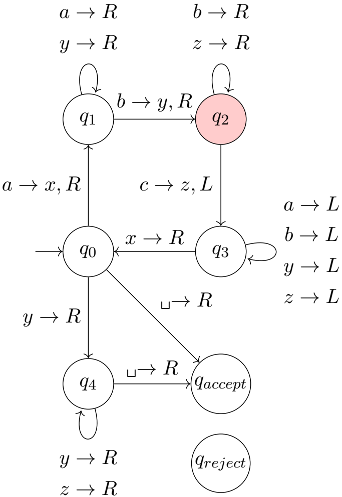




$A = \{a^n b^n c^n \mid n \geq 0\}$

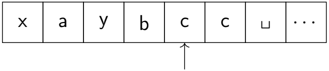
w=aabbcc

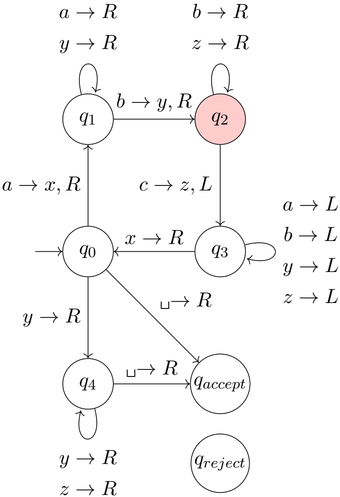




Turing Machine

$A = \{a^n b^n c^n \mid n \geq 0\}$
w=aabbcc

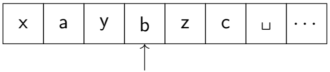


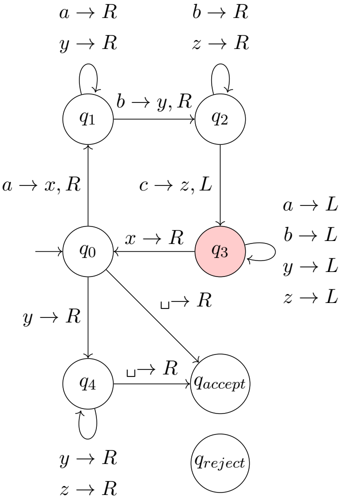


Turing Machine

$A = \{a^n b^n c^n \mid n \geq 0\}$

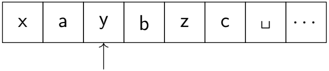
w=aabbcc

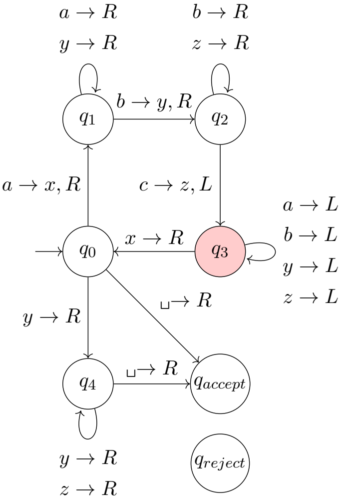




$$A = \{a^n b^n c^n \mid n \geq 0\}$$

w=aabbcc

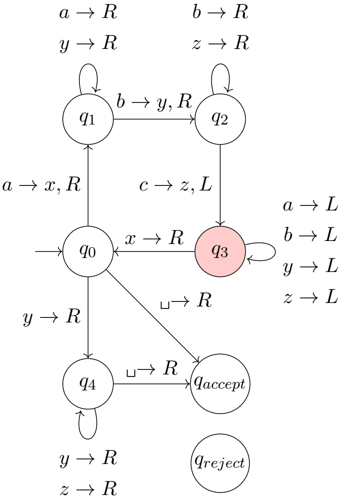




$A = \{a^n b^n c^n \mid n \geq 0\}$

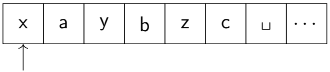
w=aabbcc

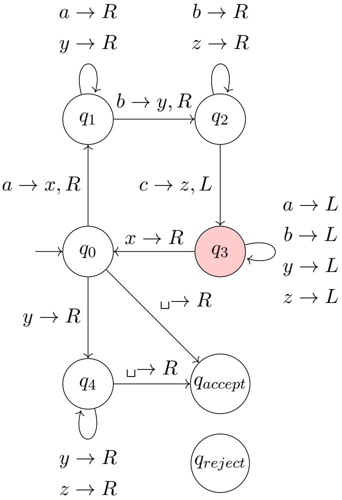
x	a	y	b	z	c	␣	···



$A = \{a^n b^n c^n \mid n \geq 0\}$

w=aabbcc

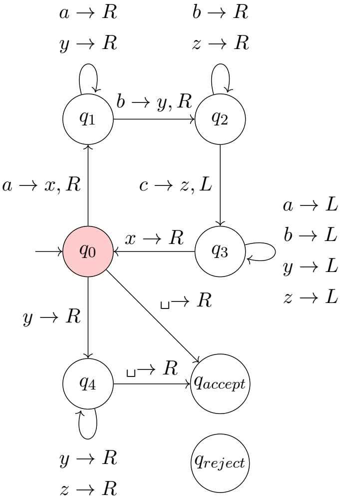




$A = \{a^n b^n c^n \mid n \geq 0\}$

w=aabbcc

x	a	y	b	z	c	␣	⋯

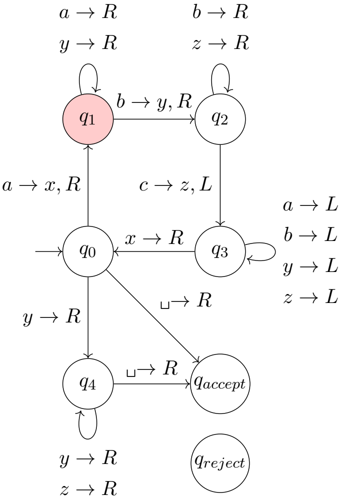


Turing Machine

$A = \{a^n b^n c^n \mid n \geq 0\}$

w=aabbcc

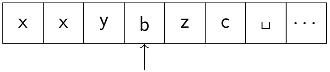
x	x	y	b	z	c	␣	⋯
		↑					

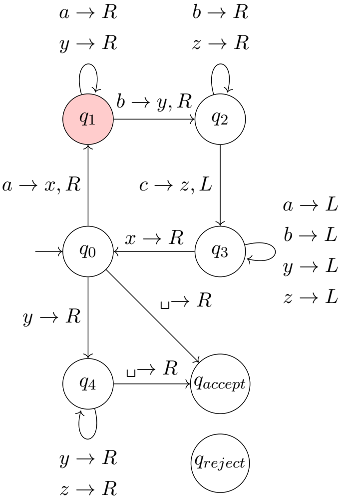


Turing Machine

$A = \{a^n b^n c^n \mid n \geq 0\}$

w=aabbcc



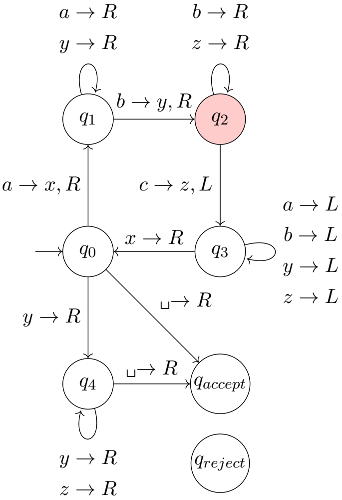


Turing Machine

$$A = \{a^n b^n c^n \mid n \geq 0\}$$

w=aabbcc

x	x	y	y	z	c	␣	⋯

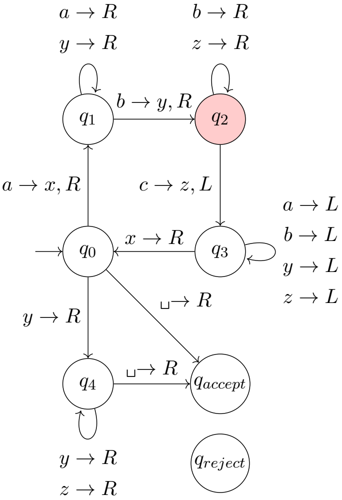


Turing Machine

$$A = \{a^n b^n c^n \mid n \geq 0\}$$

w=aabbcc

x	x	y	y	z	c	␣	⋯

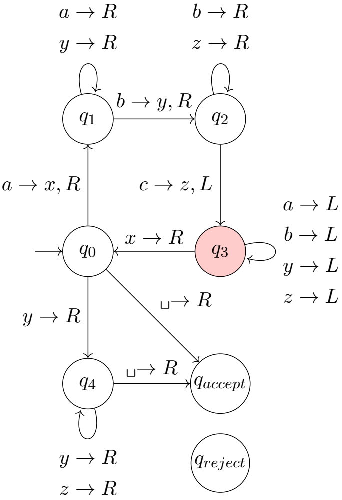


Turing Machine

$A = \{a^n b^n c^n \mid n \geq 0\}$

w=aabbcc

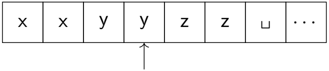
x	x	y	y	z	z	␣	⋯

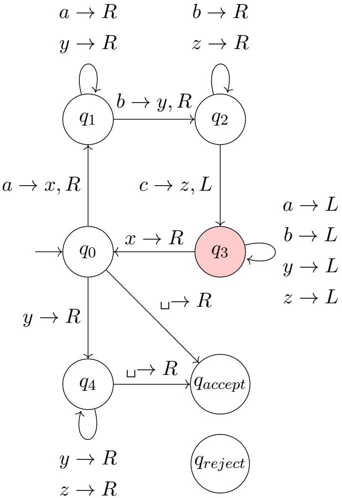


Turing Machine

$A = \{a^n b^n c^n \mid n \geq 0\}$

w=aabbcc

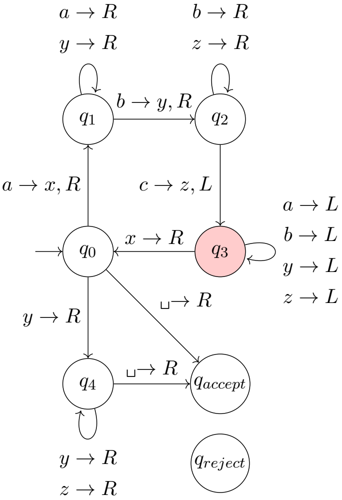




Turing Machine

$A = \{a^n b^n c^n \mid n \geq 0\}$
w=aabbcc

x	x	y	y	z	z	␣	···
		↑					

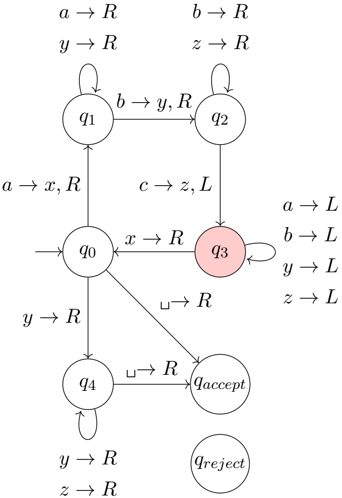


Turing Machine

$$A = \{a^n b^n c^n \mid n \geq 0\}$$

w=aabbcc

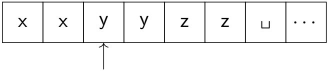
x	x	y	y	z	z	␣	...

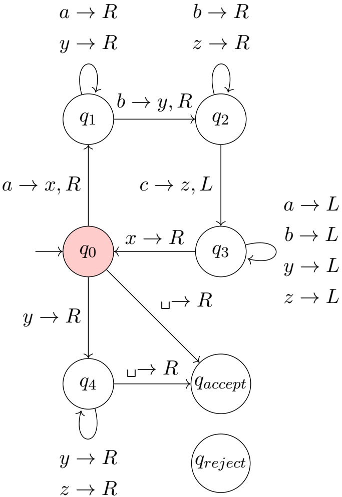


Turing Machine

$A = \{a^n b^n c^n \mid n \geq 0\}$

w=aabbcc

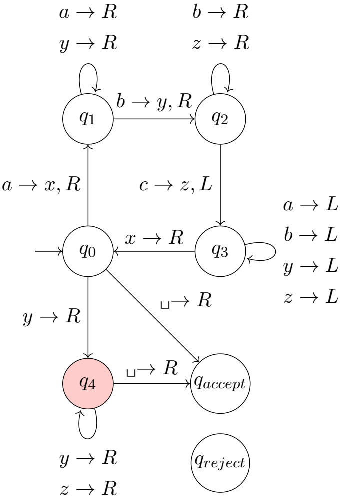




$$A = \{a^n b^n c^n \mid n \geq 0\}$$

w=aabbcc

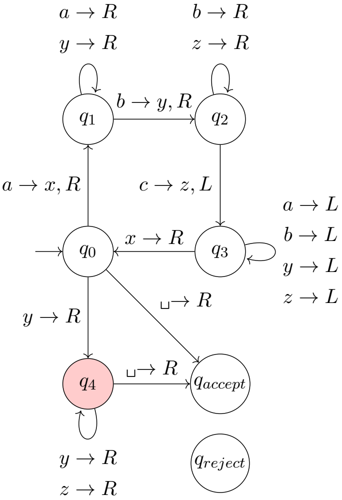
x	x	y	y	z	z	␣	...
			↑				



$$A = \{a^n b^n c^n \mid n \geq 0\}$$

w=aabbcc

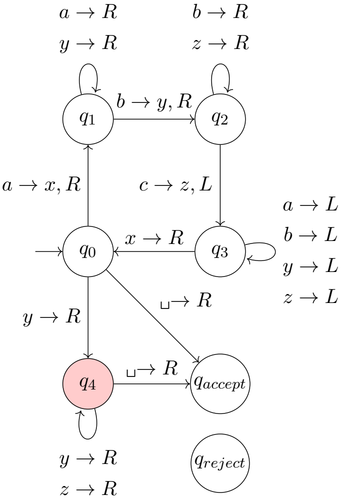
x	x	y	y	z	z	␣	⋯



$A = \{a^n b^n c^n \mid n \geq 0\}$

w=aabbcc

x	x	y	y	z	z	␣	···

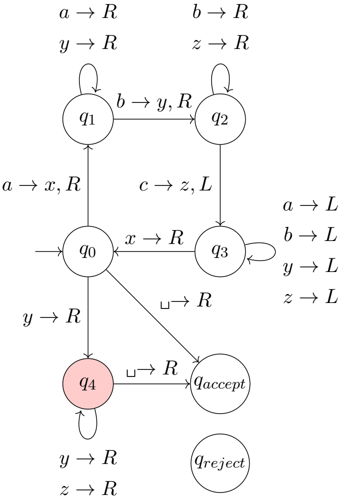


Turing Machine

$A = \{a^n b^n c^n \mid n \geq 0\}$

w=aabbcc

x	x	y	y	z	z	␣	$\cdots$

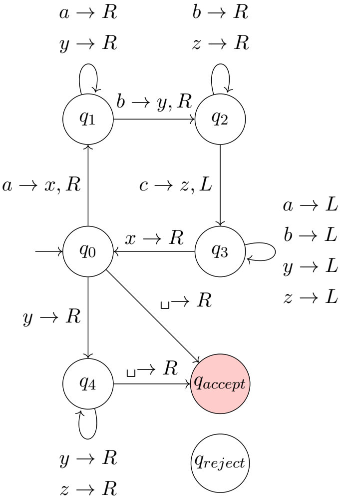


Turing Machine

$A = \{a^n b^n c^n \mid n \geq 0\}$

w=aabbcc

x	x	y	y	z	z	␣	...



Turing Machine

Definition

A Turing Machine is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ where

- Finite set of states: Q
- Input alphabet: Σ ($\sqcup \notin \Gamma$)
- Tape alphabet: Γ ($\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$)
- Transition function: δ
  - $\delta : (Q - \{q_{accept}, q_{reject}\}) \times \Gamma \to Q \times \Gamma \times \{L, R\}$,
- Initial state: $q_0 \in Q$
- A single accepting state: $q_{accept} \in Q$
- A single rejecting state: $q_{reject} \in Q$ and $q_{accept} \neq q_{reject}$

- The machine halts after entering q_{accept} or q_{reject} and there is no transition out of them.
- In actuality, we almost never give formal descriptions of Turing machines because they tend to be very big.

Turing Machine

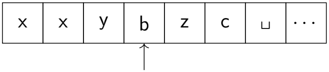
- Transition function has a transition from every state other than accepting and rejecting states and for every tape symbols $s \in \Gamma$:
  - $\delta(q, s) = (r, t, D)$,
  - If in state q and the tape head points at s,
  - Then, go to state r,
  - replace s with t in tape, and
  - move the tape head to the direction $D \in \{L, R\}$

- M receives its input $w = w_1 w_2 \dots w_n \in \Sigma^*$ on the leftmost n squares of the tape, and the rest of the tape is blank.
- The head starts on the leftmost square of the tape.
- The first $\sqcup$ appearing on the tape marks the end of the input.
- The computation proceeds according to the rules described by the transition function.
- If M ever tries to move its head to the left off the left-hand end of the tape, the head stays in the same place for that move, even though the transition function indicates L.
- The computation continues until it enters either the accept or reject states, at which point it halts. If neither occurs, M goes on forever.

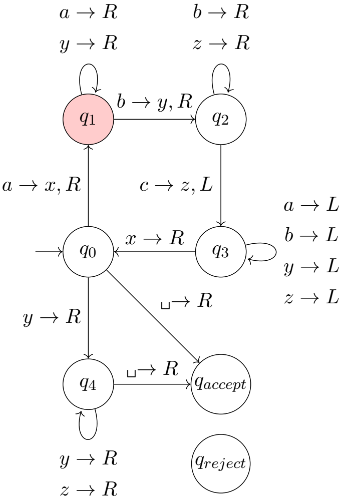
- As a Turing machine computes, changes occur in:
  - the current state,
  - the current tape contents, and
  - the current head location.
- A setting of these three items is called a configuration of the Turing machine:

$$uqv,$$

  - The current state is q,
  - The current tape content is uv,
  - The current head location is the first symbol of v.



Configuration = xxy$q_1$bzc



- As a Turing machine computes, changes occur in:
  - the current state,
  - the current tape contents, and
  - the current head location.
- A setting of these three items is called a configuration of the Turing machine:

$$uqv,$$

  - The current state is q,
  - The current tape content is uv,
  - The current head location is the first symbol of v.

- Say that configuration C1 yields configuration C2 if the Turing machine can legally go from C1 to C2 in a single step:
- Consider $a, b, c \in \Gamma$, $u, v \in \Gamma^*$, and $q_i, q_j \in Q$:
    - Leftward transition:

      $$\delta(q_i, b) = (q_j, c, L) \rightarrow uaq_ibv \text{ yields } uq_jacv$$

    - Leftward transition in the left-hand end:

      $$\delta(q_i, b) = (q_j, c, L) \rightarrow q_ibv \text{ yields } q_jcv$$

    - Rightward transition in the left-hand end:

      $$\delta(q_i, b) = (q_j, c, R) \rightarrow q_ibv \text{ yields } cq_jv$$

- Say that configuration C1 yields configuration C2 if the Turing machine can legally go from C1 to C2 in a single step:
- Consider $a, b, c \in \Gamma$, $u, v \in \Gamma^*$, and $q_i, q_j \in Q$:
  - Rightward transition:

    $$\delta(q_i, b) = (q_j, c, R) \rightarrow uaq_ibv \text{ yields } uacq_jv$$

  - Leftward transition in the right-hand end:

    $$\delta(q_i, b) = (q_j, c, L) \rightarrow uaq_ib \text{ yields } uq_jac$$

  - Rightward transition in the right-hand end:

    $$\delta(q_i, b) = (q_j, c, R) \rightarrow uq_ib\sqcup \text{ yields } ucq_j\sqcup$$

- Start configuration: $q_0 w$,
- Accepting configuration: uq_iv where $u, v \in \Gamma^*$ and $q_i = q_{accept}$,
- Rejecting configuration: uq_iv where $u, v \in \Gamma^*$ and $q_i = q_{reject}$,
- Halting configuration: Accepting and rejecting configurations,
  - Do not yield further configurations,

Definition

Turing machine M accepts input w if a sequence of configurations C_1, C_2, . . . , C_k exists, where

1. C_1 is the start configuration of M on input w,
2. each C_i yields C_{i+1}, and
3. C_k is an accepting configuration.

Definition

A language is Turing-recognizable (or recursively enumerable) if some Turing machine recognizes it.

- The Turing machine should accept all strings in the language and reject the rest (or fail to accept them).
- The outcome of running a Turing machine on an inputs may be accept, reject, or loop.
  - Reject by entering the q_{reject} or by looping.

Turing Machine

- Turing machines that halt on all inputs are called deciders.
- A decider that recognizes some language also is said to decide that language.

Definition

A language is Turing-decidable or simply decidable (or recursive) if some Turing machine decides it.

- Every decidable language is Turing-recognizable.
- There exists a language that is Turing-recognizable but undecidable.

Turing Machine

- Being decidable is a property of the language, not of a Turing machine.
- Consider a decidable language L and Turing machine M that decides L:
  - We can transform all rejects to a loop:
    - Create a new state q_{loop},
    - Change all transitions to q_{reject} to q_{loop} instead,
    - In state q_{loop}, go to right for any symbol on the tape.
  - The resulting Turing machine is no longer a decider.
  - However, the language of the Turing machine does not change.
  - The language is still Turing decidable.

Turing Machine

- Regular languages are decidable:
- Consider a regular language A with DFA $M = (Q, \Sigma, \delta_0, F)$ such that $L(M) = A$.
- We construct a Turing machine $M^{'} = (Q^{'}, \Sigma, \Gamma, q_0, q_{accept}, q_{reject})$:
  - $Q^{'} = Q \cup \{q_{accept}, q_{reject}\}$,
  - $\Gamma = \Sigma \cup \sqcup$,
  - The transition function:

$$\delta^{'}(q,t) = \begin{cases} (\delta(q,t), t, R) & \text{if } t \in \Sigma \\ (q_{accept}, \sqcup, R) & \text{if } t = \sqcup \text{ and } q \in F \\ (q_{reject}, \sqcup, R) & \text{if } t = \sqcup \text{ and } q \notin F \end{cases}$$

  - $M^{'}$ reads the input and changes the state according to δ. When the input is read, $M^{'}$ accepts or rejects based on the state of M. Thus, $L(M^{'}) = L(M)$.
  - $M^{'}$ moves to right after reading each symbol and halts once it reaches a $\sqcup$. Thus, $M^{'}$ is a decider.

Turing Machine

- We can formally describe a particular Turing machine by specifying each of its seven parts.
- However, it can be cumbersome for all but the tiniest Turing machines.
- Accordingly, we won't spend much time giving such descriptions.
- Mostly we will give only higher level descriptions.
- Nevertheless, it is important to remember that every higher level description is actually just shorthand for its formal counterpart. With patience and care we could describe any of the Turing machines in this book in complete formal detail.

Example

Describe a Truing machine for $A = \{0^{2^n} \mid n \geq 0\}$.
Consider input string w:

- Sweep left to right across the tape, crossing off every other 0.
  - Cut the number of 0s in half.
- If in stage 1 the tape contained more than a single 0 and the number of 0s was odd, reject.
  - keeps track of whether the number of 0s seen is even or odd.
  - If the number is odd, the original input could not have been a power of 2.
- If in stage 1 the tape contained a single 0, accept.
- Return the head to the left-hand end of the tape.
- Go to stage 1.

Example

Describe a Truing machine for $A = \{0^{2^n} \mid n \geq 0\}$.
Formal description of $M_2 = (Q, \Sigma, \Gamma, \delta, q_1, q_{accept}, q_{reject})$:

- $Q = \{q_1, q_2, q_3, q_4, q_5, q_{accept}, q_{reject}\}$,
- $\Sigma = \{0\}$, and
- $\Gamma = \{0, x, \sqcup\}$.