

# 1 Formal definition of a Turing Machine

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

- $Q$  is the set of states
- $\Sigma$  is the input alphabet *not* containing special blank symbol  $\sqcup$
- $\Gamma$  is the tape alphabet satisfying  $\Sigma \subset \Gamma$  and  $\sqcup \in \Gamma$
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times L, R$  is the transition function
- $q_0 \in Q$  is the start state
- $q_{accept} \in Q$  is the accept state
- $q_{reject} \in Q$  is the reject state,  $q_{reject} \neq q_{accept}$

# 2 Turing Machine computation

**Tape content** is unbounded but always finite

- The first (leftmost) blank symbol marks the end of tape content

A configuration  $C_1$  yields the configuration  $C_2$  if the Turing Machine can legally go from  $C_1$  to  $C_2$  in a single step. A **configuration** consists of:

- The current state
- The tape content
- The head location

The **start configuration** on an input  $w \in \Sigma^*$  consists of start state  $q_0$ ,  $w$  as the tape content, and the head location is the first (leftmost) position of the tape.

A configuration is **accepting** if its state is  $q_{accept}$ . A configuration is **rejecting** if its state is  $q_{reject}$ .

Accepting and rejecting configurations are **halting** configurations.

A Turing Machine  $M$  **accepts** an input  $w$  if there is a sequence of configurations  $C_1, C_2, \dots, C_k$  such that:

1.  $C_1$  is the start configuration of  $M$  on input  $w$
2.  $C_i$  yields  $C_{i+1}$  for  $1 \leq i \leq k-1$
3.  $C_k$  is an accepting configuration

The **language** of  $M$ , denoted  $L(M)$ , is the set of strings accepted by  $M$ .

### 3 Turing-recognisable languages

A language  $L$  is **Turing-recognisable** if there is a Turing Machine  $M$  that recognises it, i.e.  $L$  is the language of  $M$ .

Turing-recognisable means the same thing as **semi-decidable** and **recursively enumerable**.

If  $M$  recognises  $L$ , it may or may not halt on words not in  $L$ .

### 4 Turing-decidable languages

A language  $L$  is **Turing-decidable** if there is a Turing Machine  $M$  that accepts every  $w \in L$  and rejects every  $w \notin L$ .

Turing-decidable means the same thing as **recursive**.

If  $M$  decides  $L$ , it always halts.