



planetmath.org

Math for the people, by the people.

formal definition of a Turing machine

Canonical name	FormalDefinitionOfATuringMachine
Date of creation	2013-03-22 19:09:47
Last modified on	2013-03-22 19:09:47
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	14
Author	CWoo (3771)
Entry type	Definition
Classification	msc 68Q10
Classification	msc 68Q05
Classification	msc 03D10
Related topic	TuringComputable

Based on the informal description of a Turing machine in the parent entry, we give it a formal mathematical definition:

Definition. A Turing machine T is a 7-tuple consists of the following:

1. an alphabet S called the state alphabet,
2. an element $s \in S$ called the start state,
3. an element $t \in S$ called the accept state, and
4. an element $r \in S$ such that $r \neq t$, called the reject state,
5. an alphabet Σ called the input alphabet,
6. a symbol called the blank symbol B not in Σ ,
7. a function $\delta : S \times \Sigma \rightarrow S \times \Sigma \times \{L, R\}$ called the transition function, such that

$$\delta(t, a) = \delta(t, b, X) \quad \text{and} \quad \delta(r, c) = \delta(r, d, Y),$$

where $a, b, c, d \in \Sigma \cup \{B\}$ and $X, Y \in \{L, R\}$.

t, r are collectively called the *halt states* of T .

Actually, the definition above is only part of the story. What is described in the definition is the “finite control” (the brain) portion of a Turing machine. All by itself, the finite control is useless. In order for a Turing machine to do computations, two other ingredients are needed to complete the description: a tape, and a reading head. The two can be formalized as follows:

Definition. A *tape*, or formally a *tape description*, is a function $\tau : \mathbb{Z} \rightarrow \Sigma \cup \{B\}$. A *position of the reading head* is an integer $n \in \mathbb{Z}$.

In other words, the tape of a Turing machine T is infinitely long in both direction, and is divided up into squares. The content of each square is either a symbol in Σ or blank (B). The purpose of the tape is to store input, output, as well as any other strings that are used during computations.

The finite control and the tape of a Turing machine are connected by a reading head, which points over the tape, has the ability to move left or right along the tape, and reads the tape one square at a time. The position of the reading head is the square it is currently reading.

The Turing machine defined this way is also known as a two-way deterministic Turing machine.

To formally describe the computational process of a Turing machine, we need to know three things: what is the current state of the finite control, what is on the tape, and what is the current position of the reading head:

Definition. A *configuration* of a Turing machine is a triple (p, τ, m) , where $p \in S$, called the state of the configuration, τ is a tape description called the tape description of the configuration, and m is a position of the reading head called *the* position of the reading head of the configuration.

We are now ready to describe what it means for a Turing machine to compute something:

Definition. Given a Turing machine T (with the associated tape and reading head), a *computation step* \rightarrow is a binary relation on the set of all configurations of T , defined as follows: if

$$(p, \tau, m) \rightarrow (q, \tau', n),$$

then

$$(q, \tau', n) = \begin{cases} (q, \tau', m+1) & \text{provided that } \delta(p, \tau(m)) = (q, b, R), \\ (q, \tau', m-1) & \text{provided that } \delta(p, \tau(m)) = (q, b, L), \end{cases}$$

where

$$\tau'(i) = \begin{cases} b & \text{if } i = m, \\ \tau(i) & \text{otherwise.} \end{cases}$$

We can of course take the reflexive transitive closure of \rightarrow to obtain \rightarrow^* .

Definition. An input to a Turing machine is a (finite) word w over Σ . If the length of w is n , we define the tape description τ_w of w as follows:

$$\tau_w(i) = \begin{cases} w_i & \text{the } i\text{-th letter of } w, \text{ where } 0 \leq i \leq n, \text{ and} \\ B & \text{otherwise.} \end{cases}$$

Note that τ_λ is the constant function whose value is B , where λ is the empty word.

Definition. Given an input w and a Turing machine T , a computation of w by T is a finite sequence

$$(s, \tau_w, 1) = (p_1, \tau_1, m_1), (p_2, \tau_2, m_2), \dots, (p_k, \tau_k, m_k),$$

where $(p_i, \tau_i, m_i) \rightarrow (p_{i+1}, \tau_{i+1}, m_{i+1})$. From the computation of w above, we also say that the computation of w leads to state p_k . T is said to *accept* an input w iff it there is a computation of w leading to the accepting state t . In

other words, w is accepted by T iff $(s, \tau_w, 1) \rightarrow^* (t, \tau, m)$ for some τ and m . Similarly, w is rejected by T iff $(s, \tau_w, 1) \rightarrow^* (r, \tau, m)$ for some τ and m . T is said to *halt* on input w iff it either accepts or rejects w . Otherwise, T is said to *loop* on w .

By the definition of the transition function δ , once the computation of w leads to the accepting state t , it never leaves that state. Likewise, once the computation of w leads to the reject state r , no further computation will lead to a non-reject state.

Definition. Let T be a Turing machine. The set of words accepted by T is denoted by $L(T)$. A set A of words over Σ is said to be *Turing acceptable* if there is a Turing machine T such that $A = L(T)$. If T halts on every input word, then T is said to be *total*.

Remarks.

- It can be shown that $A \subseteq \Sigma^*$ is Turing acceptable iff it is recursively enumerable. Furthermore, A is recursive iff there it is Turing acceptable by a total Turing machine.
- The Turing machines described above are known as language acceptors. However, any Turing machine can be modified so it is capable of producing outputs. Using this modification, one can think of a Turing machine as a partial function where an input is in its domain iff the computation leads to a halting state. More can be found <http://planetmath.org/TuringComputablehere>.