2 Natural numbers and Turing machines

2.1 $\mathbb{N}$, the natural number system

To make things definite, by 'natural numbers' we mean the natural numbers to base 10, as learnt in school:

$$\mathbb{N} = \{0, 1, 2, \ldots, 9, 10, 11, \ldots, 99, 100, \ldots\}$$

Natural numbers are unsigned decimal integers, without leading zeroes (except for 0 itself).

Alternative 'models' of the natural numbers include binary numbers, strings of 1s, hexadecimal, and so on. But the most 'natural' natural numbers are unsigned decimals.

Notation for partial functions. We write $f : X \rightharpoonup Y$ to mean that f is a partial function from X into Y, i.e., f is a function whose domain is a subset of X, and whose range is a subset of Y.

2.2 Alphabets and strings

By 'alphabet' we could mean the Roman alphabet, or the ten decimal digits, or the two binary digits, or the sixteen hexadecimal digits, or the set of all printable ASCII characters, and so on. Formally, any *finite nonempty set* will serve as an alphabet. Its elements are called *letters* or *symbols.*

If Σ is an alphabet, then a *string over* Σ is a finite sequence $a_1 \ldots a_k$ where $k \geqslant 0$ and for $1 \leqslant j \leqslant k$, $a_j \in \Sigma$. If $k = 0$ then the string is *empty.* The empty string is usually denoted by the small Greek letter λ.[1]

Σ^* is the set of strings over Σ. Without fully explaining the notation, we can make it clear what the decimal number system is; this is our 'natural' model of the natural numbers.

$$\mathbb{N} = \{0\} \cup \{1, \ldots, 9\}\{0, 1, \ldots, 9\}^*$$

Meaning: a natural number (in the decimal system) is either 0 or a nonempty string of decimal digits, beginning with a nonzero digit.

2.3 Idea of a Turing machine

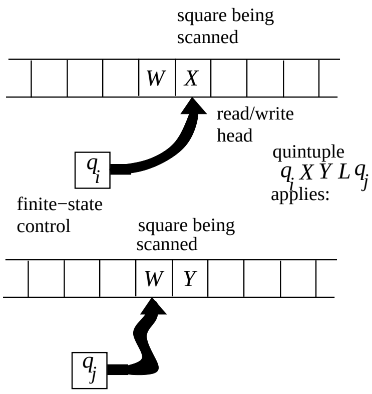
A Turing machine is a kind of computer which applies simple operations working with a potentially infinite memory. The memory is an infinite ('paper') tape divided into squares. Each square contains one symbol from a fixed finite set, the *tape alphabet* Γ. See the illustration below.

At any time all but finitely many of the tape squares are blank. The tape alphabet Γ contains a distinguished symbol B representing the blank.[2] (Only squares near the read/write head have been shown nonblank; possibly other squares are nonblank.)

Generally, B is the blank symbol, though occasionally we have used B instead.

[1] Possibly λ is actually one of the letters in Σ; then we try something else, but it doesn't happen in these notes.

[2] So B means blank, not the letter B.



One thinks of the Turing machine as having memory, in the form of a 'paper' tape, a *finite-state control*, which can 'adopt' one of finitely many 'states,' and a *read/write head* which is positioned at a tape square, and can move to the next square on the left or right.

The following data

- The state, q_i, say;
- The entire tape contents (but only a finite part is nonblank)
- The tape square where the read/write head is positioned

together define a *configuration* of the Turing machine.

The next action taken by the Turing machine depends on two things

- The state q_i of the finite control, and
- The symbol X in the square being scanned.

Its possible actions are presented as a list of *quintuples*

$$q_i \; X \; Y \; \left\{ \begin{array}{c} L \\ R \end{array} \right\} \; q_j$$

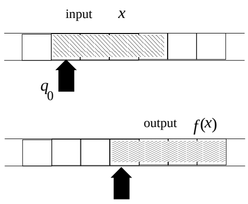
Required: for each state q_i and tape symbol X, there is at most one quintuple $q_i X \ldots$ Implicitly, the quintuples are equivalent to a partial function from $K \times \Gamma$ to $\Gamma \times \{L, R\} \times K$. This partial function is usually denoted δ and called the transition function.

$$\delta : K \times \Gamma \nrightarrow \Gamma \times \{L, R\} \times K.$$

If in state q_i scanning symbol X no such quintuple exists, then the machine does nothing more; it is in a *halting* configuration.

Otherwise, suppose the unique quintuple $q_i XY(R/L)q_j$. Then the machine

- Replaces X by Y in the square being scanned.



- Moves to the next square on left (L) or right (R).
- Adopts the state q_j.

A Turing machine computes a partial function from strings to strings. It must have a distinguished *initial state*, usually denoted q_0. Also, it must have an *input alphabet* Σ. $\Sigma \subseteq \Gamma \backslash \{B\}$. That is, the input alphabet is part of the tape alphabet, and it does not include the blank symbol B.

(2.1) Example. States $K = \{q_0, q_1, q_2\}$, input alphabet $\Sigma = \{1\}$, tape alphabet $\Gamma = \{1, 0, B\}$. Quintuples

$$q_0 1 B R q_1,\ q_1 1 B R q_0,\ q_0 B 0 R q_2,\ q_1 B 1 R q_2,\ q_2 B B L q_3$$

This Turing machine will halt with nonblank tape contents 0 or 1 according as the input string has even or odd length. In other words, on input 1^n, it computes $n \mod 2$. (Note that the alphabet consists of 1 alone.)

(2.2) Definition *A Turing machine M* computes *a partial function $f : \Sigma^* \nrightarrow \Pi^*$ if (a) Π does not include the blank symbol B, (b) whenever M halts, the tape contains a string $y \in \Pi^*$, surrounded by blanks; (c) given an input string x, (i) if $f(x)$ is defined then M halts with $f(x)$ on the tape, and the read/write head positioned at the leftmost symbol of $f(x)$) (blank if $f(x) = \lambda$); (ii) and if $f(x)$ is undefined then the machine loops on input x.*

2.4 Configurations encoded by strings

Let M be a Turing machine, with components $K, \Sigma, B, \Gamma, q_0, \delta$. The quintuples define the partial function δ:

$$\delta : \ K \times \Gamma \nrightarrow \Gamma \times \{L, R\} \times K.$$

The state-set K can be replaced by any other set with the same cardinality; one makes corresponding replacements in the quintuples. So we may assume that K and Γ are disjoint as sets.

A configuration can be represented by the string $\alpha q_i \beta$, where

- The combined string $\alpha\beta$ represents the (nonblank) tape contents.

  The string $\alpha\beta$ must contain all nonblank tape symbols, but it can include blanks as well.

- The square being scanned contains the leftmost symbol of β. Since β can be blank, we rephrase it: the square being scanned contains the leftmost symbol of βB.

  That is, the read/write head is positioned at the leftmost symbol of βB.

- The state is q_i.
- To make the encoding unique, the first letter (if any) of α must be nonblank, and the last letter (if any) of β must be nonblank.
- Turing machines are *simple* (i.e., straightforward in certain ways). It is easy to define the *semantics* of Turing machines. Given a configuration

  $$\alpha q_i X \beta$$

  if no quintuple begins with $q_i X$ then it is a halting configuration. If one does, for example,

  $$q_i X Y R q_j$$

  then applying this quintuple yields the configuration

  $$\alpha Y q_j \beta$$

  (There are a few more cases to look at.)

(2.3) Definition *Suppose that M is a Turing machine with input alphabet Σ. For any $x \in \Sigma^*$, the* initial configuration with input x *is*

$$q_0 x$$

and if σ, τ are configurations where σ is not halting, and using the one applicable quintuple on σ we obtain the (unique) configuration τ, then we write

$$\sigma \ \vdash_M \ \tau$$

and say that σ yields τ immediately*.*

The reflexive transitive closure of this relation is denoted $\vdash_M^*$.

(2.4) Lemma *Given a Turing machine M with an input string x, there exists a unique finite or infinite sequence of configurations*

$$\sigma_0 = q_0 x, \ \sigma_1, \ \sigma_2, \ldots$$

where if σ_i is not halting then $\sigma_i \ \vdash_M \ \sigma_{i+1}$, and if it is halting then it is the last in the sequence, and the entire sequence depends uniquely on x. (Intuitive; no proof.) ∎

(2.5) Definition *If the above sequence is finite then we say that M* halts on input x*, referring to the last configuration as the* halting configuration*: otherwise it* loops on input x*.*

Let us illustrate this with the Turing machine of §2.1, input string 1111.

$$q_0 1111 \ \vdash_M \quad q_1 111 \ \vdash_M \quad q_0 11 \ \vdash_M \quad q_1 1 \ \vdash_M \quad q_0 \ \vdash_M \quad 0 q_2$$