

und.1 Enumerating Turing Machines

tur:und:enu:
sec

We can show that the set of all Turing-machines is **enumerable**. This follows from the fact that each Turing machine can be finitely described. The set of states and the tape vocabulary are finite sets. The transition function is a partial function from $Q \times \Sigma$ to $Q \times \Sigma \times \{L, R, N\}$, and so likewise can be specified by listing its values for the finitely many argument pairs for which it is defined. Of course, strictly speaking, the states and vocabulary can be anything; but the *behavior* of the Turing machine is independent of which objects serve as states and vocabulary. So we may assume, for instance, that the states and vocabulary symbols are natural numbers, or that the states and vocabulary are all strings of letters and digits.

explanation

Suppose we fix a **denumerable** vocabulary for specifying Turing machines: $\sigma_0 = \triangleright, \sigma_1 = 0, \sigma_2 = 1, \sigma_3, \dots, R, L, N, q_0, q_1, \dots$. Then any Turing machine can be specified by some finite string of symbols from this alphabet (though not every finite string of symbols specifies a Turing machine). For instance, suppose we have a Turing machine $M = \langle Q, \Sigma, q, \delta \rangle$ where

$$Q = \{q'_0, \dots, q'_n\} \subseteq \{q_0, q_1, \dots\} \text{ and} \\ \Sigma = \{\triangleright, \sigma'_1, \sigma'_2, \dots, \sigma'_m\} \subseteq \{\sigma_0, \sigma_1, \dots\}.$$

We could specify it by the string

$$q'_0 q'_1 \dots q'_n \triangleright \sigma'_1 \dots \sigma'_m \triangleright q \triangleright S(\sigma'_0, q'_0) \triangleright \dots \triangleright S(\sigma'_m, q'_n)$$

where $S(\sigma'_i, q'_j)$ is the string $\sigma'_i q'_j \delta(\sigma'_i, q'_j)$ if $\delta(\sigma'_i, q'_j)$ is defined, and $\sigma'_i q'_j$ otherwise.

Theorem und.1. *There are functions from \mathbb{N} to \mathbb{N} which are not Turing computable.*

Proof. We know that the set of finite strings of symbols from a **denumerable** alphabet is **enumerable**. This gives us that the set of descriptions of Turing machines, as a subset of the finite strings from the **enumerable** vocabulary $\{q_0, q_1, \dots, \triangleright, \sigma_1, \sigma_2, \dots\}$, is itself enumerable. Since every Turing computable function is computed by some (in fact, many) Turing machines, this means that the set of all Turing computable functions from \mathbb{N} to \mathbb{N} is also enumerable.

On the other hand, the set of all functions from \mathbb{N} to \mathbb{N} is not **enumerable**. This follows immediately from the fact that not even the set of all functions of one argument from \mathbb{N} to the set $\{0, 1\}$ is **enumerable**. If all functions were computable by some Turing machine we could enumerate the set of all functions. So there are some functions that are not Turing-computable. \square

Photo Credits

Bibliography