A function $f:N^k \rightarrow N$ is Turing-computable^{*} if and only if there exists a Turing machine M such that for all $\langle n_1, n_2, ..., n_k \rangle$: if $f(n_1, n_2, ..., n_k) = n$, then M, when started (in q_1) on $[n_1, n_2, ..., n_k]$, will halt (in q_0) with [n].

Relative to an enumeration $E = M_1, M_2, ...$ of Turing-machines, the self-halting function $h_E: N \rightarrow N$ is the function defined as $h_E(n) = 1$ if M_n , when started on [n], halts $h_E(n) = 0$ if M_n , when started on [n], does not halt

Relative to an enumeration $E = M_1, M_2, ...$ of Turing-machines, the halting function $h_E:N \times N \to N$ is the function defined as $h_E(m,n) = 1$ if M_m , when started on [n], halts $h_E(m,n) = 0$ if M_m , when started on [n], does not halt

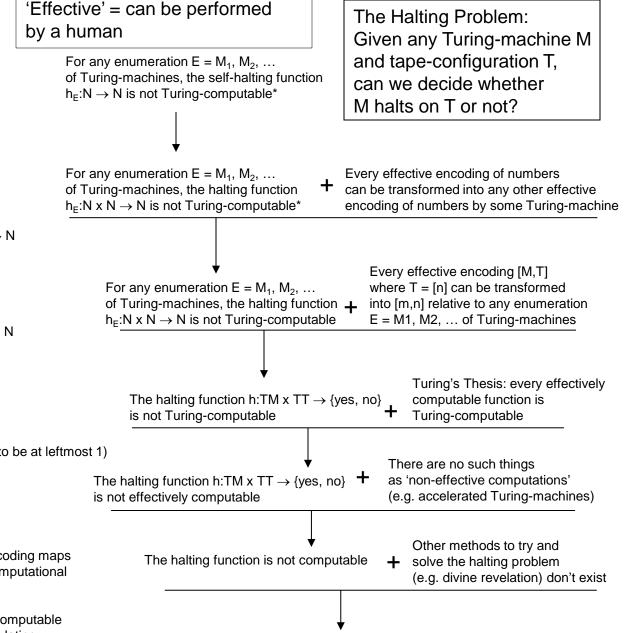
TM is the set of all Turing-machines

TT is the set of all tape-configurations (specifying tape content; head position is assumed to be at leftmost 1)

The halting function h:TM x TT \rightarrow {yes, no} is the function defined as h(M,T) = yes if M, when started on T, halts h(M,T) = no if M, when started on T, does not halt

Given some computational method, an effective encoding maps any $x \in X$ to a representation [x] that that kind of computational method can work on.

The halting function h:TM x TT \rightarrow {yes, no} is MM-computable if and only if there exists an effective symbol-manipulating algorithm (i.e. computation, program, or machine) H of kind MM, such that for any M and T: H starts on [M,T], and halts on [yes] or [no], depending on whether M with T halts or does not halt respectively.



The Halting Problem is not solvable