computable vs. uncomputable functions

-> not every function $f: \{0,1\}^* \to \{0,1\}^*$ is computable

-> every bit string can be mapped to a natural number $\mathbb{N}$

-> hence, set of all functions:

$f: \mathbb{N} \to \mathbb{N}$

-> however, the set of all functions on natural numbers is uncountable:

↳ the reason is that interval $(0,1)$ corresponds to all functions

$g: \mathbb{N} \to \{0, \dots, 9\}$, $g(i) =$ „ i-th digit of the number"

-> e.g.: $0.78566\dots$, $g(1) = 7, g(2) = 8, g(3) = 5, g(4) = 6, g(5) = 6, \dots$

↳ since $(0,1)$ is uncountable, we have that there are uncountable many functions f on natural numbers

-> on the other hand, there are only a countable number of algorithms/computable functions:

↳ an algorithm has finite description, thus representable by a finite number of bits -> representable as a natural number.

-> thus number of algorithms/computable functions << number of all problems