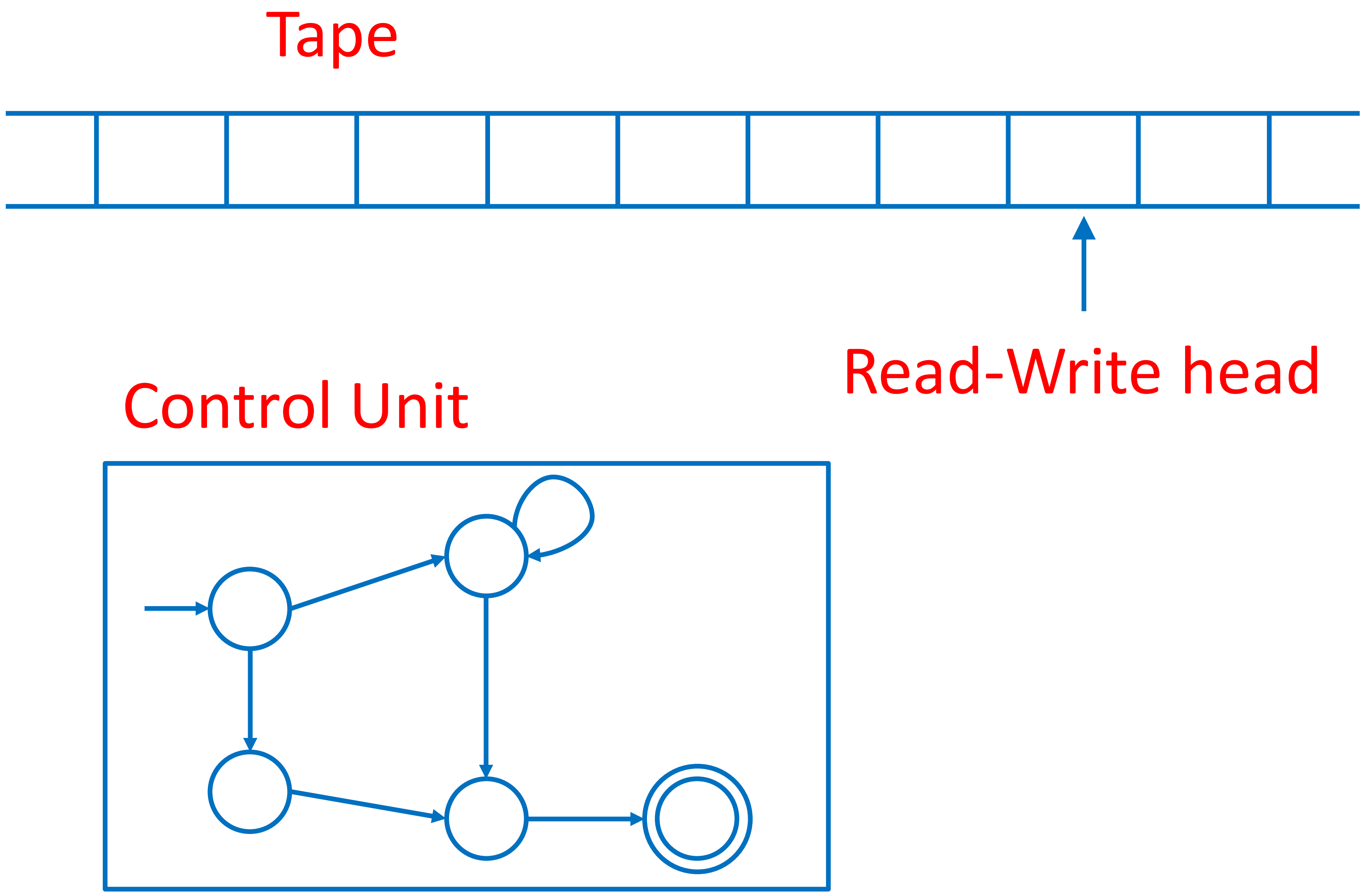
Turing Machines As Transducers

A Standard Turing Machine



A Turing Machine as a Transducer

Transducer:

- The **input** for a computation will be all the nonblank symbols on the tape at the initial time.
- At the conclusion of the computation, the **output** will be whatever is on the tape.

$$q_0 w \vdash^* q_f f(w)$$

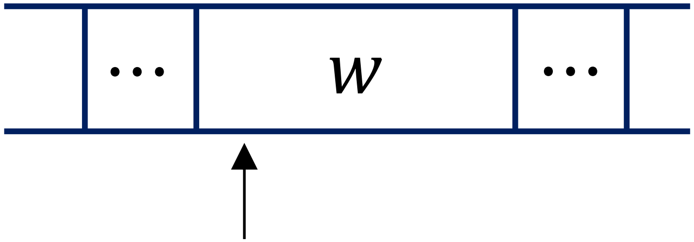
A Computable Functions

Definition: A function f with domain D is said to be **Turing-computable** or just **computable** if there exists some Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ such that

$$q_0 w \vdash_M^* q_f f(w), \qquad q_f \in F$$

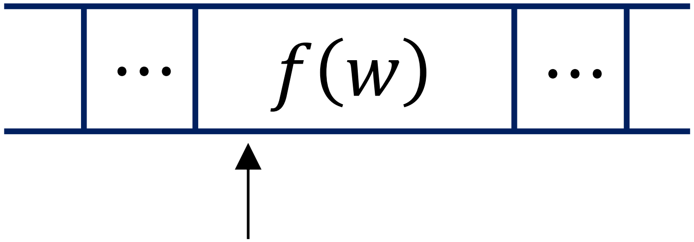
for all $w \in D$.

Initial configuration



q_0 initial state

Final configuration



q_f final state

A Computable Functions

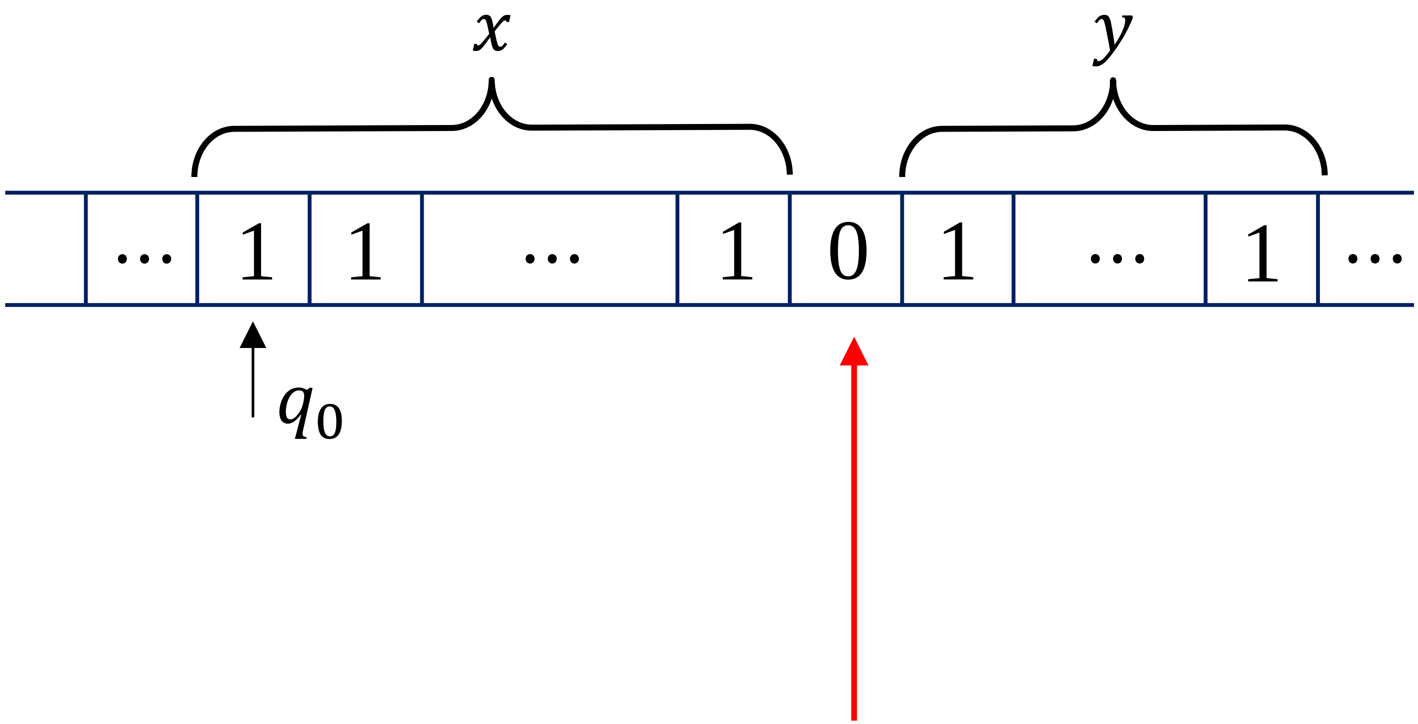
Example: Given two positive integers x and y, design a Turing machine that computes $x + y$.

Solution:

- x and y are represented by $w(x) \in \{1\}^+$ and $w(y) \in \{1\}^+$ such that $|w(x)| = x$ and $|w(y)| = y$.

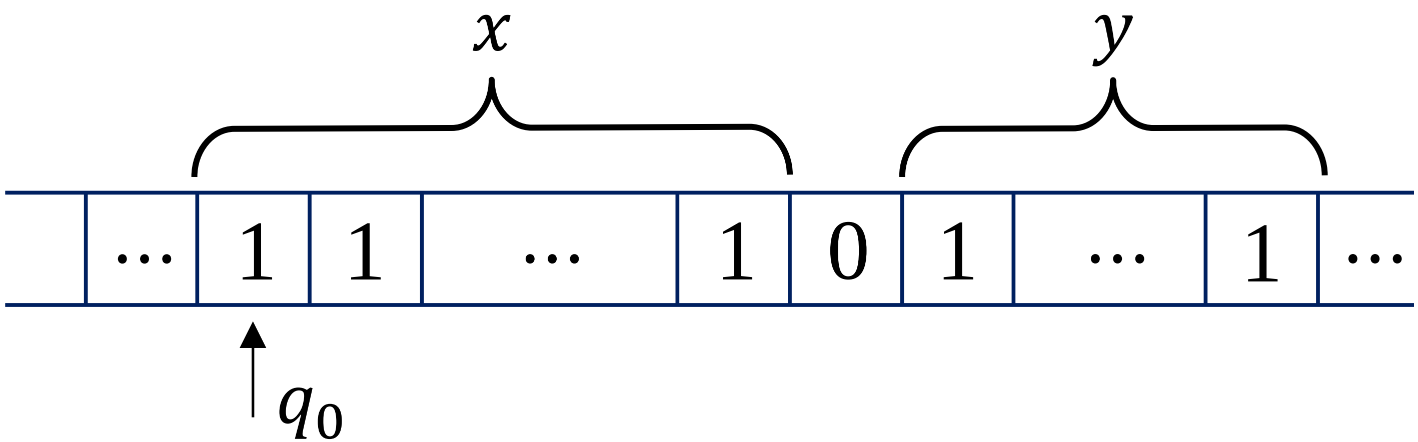
- $q_0 w(x) 0 w(y) \vdash^* q_f w(x + y) 0$

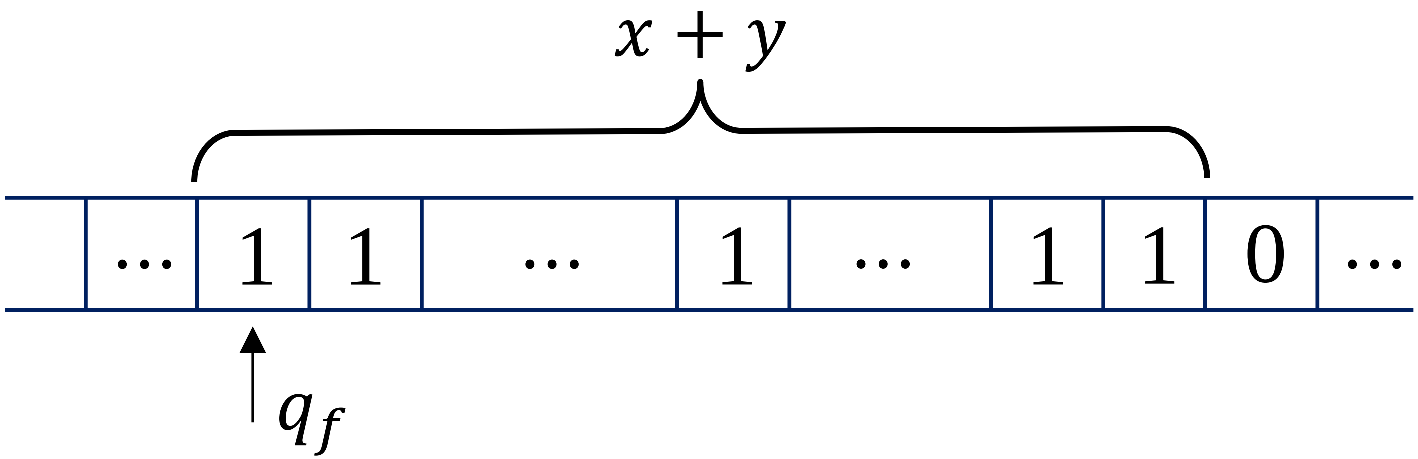
A Computable Functions



The 0 is the delimiter that separates the two numbers

A Computable Functions





A Computable Functions

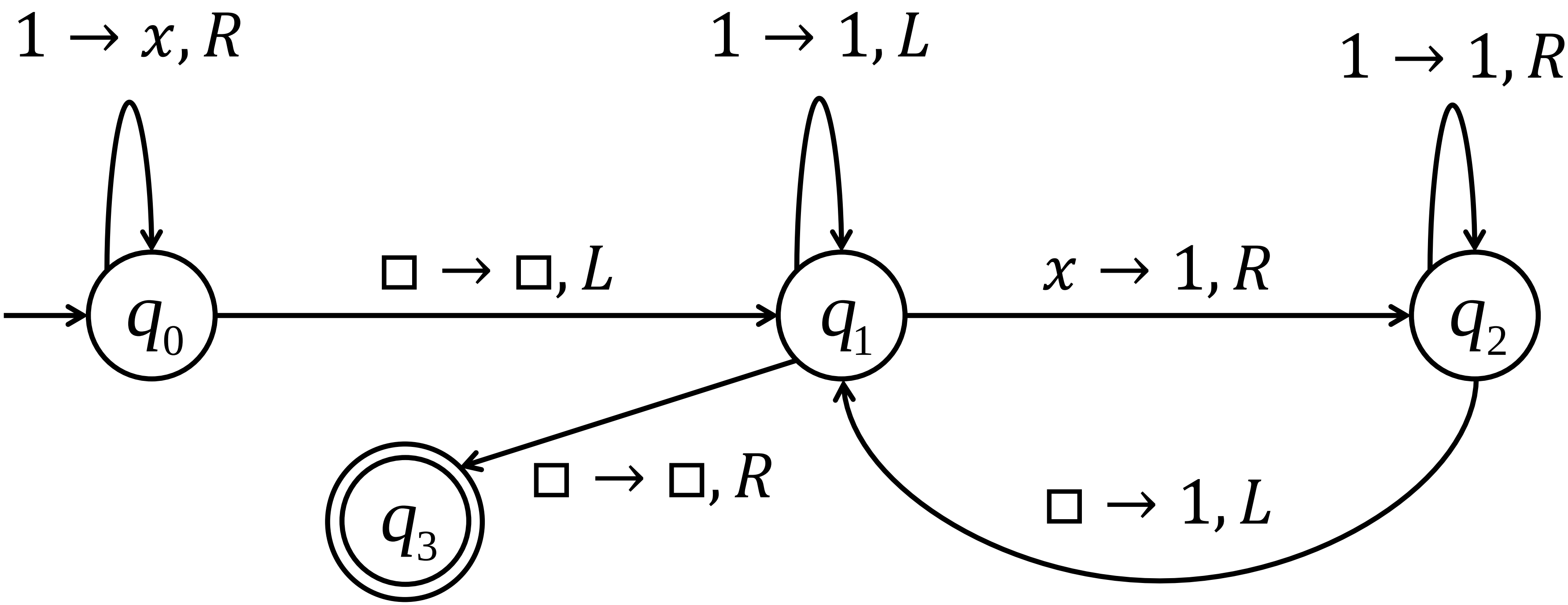
Example: Given a positive integer a, design a Turing machine that computes $2a$ ($w(a) \in \{1\}^+$ with $|w(a)| = a$).

Solution: To solve the problem, we implement the following process:

1. Replace every 1 by an x.
2. Find the rightmost x and replace it with 1.
3. Travel to the right end of the current nonblank region and create a 1 there.
4. Repeat Steps 2 and 3 until there are no more x's

A Computable Functions

Example: Given a positive integer a, design a Turing machine that computes $2a$ ($w(a) \in \{1\}^+$ with $|w(a)| = a$).



A Computable Functions

Another Example:

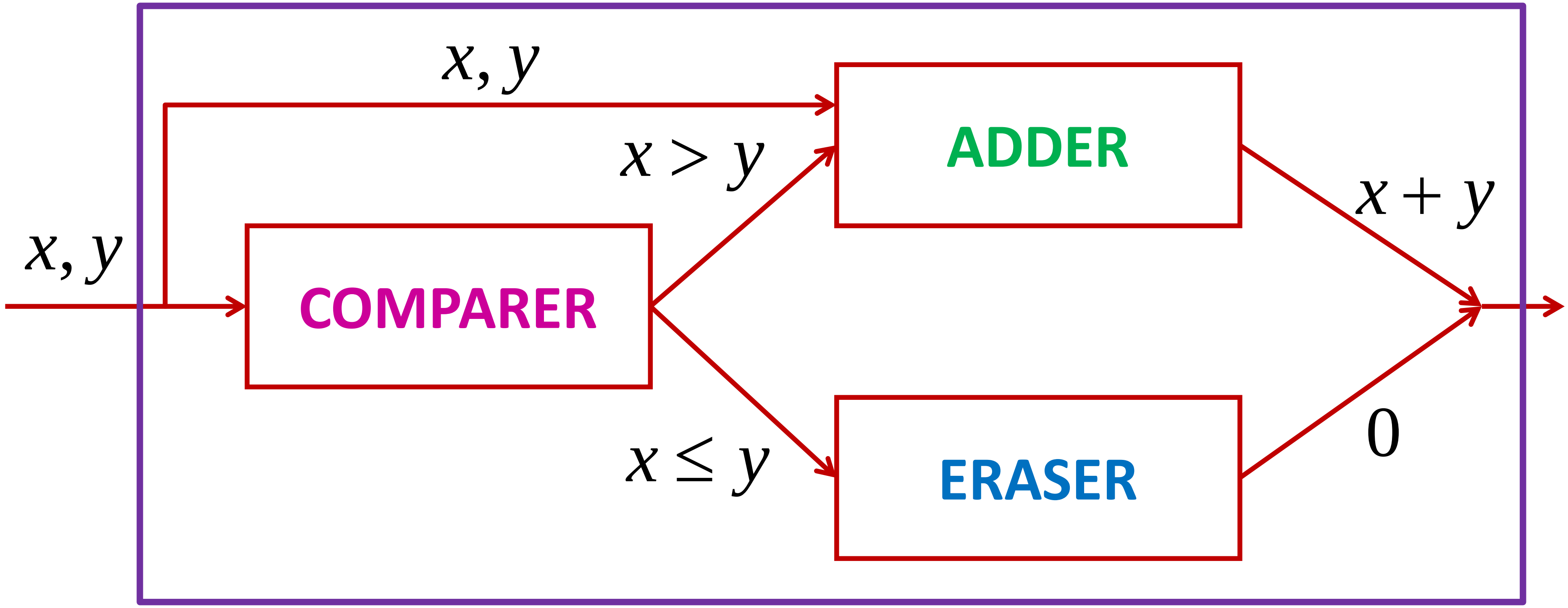
Given positive integers x and y, design a Turing machine that will halt in a final state q_y if $x \geq y$, and that will halt in a non-final state q_n if $x < y$.

Combining Turing Machines

Example:

Design a Turing machine that computes the function

$$f(x) = \begin{cases} x + y, & \text{if } x > y \\ 0, & \text{if } x \leq y \end{cases}.$$



Turing’s Thesis

Turing Thesis

Question:

Do Turing machines have the same power with digital computers?

Intuitive answer: YES

There is no formal answer!!!

Turing Thesis

Turing's thesis (1930):

> **Any computation** carried out by **mechanical / digital means** can be performed by a **Turing Machine**

Turing Thesis

Computer Science Law:

A computation is **mechanical** if and only if it can be performed by a Turing Machine

There is no known model of computation more powerful than Turing Machines

Turing Thesis

Definition of Algorithm:

An algorithm for function $f(x)$ is a Turing machine which computes $f(x)$.

Turing Thesis

Algorithms are Turing Machines

When we say:

There exists an algorithm

We mean:

There exists a Turing Machine that executes the algorithm