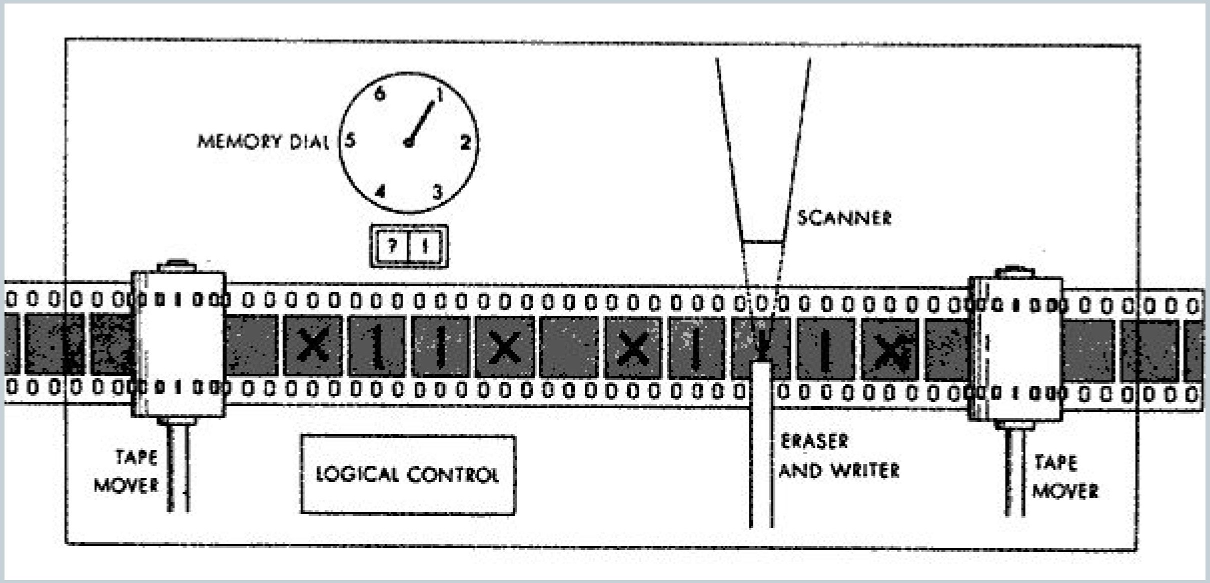
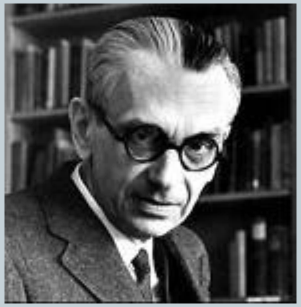
Turing's Influence on Computational Complexity

LANCE FORTNOW
NORTHWESTERN UNIVERSITY



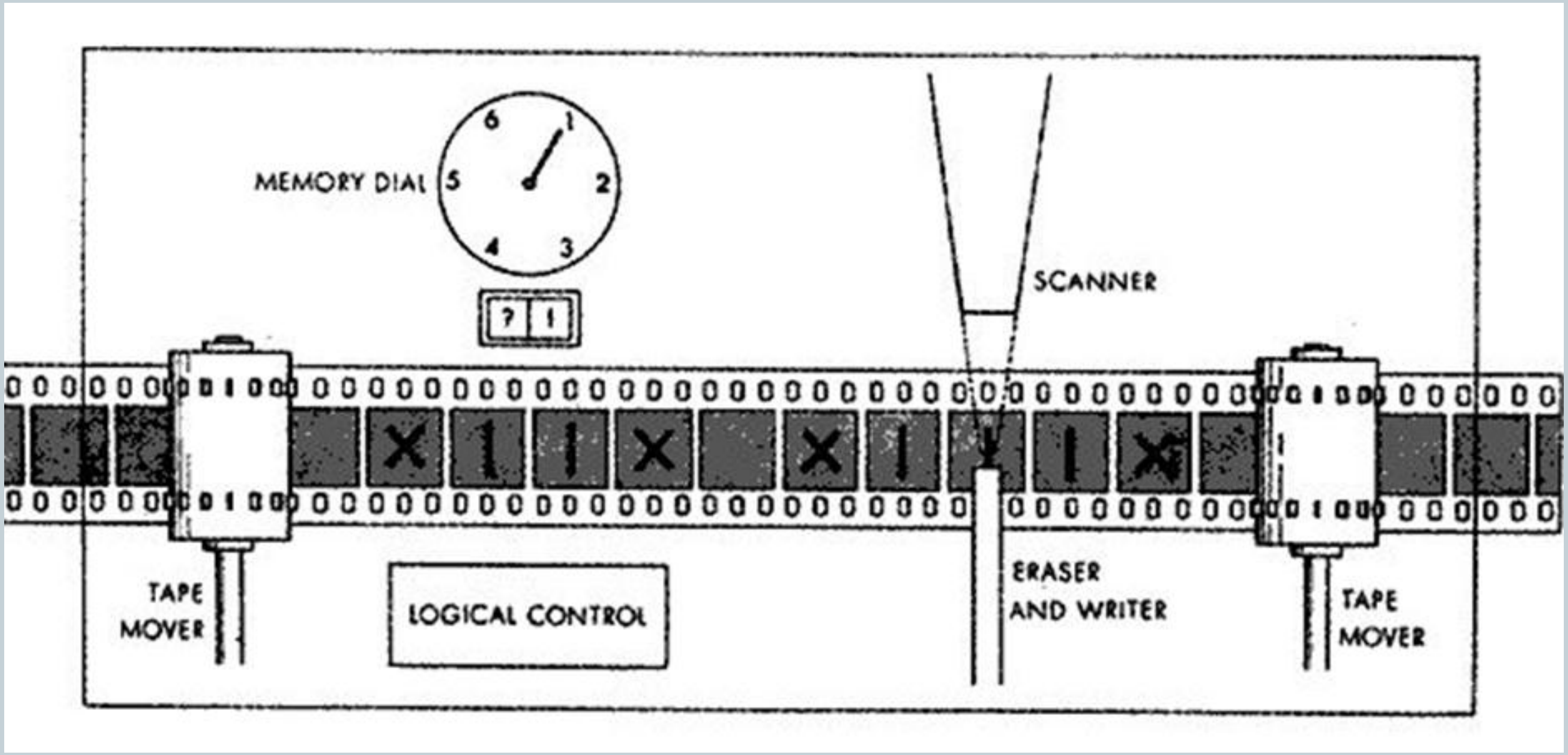
Lost Letters



- Written: 1956
- Discovered: 1988

- One can obviously easily construct a Turing machine, which for every formula F in first order predicate logic and every natural number n, allows one to decide if there is a proof of F of length n (length = number of symbols). Let $\psi(F,n)$ be the number of steps the machine requires for this and let $\varphi(n) = \max_F \psi(F,n)$. The question is how fast $\varphi(n)$ grows for an optimal machine.

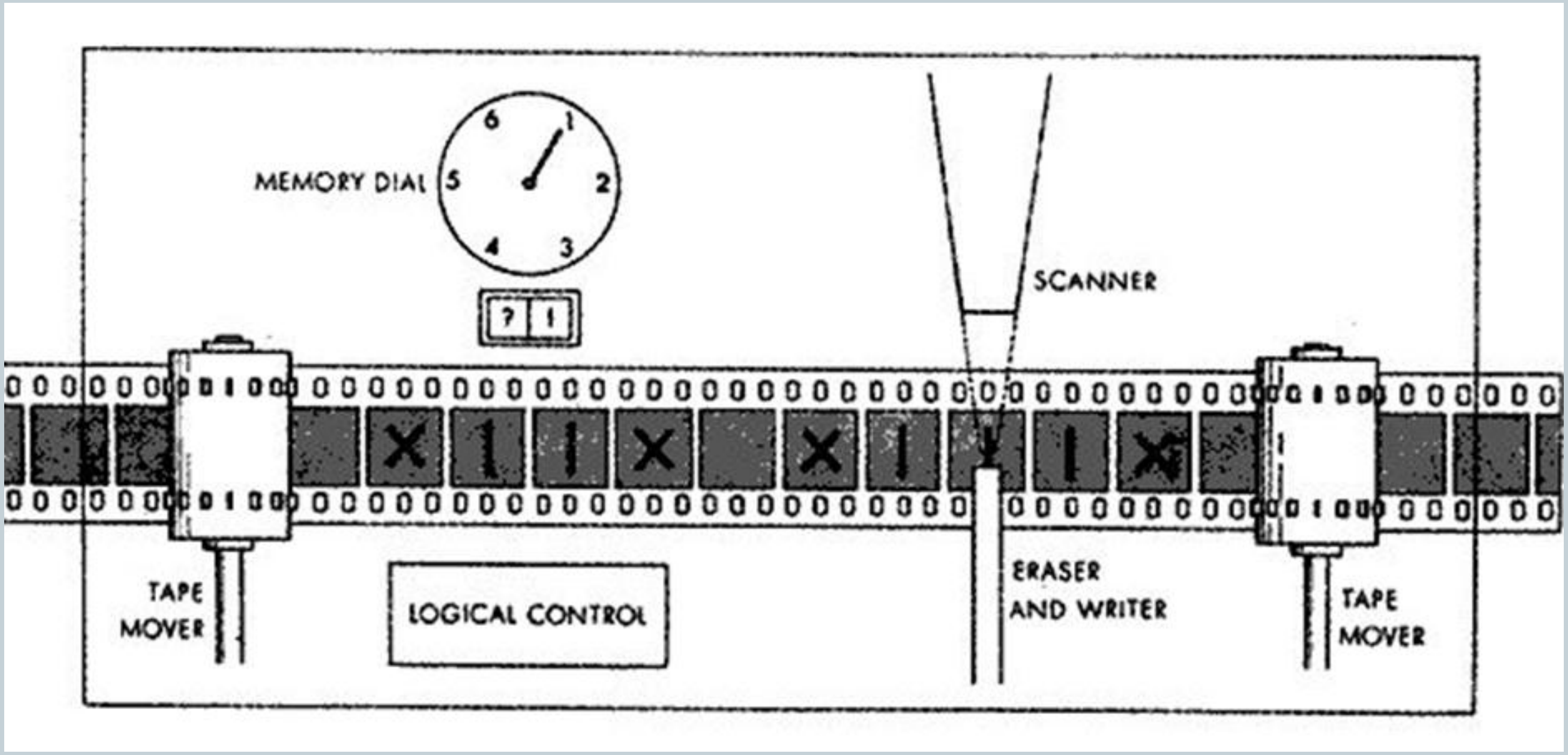
The Turing Machine



Computability

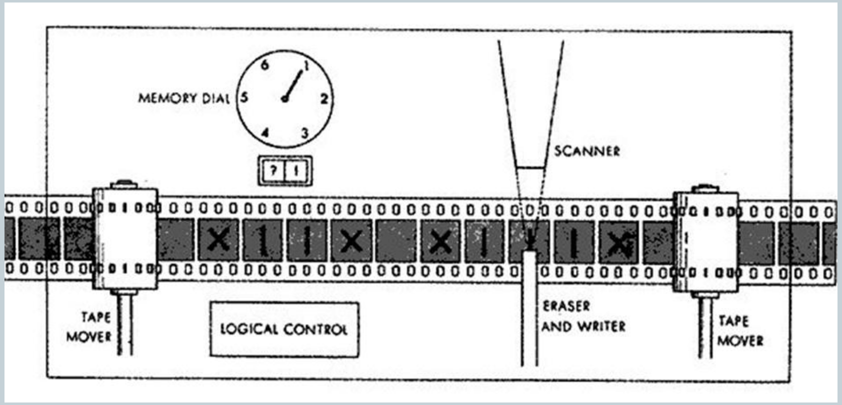
- Functions closed under
  - Constant zero function
  - Successor function
  - Projection
  - Composition
  - Primitive Recursion
  - μ-operator

The Turing Machine



The Turing Machine

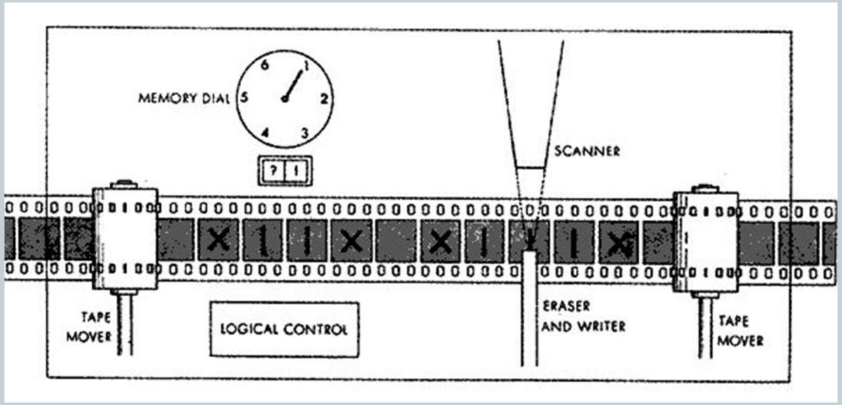
- Σ – Input Alphabet
- Γ – Tape Alphabet
- B –Blank Symbol
- Q – States
- q_o – Start State
- q_{acc} – Accept State
- q_{rej} – Reject State
- δ - Transition Function
  - $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$



The Turing Machine

- Σ – Input Alphabet
- Γ – Tape Alphabet
- B –Blank Symbol
- Q – States
- q_o – Start State
- q_{acc} – Accept State
- q_{rej} – Reject State
- δ - Transition Function
  - $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$

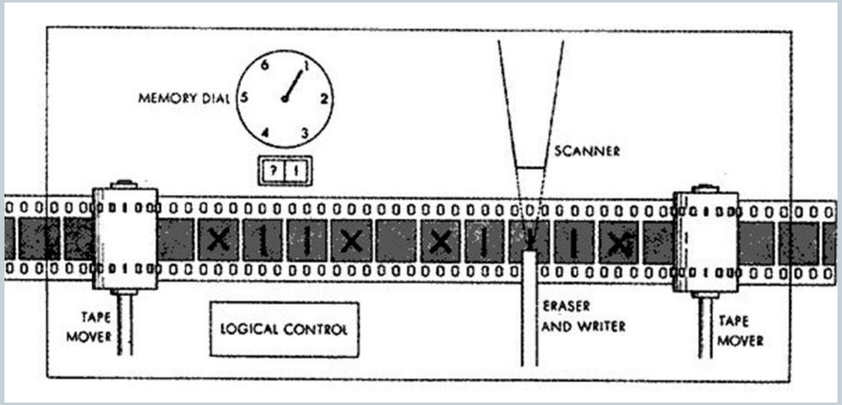
- Data is a finite sequence or string of elements from a finite alphabet



The Turing Machine

- Σ – Input Alphabet
- Γ – Tape Alphabet
- B –Blank Symbol
- Q – States
- q_o – Start State
- q_{acc} – Accept State
- q_{rej} – Reject State
- δ - Transition Function
  - $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$

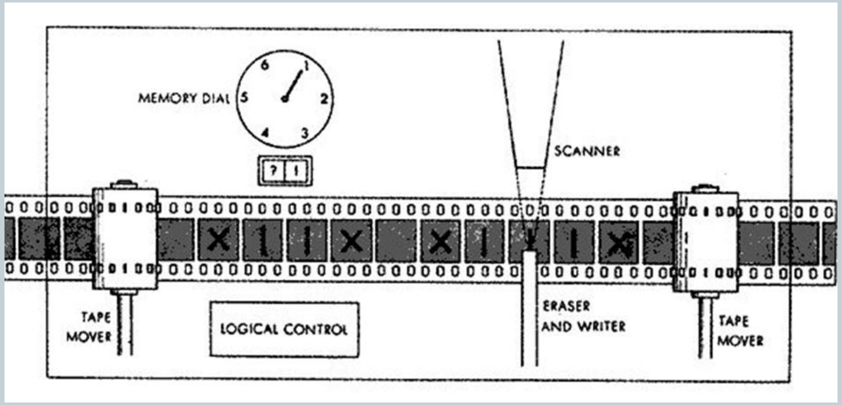
- Data is a finite sequence or string of elements from a finite alphabet
- Size of the data is the length of the string



The Turing Machine

- Σ – Input Alphabet
- Γ – Tape Alphabet
- B –Blank Symbol
- Q – States
- q_o – Start State
- q_{acc} – Accept State
- q_{rej} – Reject State
- δ - Transition Function
  - $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$

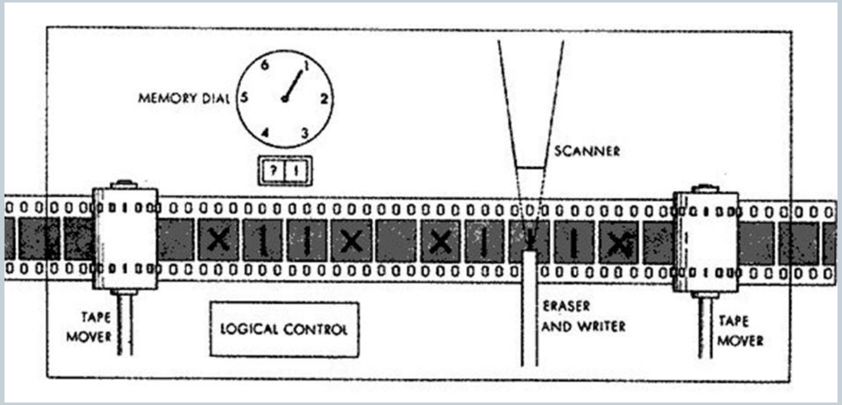
- Data is a finite sequence or string of elements from a finite alphabet
- Size of the data is the length of the string
- Input Size



The Turing Machine

- Σ – Input Alphabet
- Γ – Tape Alphabet
- B –Blank Symbol
- Q – States
- q_o – Start State
- q_{acc} – Accept State
- q_{rej} – Reject State
- δ - Transition Function
  - $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$

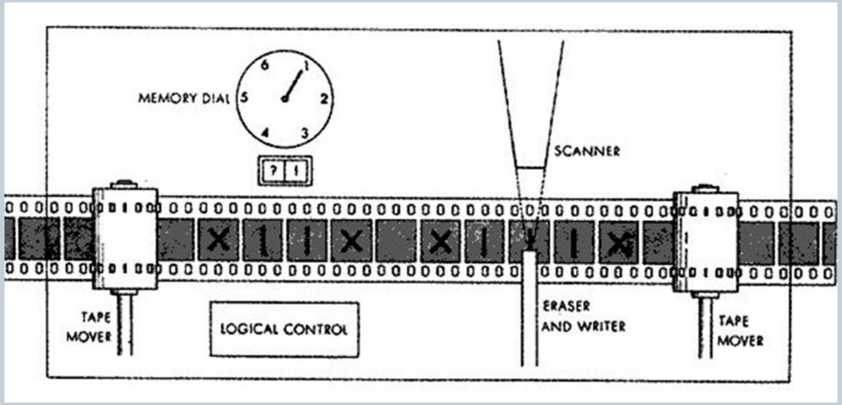
- The finite computer program is fully described by δ



The Turing Machine

- Σ – Input Alphabet
- Γ – Tape Alphabet
- B –Blank Symbol
- Q – States
- q_o – Start State
- q_{acc} – Accept State
- q_{rej} – Reject State
- δ - Transition Function
  - $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$

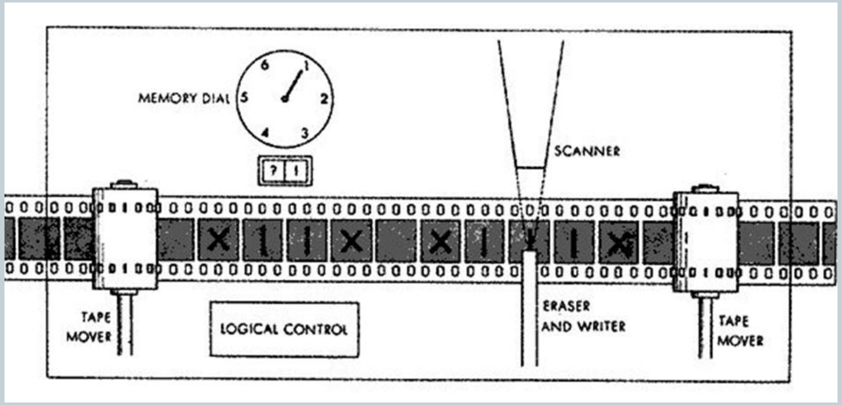
- The finite computer program is fully described by δ
- Easy to convert δ to a string



The Turing Machine

- Σ – Input Alphabet
- Γ – Tape Alphabet
- B –Blank Symbol
- Q – States
- q_o – Start State
- q_{acc} – Accept State
- q_{rej} – Reject State
- δ - Transition Function
  - $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$

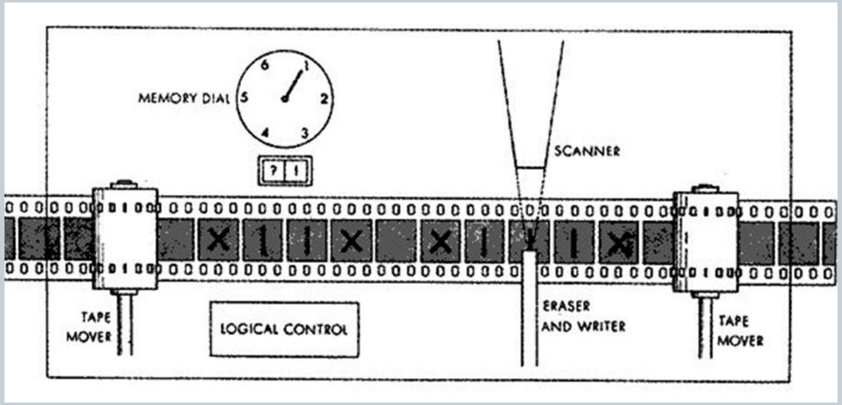
- Simple method to measure computation time
  - Number of applications of the δ function



The Turing Machine

- Σ – Input Alphabet
- Γ – Tape Alphabet
- B –Blank Symbol
- Q – States
- q_o – Start State
- q_{acc} – Accept State
- q_{rej} – Reject State
- δ - Transition Function
  - $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$

- Simple method to measure computation space or memory
  - Number of different tape cells used during computation



Turing Machine is Extensible

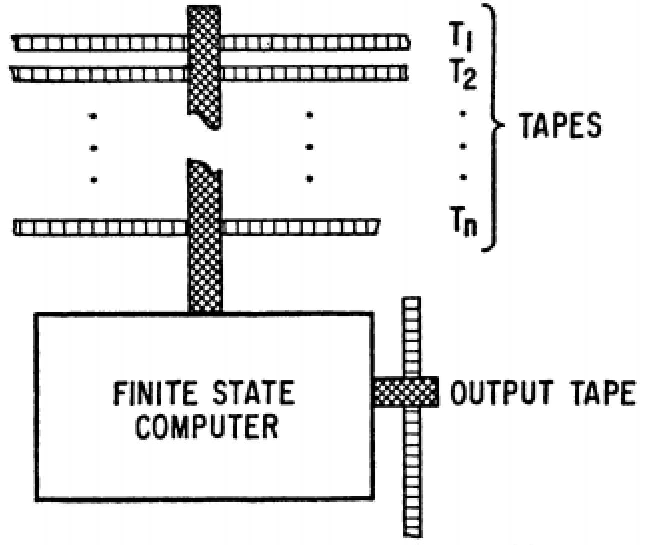


FIGURE 1. An *n*-tape Turing machine

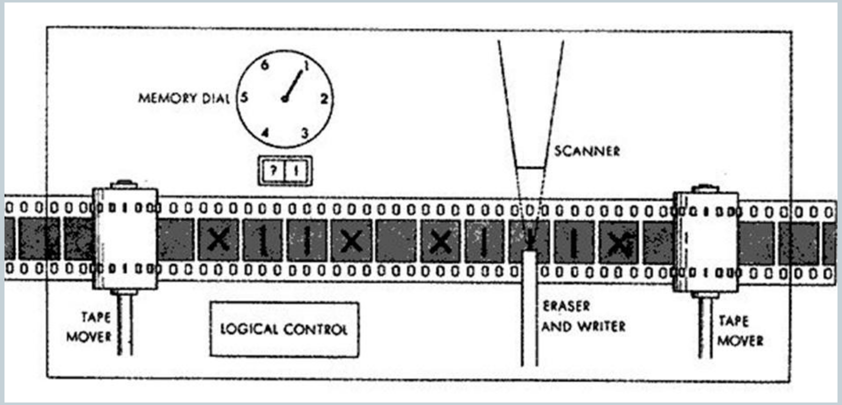
Multitape Turing Machine

Computational Complexity

- "On the Computational Complexity of Algorithms"
  - Hartmanis-Stearns Trans. AMS 1965 (developed in 1962)
    - The computational complexity of a sequence is measured by how fast a multitape Turing machine can print out the terms of the sequence. This particular abstract model of a computing device is chosen because much of the work in this area is stimulated by the rapidly growing importance of computation through the use of digital computers, and all digital computers in a slightly idealized form belong to the class of multitape Turing machines.

Deterministic Turing Machine

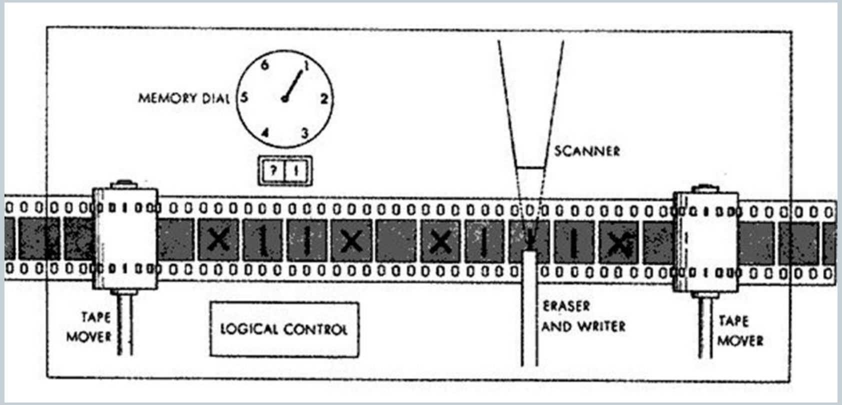
- Σ – Input Alphabet
- Γ – Tape Alphabet
- B –Blank Symbol
- Q – States
- q_o – Start State
- q_{acc} – Accept State
- q_{rej} – Reject State
- δ - Transition Function
  - $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$



Nondeterministic Turing Machine

- Σ – Input Alphabet
- Γ – Tape Alphabet
- B –Blank Symbol
- Q – States
- q_o – Start State
- q_{acc} – Accept State
- q_{rej} – Reject State
- δ - Transition Function
  - $\delta: Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L,R\}}$

- A nondeterministic machine accepts if there is a choice of applications of the δ function that leads to an accept state.



P versus NP

- **P**: Set of problems computable by a deterministic Turing machine in time polynomial in the size of the input.
- **NP**: Set of problems computable by a nondeterministic Turing machine in time polynomial in the size of the input.
- Does **P = NP?**
  - Exact model of Turing machine is irrelevant

Other Variations of Turing Machines

Other Variations of Turing Machines

- Random Access Machine
  - Tapes are pointers into another infinite tape

Other Variations of Turing Machines

- Random Access Machine
  - Tapes are pointers into another infinite tape
- Parallel Computation
  - Several Turing machines sharing a random-access tape

Other Variations of Turing Machines

- Random Access Machine
  - Tapes are pointers into another infinite tape
- Parallel Computation
  - Several Turing machines sharing a random-access tape
- Probabilistic Computation
  - Special coin state which goes to heads state or tails state
  - Tape full of random bits

Other Variations of Turing Machines

- Random Access Machine
  - Tapes are pointers into another infinite tape
- Parallel Computation
  - Several Turing machines sharing a random-access tape
- Probabilistic Computation
  - Special coin state which goes to heads state or tails state
  - Tape full of random bits
- Relativized Computation
  - Oracle tape to write question and oracle state to ask

Other Variations of Turing Machines

- Random Access Machine
  - Tapes are pointers into another infinite tape
- Parallel Computation
  - Several Turing machines sharing a random-access tape
- Probabilistic Computation
  - Special coin state which goes to heads state or tails state
  - Tape full of random bits
- Relativized Computation
  - Oracle tape to write question and oracle state to ask
- Quantum Computation

Why is the Turing machine good for Complexity?

- "On Computable Numbers" – Turing 1936
  - Let us imagine the operations performed by a computer to be split up into "simple operations" which are so elementary that it is not easy to imagine them further divided. Every such operation consists of some change of the physical system consisting of the computer and his tape. We know the state of the system if we know the sequence of symbols on the tape, which of these are observed by the computer and the state of mind of the computer.