Lecture 39

Turing Machines

Turing Machines: How do they look like?

Turing machines are similar to a DFAs but with an unrestricted memory.

Control $\boxed{q_{start}}$

Tape head, through which TM has access to one cell at a time

Tape $\cdots \; \boxed{\sqcup} \boxed{0} \boxed{0} \boxed{1} \boxed{0} \boxed{1} \boxed{1} \boxed{\sqcup} \boxed{\sqcup} \; \cdots$

Tape Symbols $\{0, 1, \sqcup\}$

States $\{q_{start}, q_1, \ldots, q_k, q_{accept}, q_{reject}\}$

Transition Fun. $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

Turing Machines: History

Hilbert's Entscheidungsproblem (1928):

Is there an "effective procedure" that given a set of axioms, some rules, and a mathematical proposition decides whether the mathematical proposition is provable from the axioms using the given rules?

What's an effective procedure?

Several candidates came in 30s and later, namely, Church's λ-calculus, Turing's $\mathscr{A}$ machine, Emil Post's Post System, etc.

Negative answer to Entscheidungsproblem:

Turing presented an unsolvable problem called **Halting Problem** on $\mathscr{A}$ machine and reduced Halting Problem to Entscheidungsproblem. Thus proving Entscheidungsproblem unsolvable as well on $\mathscr{A}$ machines.

Turing Machines: Significance

Why Turing machines are important?

- **Church-Turing Thesis:** Every physically realizable computational device can be simulated by a Turing machine.

  That is, if a problem can be solved on any other physically realizable computational model, then it can be solved on Turing machines as well.

- Due to its simple mathematical nature TMs are used to prove impossibility results such as
  - Proving some problems are unsolvable for computers.
  - Proving existence of problems that can be solved in $O(n^2)$ time, but not in $O(n)$ time.

Turing Machines: Formal Definition

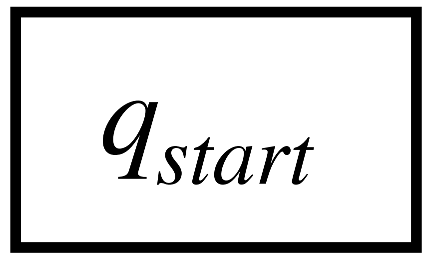
Definition: Turing machine is a 4-tuple, $(Q, \Sigma, \Gamma, \delta)$, where

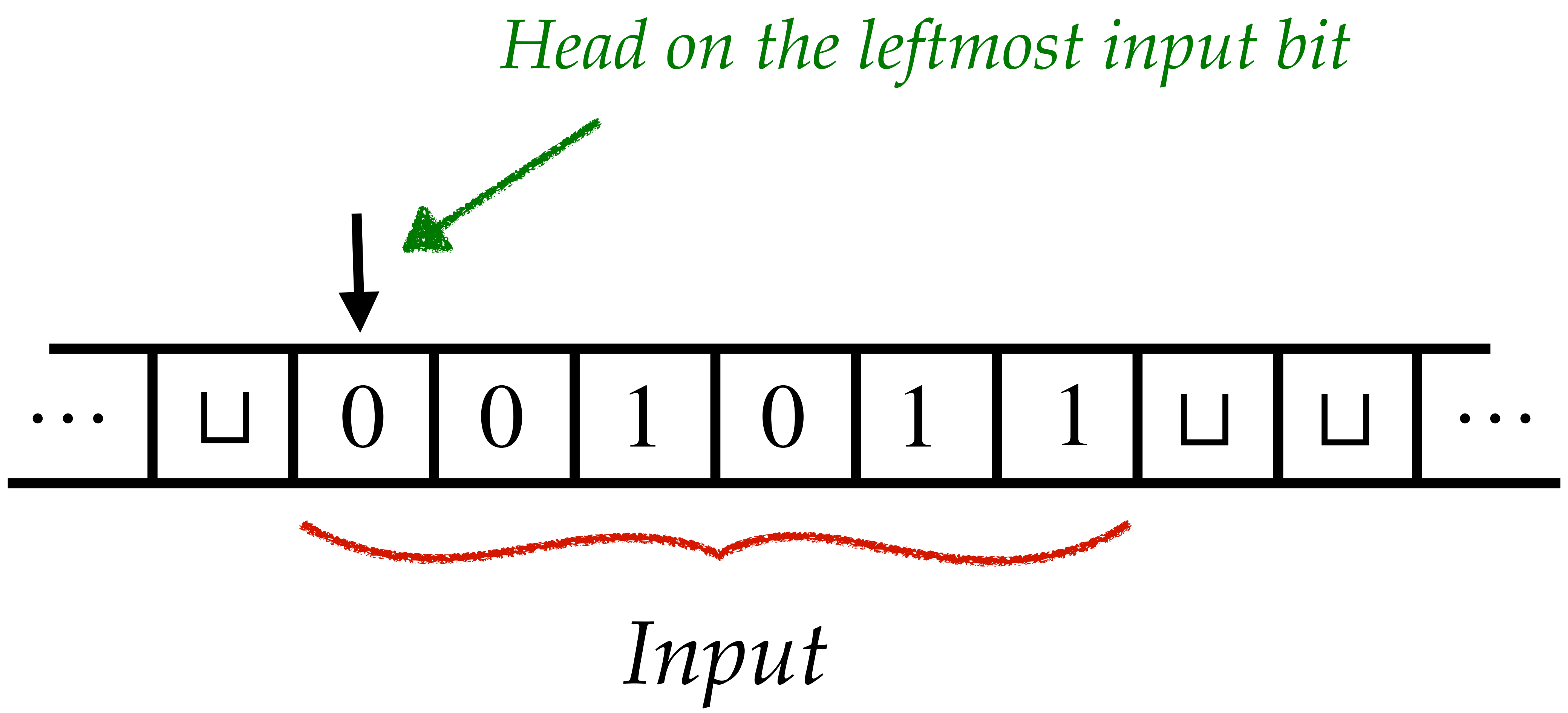
- Q is the finite set of states that contains special states q_{start}, q_{accept}, and q_{reject}.
- Σ is a finite set of input symbols. Usually, $\Sigma = \{0,1\}$.
- Γ is a finite set of tape symbols such that blank symbol $\sqcup \in \Gamma$, and $\Sigma \subseteq \Gamma$.
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is the transition function.

Note: Turing machine halts when it reaches q_{accept} or q_{reject}. Hence, no transitions are defined for q_{accept} and q_{reject}.

Turing Machines: Working

The Initial Setup:





Turing Machine in One Step:

- Gets the state from control and reads the symbol under head.
- If the state is q_{accept} or q_{reject}, it halts. Otherwise it uses transition function to:
  - Change (or not) the symbol under the head.
  - Change (or not) the state in the control.
  - Move the tape head to left or right.

Decidable and Recognisable Languages

Note: A Turing machine on an input can have three outcomes: **accept**, **reject**, or **loop**.

Definition: A language L over Σ is called **Turing decidable** or **decidable** if there exists a TM M such that for every $x \in \Sigma*$

$$x \in L \implies M \text{ on input } x \text{ halts with } q_{accept}$$

$$x \notin L \implies M \text{ on input } x \text{ halts with } q_{reject}$$

Definition: A language L over Σ is called **Turing recognisable** or **recognisable** if there exists a TM M such that for every $x \in \Sigma*$

$$x \in L \implies M \text{ on input } x \text{ halts with } q_{accept}$$

$$x \notin L \implies M \text{ on input } x \text{ halts with } q_{reject} \text{ or it enters a loop}$$