INT201 Decision, Computation and Language

Lecture 9 – Turing Machine

Dr Yushi Li

Xi’an Jiaotong-Liverpool University
西交利物浦大学

DFA, NFA and PDA

DFA

- M = $(Q, \Sigma, \delta, q, F)$
- $\delta : Q \times \Sigma \rightarrow Q$

Finite control (δ) based on

- State
- Input symbol

NFA

- M = $(Q, \Sigma, \delta, q, F)$
- $\delta : Q \times \Sigma_\epsilon \rightarrow P(Q)$

PDA

- $M = (Q, \Sigma, \Gamma, \delta, q, F)$:
- $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow Q \times \{N, R\} \times \Gamma_\varepsilon{}^*$

Finite control (δ) based on

- State
- Input symbol
- Variable popped from stack

Turing Machine

Finite Automata	Pushdown Automata	Turing Machine
Regular	Context-free	Regular, context-free, context-sensitive, recursively enumerable.

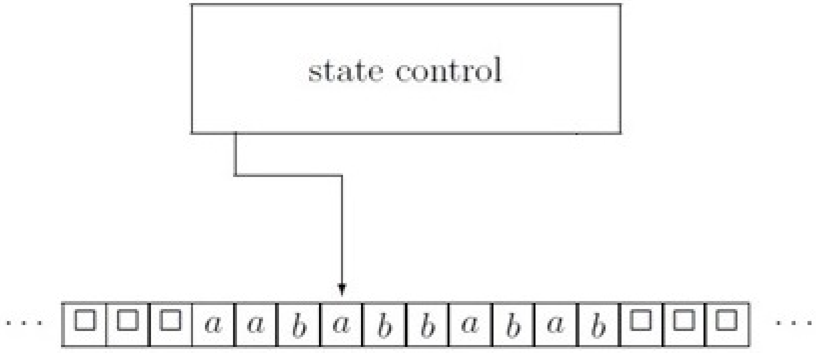
Previous machines can be used to accept or generate regular and context-free languages. However, they are not powerful enough to accept simple language such as

$$A = \{a^m b^n c^{mn} : m \geq 0, n \geq 0\}.$$

Turing machine is a simple model of real computer.

Turing Machine

- k ($k \geq 1$) infinitely long tape (The tape is infinite both to the left and to the right), divided into cells. Each cell stores a symbol belonging to Γ (tape alphabet).
- Tape head (↓) can move both right and left, one cell per move. It read from or write to a tape
- State control can be in any one of a finite number of states Q. It is based on: state and symbol read from tape
- Machine has one start state, one accept state and one reject state.
- Machine can run forever: infinite loop.



Properties of Turing Machine

- Turing machine can both read from tape and write on it.
- Tape head can move both right and left.
- Tape is infinite and can be used for storage.
- Accept and reject states take immediate effect.

Turing Machine

Example

Machine for language $A = \{\ s\#s \mid s \in \{0, 1\}^*\ \}$, input string is 01101#01101 ∈ A.

Turing Machine

Example

Turing Machine

Definition

A Turing machine (TM) is a 7-tuple $M = (\Sigma, \Gamma, Q, \delta, q, q_{accept}, q_{reject})$, where

- Σ is a finite set, called the input alphabet; the blank symbol ␣ is not contained in Σ,
- Γ is a finite set, called the tape alphabet; this alphabet contains the blank symbol ␣, and $\Sigma \subseteq \Gamma$,
- Q is a finite set, whose elements are called states,
- q is an element of Q; it is called the start state,
- q_{accept} is an element of Q; it is called the accept state,
- q_{reject} is an element of Q; it is called the reject state, $q_{reject} \neq q_{accept}$
- δ is called the transition function, which is a function $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$.
  L: move to left, R: move to right, N: no move.

Turing Machine

Transition function

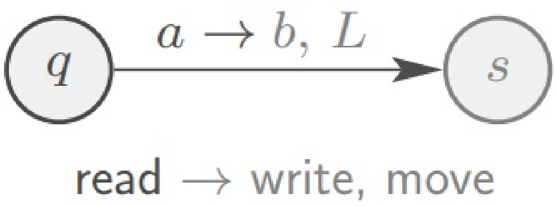
$\delta(q, a) = (s, b, L)$

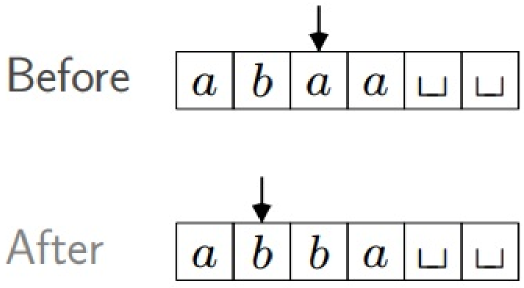
If TM

- in state $q \in Q$,
- tape head reads tape symbol $a \in \Gamma$

Then TM

- moves to state $s \in Q$
- overwrites a with b ∈ Γ
- moves head left (i.e., L ∈ {L, R})





Turing Machine

Computation steps

- Before the computation step, the Turing machine is in a state $r \in Q$, and the tape head is on a certain cell.
- TM M proceeds according to transition function:

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$$

- Depending on r and k symbols read from tape:

  (a) switches to a state r' ∈ Q;

  (b) tape head writes a symbol of Γ in the cell it is currently scanning;

  (c) tape head moves one cell to the left or right or stay at the current cell.

- Computation continues until q_{reject} or q_{accept} is entered.
- Otherwise, M will run forever (input string is neither accepted nor rejected)

Turing Machine

Example

TM M for language

$$A = \{0^{2^n} \mid n \geq 0\},$$

which consists of strings of 0s whose length is a power of 2.

On input string w:

- Sweep left to right across the tape, crossing off every other 0.
- If in stage 1 the tape contained a single 0, accept.
- If in stage 1 the tape contained more than a single 0 and the number of 0s was odd, reject.
- Return the head to the left end of the tape.
- Go to stage 1.

Turing Machine

Example

Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_1, q_{accept}, q_{reject})$, where

$Q = \{q_1, q_2, q_3, q_4, q_5, q_{accept}, q_{reject}\}$

$\Sigma = \{0\}$

$\Gamma = \{0, X, \sqcup\}$

q_1 is start state

q_{accept} is accept state

q_{reject} is reject state

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

Turing Machine

Example

Example

Run M when input $w = 0000$

Example

Run M when input $w = 0000$

Turing Machine

- **Start configuration**. The input is a string over the input alphabet Σ. Initially, this input string is stored on the first tape, and the head of this tape is on the leftmost symbol of the input string.

- **Computation and termination**. Starting in the start configuration, the Turing machine performs a sequence of computation steps. The computation terminates at the moment when the Turing machine enters the accept state q_{accept} or the reject state q_{reject}. (If the machine never enters q_{accept} and q_{reject} the computation does not terminate.)

- **Acceptance**. The Turing machine M accepts the input string $w \in \Sigma^*$, if the computation on this input terminates in the state q_{accept}.

TM Configuration

Provides a "snapshot" of TM at any point during computation:

- state
- tape contents
- head location

Example

Configuration $1011q01$:

- current state is q
- LHS of tape is 1011
- RHS of tape is 01
- head is on RHS 0

TM Configuration

Definition

Configuration of a TM $M = (Q, \Sigma, \Gamma, \delta, q, q_{accept}, q_{reject})$ is a string uqv with $u, v \in \Gamma^*$ and $q \in Q$, and specifies that currently

- M is in state q
- tape contains uv
- tape head is pointing to the cell containing the first symbol in v.

TM Transitions

Definition

Configuration $C1$ yields configuration $C2$ if the Turing machine can legally go from $C1$ to $C2$ in a single step. For TM $M = (Q, \Sigma, \Gamma, \delta, q, q_{accept}, q_{reject})$, suppose

- $u, v \in \Gamma^*$
- $a, b, c \in \Gamma$
- $q_i, q_j \in Q$
- transition function $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$.

Example

configuration uaq_ibv yields configuration uq_jacv

if $\delta(q_i, b)=(q_j, c, L)$.

TM Computation

Definition

Given a TM $M = (Q, \Sigma, \Gamma, \delta, q, q_{accept}, q_{reject})$ and input string $w \in \Sigma^*$. M accepts input w if there is a finite sequence of configurations $C_1, C_2,\ldots,C_k$ for some $k \geq 1$ with

- C_1 is the starting configuration q0w
- C_i yields C_{i+1} for all $i = 1, \ldots, k - 1$ (sequence of configurations obeys transition function δ)
- C_k is an accepting configuration $uq_{accept}v$ for some $u, v \in \Gamma^*$.

Language accepted by TM

Definition

The language $L(M)$ accepted by the Turing machine M is the set of all strings in Σ^* that are accepted by M.

Language A is **Turing-recognizable** if there is a TM M such that $A = L(M)$

- Also called recursively enumerable or enumerable language.
- On an input $w \in L(M)$, the machine M can either halt in a rejecting state, or it can loop indefinitely.
- Turing-recognizable not practical because never know if TM will halt.

Decider

Definition

A **decider** is TM that halts on all inputs, i.e., never loops.

Language $A = L(M)$ is decided by TM M if on each possible input $w \in \Sigma^*$, the TM finishes in a halting configuration, i.e.,

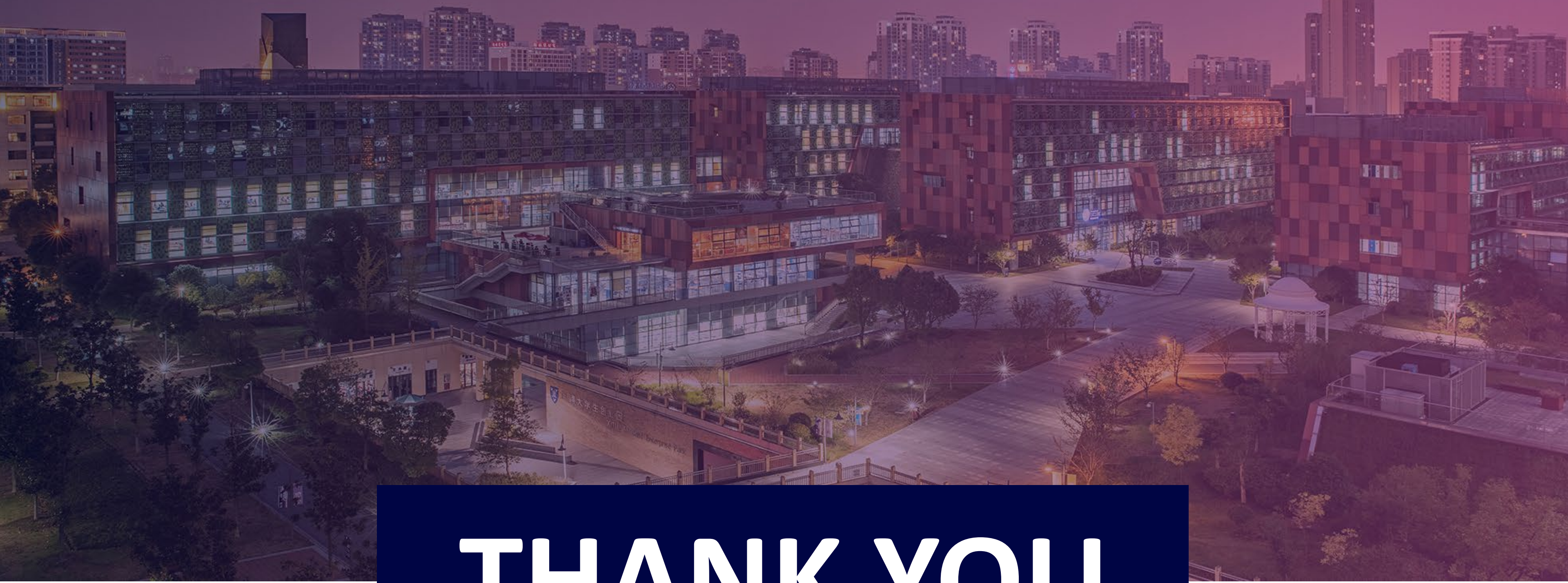
- M ends in q_{accept} for each $w \in A$
- M ends in q_{reject} for each $w \in A$.

A is **Turing-decidable** if $\exists$ TM M that decides A

- Also called recursive or decidable language.
- Differences to Turing-recognizable language:

  (a) Turing-decidable language has TM that halts on every string $w \in \Sigma^*$

  (b) TM for Turing-recognizable language may loop on strings $w \notin$ this language



THANK YOU

Xi’an Jiaotong-Liverpool University
西交利物浦大学

XJTLU | SCHOOL OF FILM AND TV ARTS