Modern Complexity Theory (CS1.405)

Dr. Ashok Kumar Das

IEEE Senior Member
Professor
Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad

E-mail: *ashok.das@iiit.ac.in*
URL: `http://www.iiit.ac.in/people/faculty/ashokkdas`
`https://sites.google.com/view/iitkgpakdas/`

Turing Machines

Turing Machines

- The Turing machine (TM) is a much more powerful model, which was first proposed by Alan Turing in 1936.
- A TM is similar to a finite automaton but with an unlimited and unrestricted memory, which is much more accurate model of a general-purpose computer.
- A TM can do everything that a real computer can do. Nonetheless, even a TM can not solve certain problems. In a very real sense, these problems are beyond the theoretical limits of computation.
- The Turing machine model uses an infinite tape as an unlimited memory.
  It has a tape head that can read and write symbols and move around on the tape.

Turing Machines

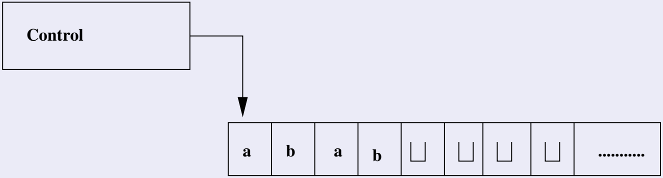

Figure: Schematic of a Turing machine

- Initially the tape contains only the input string and is blank everywhere else.
- If the machine needs to store information, it may write this information on the tape.

Turing Machines

- The outputs “accept” and “reject” are obtained by entering designated accepting and rejecting states.
- However, if it does not enter an accepting or a rejecting state, it will go on forever, never halting.

Turing Machines

Turing Machines

- Differences between finite automata and Turing machines
  1. A TM can both write on the tape and read from it.
  2. The read-write head can move to the left and to the right.
  3. The tape is infinite.
  4. The special states for rejecting and accepting take effect immediately.

Turing Machines

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Formal Definition of a Deterministic Turing Machine (DTM)

A Deterministic Turing Machine (DTM) is a 7-tuple $\langle Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject} \rangle$, where Q, Σ, Γ are all finite sets, and

- Q is the set of states,
- Σ is the input alphabet not containing the blank symbol $\sqcup$, e.g., Σ excluding $\sqcup$,
- Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$,
- $q_0 \in Q$ is the start state,
- $q_{accept} \in Q$ is the accept state,
- $q_{reject} \in Q$ is the reject state, where $q_{reject} \neq q_{accept}$,

Formal Definition of a Deterministic Turing Machine (DTM) (Continued...)

- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is the transition function.
  - It is the heart of the definition of a TM.
  - It tells the machine gets from one step to the next.
  - When the machine is in a certain state q and the head is over a tape square containing a symbol a, and if $\delta(q, a) = (r, b, L)$, the machine writes the symbol b replacing a, and goes to state r. The third component is either L or R, and indicates the head moves to the left or right after writing.
  - L indicates a move to the left.
  - R indicates a move to the right.

Turing Machines

Computation by a Deterministic Turing Machine (DTM) on an input string $w = w_1 w_2 \ldots w_n \in \Sigma^*$

A Deterministic Turing Machine (DTM) $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject} \rangle$ computes as follows

- Initially, M receives its input $w = w_1 w_2 \ldots w_n$ on the leftmost n squares of the tape, and the rest of the tape is blank (i.e., filled with blank symbols, $\sqcup$).
- The head starts on the leftmost square of the tape. Note that Σ does not contain the blank symbol, so the first blank appearing on the tape marks the end of the input.
- Once M has started, the computation proceeds according to the rules described by the transition function, δ.

Turing Machines

Computation by a Deterministic Turing Machine (DTM) on an input string $w = w_1 w_2 \ldots w_n \in \Sigma^*$

- Note that if M ever tries to move its head to the left off the the left-hand end of the tape, the head stays in the same place for that move, even though the transition function, δ indicates L.
- The computation continues until it enters either the accept or reject states at which point it halts. If neither occurs, M goes on forever.

Configuration of a Turing Machine

- As a TM computes, changes occur in the current state, the current tape contents, and the current head location. A setting of these three items is called a *configuration* of the Turing machine.
- For a state q and two strings u and v over the tape alphabet Γ, we write uqv for the configuration where state is q, the current tape contents is uv and the current head location is the first symbol of v.

Turing Machines

Example [Configuration of a Turing Machine]

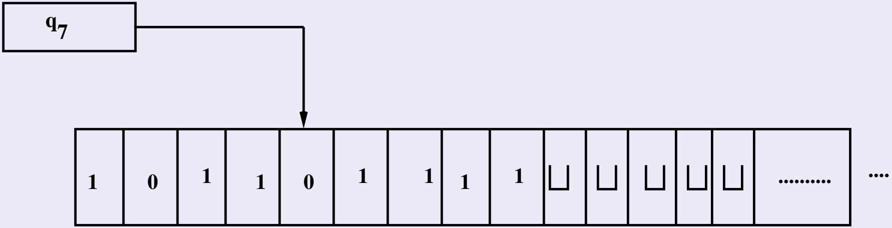

Figure: A Turing machine with configuration $1011q_7 01111$

Turing Machines

Configuration of a Turing Machine

- We say that the configuration C_1 "yields" another configuration C_2 if the Turing machine can legally go from C_1 to C_2 in a single step.
- Define this notion formally as follows:
  - Suppose that we have a, b, and $c \in \Gamma$ and u and $v \in \Gamma^*$, and states q_i and q_j.
  - Let uaq_ibv and uq_jacv be two configurations.
  - We say uaq_ibv "yields" uq_jacv if the transition function is $\delta(q_i, b) = (q_j, c, L)$.
  - We say uaq_ibv "yields" $uacq_jv$ if the transition function is $\delta(q_i, b) = (q_j, c, R)$.

Configuration of a Turing Machine

- The start configuration of M on input w is the configuration q_0w. It indicates that the machine is in the start state q_0 with its head at the leftmost position on the tape.
- In an accepting configuration the state of the configuration is q_{accept}.
- In a rejecting configuration the state of the configuration is q_{reject}.
- We say that accepting and rejecting configurations are "halting" configurations and do not yield further configurations.

Turing Machines

Configuration of a Turing Machine

A deterministic Turing machine (DTM) M accepts input w if a sequence of configurations $C_1, C_2, \ldots, C_k$ exists, where

1. C_1 is the start configuration of M on input w,
2. Each C_i yields C_{i+1}, and
3. C_k is accepting configuration.

Turing Machines

Configuration of a Turing Machine

Definition

A DTM M that accepts a collection (set) of strings is called the language of M, or the language recognized by M, denoted by $L(M)$. Thus, $L(M) = \{w | M \, \text{accepts} \, w\}$.

Definition

A language L is called Turing-recognizable or recursively enumerable language if some Turing machine recognizes it.

Turing Machines

Outcomes of a Turing machine on input

When we start a Turing machine on input, three outcomes are possible.

1. accept
2. reject
3. loop

Turing Machines

Outcomes of a Turing machine on input

- By loop, we mean that the machine simply does not halt.
- Sometimes distinguishing a machine that is looping from one that is merely taking a long time is difficult.
- For this reason, we prefer Turing machines that halt on all inputs; such machines never loop.
- These machines are called “deciders” because they always make a decision to accept or reject.
- A decider that recognizes some language also is said to decide that language.

Turing Machines

Outcomes of a Turing machine on input

Definition

A language L is Turing-decidable or simply decidable (or a recursive language) if some Turing machine decides it.

Theorem

Every decidable language is Turing-recognizable, but certain Turing-recognizable languages are not decidable.

Problem: Consider the language $L = \{a^i b^j c^k | i \times j = k \text{ and } i, j, k \geq 1\}$. Design a DTM M that decides L.

Turing Machines

Variants of Turing Machines [Multi-Tape Turing Machines]

- A multi-tape Turing machine (MTM) is like an ordinary Turing machine (TM) with several tapes.
- Each tape has its own head for reading and writing.
- Initially, the input appears on tape 1, and the others start out blank.
- The transition function δ is changed to allow for reading, writing, and moving the heads on some or all of the tapes simultaneously.
- Formally, it is defined as $\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, S\}^k$, where k is the number of tapes.
- The expression $\delta(q_i, a_1, a_2, \ldots, a_k) = (q_j, b_1, b_2, \ldots, b_k, L, R, \ldots, L)$ means that, if the machine is in state q_i and heads 1 through k for reading symbols a_1 through a_k, the machine goes to state q_j, writes symbols b_1 through b_k, and directs each head to move left (L) or right (R) or to stay put (S), as specified.

Variants of Turing Machines [Multi-Tape Turing Machines]

Theorem

Every multi-tape Turing machine has an equivalent single-tape Turing machine.

Corollary

A language is Turing-recognizable if and only if some multi-tape Turing machine recognizes it.

Variants of Turing Machines [Multi-Tape Turing Machines]

Problem: Consider the following multi-tape Turing machine *M* as shown below. Design a single-tape Turing machine, *S*.

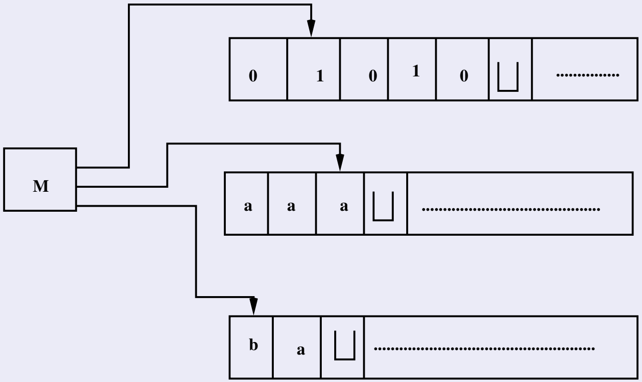

Figure: Multi-tape Turing machine

Variants of Turing Machines [Multi-Tape Turing Machines]

Solution: The "dotted" tape symbols are simply new symbols that have been added to the tape alphabet, Γ. The new symbol # is used as a delimiter to separate the contents of the different tapes.

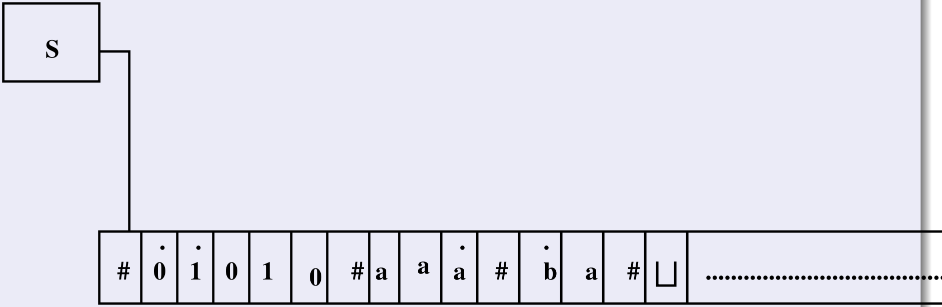

Figure: Single-tape Turing machine

Nondeterministic Turing Machines

- A nondeterministic Turing machine (NTM) is defined as follows.
  - At any point in a computation, the machine may proceed according to several possibilities.
  - The transition function for a nondeterministic Turing machine has the form:
    $\delta : Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R\})$,
    where $\mathcal{P}(S)$ represents the power set of S.
  - The computation of a nondeterministic Turing machine is a tree whose branches correspond to different possibilities for the machine. If some branch of the computation leads to the accept state, the machine accepts its input.

Nondeterministic Turing Machines

Theorem

Every nondeterministic Turing machine has an equivalent deterministic Turing machine.

Corollary

A language is Turing-recognizable if and only if some nondeterministic Turing machine recognizes it.

Corollary

A language is decidable if and only if some nondeterministic Turing machine decides it.

Turing Machines

Problem [Deterministic Turing Machines]: Let the language A be $A = \{0^k 1^k | k \geq 0\}$. Design a Turing machine that decides it.

Solution: The following is a low-level description of a DTM M that decides the language A:

$M =$ "On input string w:

1. Scan across the tape and "reject" if a 0 is found to the right of a 1.
2. Repeat if both 0s and 1s remain on the tape:
   - Scan across the tape, crossing off a single 0 and a single 1.
3. If 0s still remain after all the 1s have been crossed off, or if 1s still remain after all the 0s have been crossed off, "reject". Otherwise, if neither 0s nor 1s remain on the tape, "accept"."

End of this lecture