Lecture #10: Introduction to Turing Machines
Key Concepts

Definition of a Turing Machine

Definition 1. A ***Turing machine*** is a 7-tuple,

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}}),$$

where

1. Q is the (finite and nonempty) set of ***states***, and is sometimes called the ***finite control*** of M;
2. Σ is the (finite and nonempty) ***input alphabet*** — and M processes strings in $\Sigma^\star$. Σ *does not* include the ***blank symbol*** $\sqcup$.
3. Γ is the (finite and nonempty) ***tape alphabet***; $\Sigma \subseteq \Gamma$, $\sqcup \in \Gamma$ — and Γ may also include a finite number of additional symbols that are not in $\Sigma \cup \{\sqcup\}$. However, $Q \cap \Gamma = \emptyset$.
4. The ***transition function*** is a *partial* function
   $$\delta : Q \times \Gamma \to Q \times \Gamma \times \{\mathtt{L}, \mathtt{R}\}.$$
   - $\delta(q, \sigma)$ should be defined, for every state
     $$q \in Q \setminus \{q_{\text{accept}}, q_{\text{reject}}\}$$
     and for every symbol $\sigma \in \Gamma$.
   - Neither $\delta(q_{\text{accept}}, \sigma)$ nor $\delta(q_{\text{reject}}, \sigma)$ should be defined for any symbol $\sigma \in \Gamma$.
5. $q_0 \in Q$ is the ***start state***.
6. $q_{\text{accept}} \in Q$ is the ***accept state***.
7. $q_{\text{reject}} \in Q \setminus \{q_{\text{accept}}\}$ is the ***reject state***.

A Turing machine M accesses and stores on a one-way information ***tape*** — consisting of an infinite sequence of “cells” that each store a symbol in Γ. M also has a ***tape head*** which points to exactly one cell on the tape at any time.

Configurations — and Their Representations as Strings

At any point of a Turing machine M's computation on an input string, M is exactly one state in Q. All but finitely many cells of the tape store the blank symbol, "$\sqcup$". A ***configuration*** of M includes all of this information, along with the location of the tape head.

A configuration of M can be represented using a string in $(Q \cup \Gamma)^\star$, which stores exactly copy of a symbol in Q (so that all the other symbols in the string belong to Γ).

In particular, $q \in Q$ and let $\omega_1, \omega_2 \in \Gamma^\star$ such that ω_2 does not end with "$\sqcup$". Then the string

$$\omega_1 \, q \, \omega_2$$

represents a configuration in which M is in state q. The tape head points to a cell whose distance from from the leftmost cell of tape is $|\omega_1$ — so that the cell is pointing to the leftmost cell of the tape if and only if $|\omega_1| = 0$ (so that $\omega_1 = \lambda$) — and the cells on the tape, that are to the left of the tape head, store the symbols in ω_1. The string ω_2 represents the contents of the cells of the tape, beginning with the location of the location, and proceeding to the right until the last non-blank symbol on the tape — so that $\omega_2 = \lambda$ if and only if all the non-blank symbols on the tape are stored in cells that are to the left of the current location of the tape head.

Initial Configuration

Let $\omega \in \Sigma^\star$. When M's execution on input ω begins, M is in its start state, q_0.The string ω is written on the leftmost cells of the tape, and all other cells of the tape store copies of "$\sqcup$". The tape points to the leftmost cell of the tape — so that M's ***initial configuration*** for ω is represented by the string

$$q_0 \, \omega$$

Moves of M — Applying the Transition Function

Suppose again that M is in a configuration (represented by the string)

$$\omega_1 \, q \, \omega_2$$

where $\omega_1, \omega_2 \in \Gamma^\star$ — and where $q \in Q \setminus \{q_{\text{accept}}, q_{\text{reject}}\}$. Let $\sigma \in \Gamma$ such that σ is the leftmost symbol in ω_2 if ω_2 is not the empty string and such that $\sigma = \sqcup$ otherwise — so that σ is the symbol that is currently visible on M's tape.

Then M takes at least one more step during this conversation, and transition function is used to determine what should happen next: If

$$\delta(q, \sigma) = (r, \tau, d)$$

where $r \in Q$, $\tau \in \Gamma$, and $d \in \{\mathtt{L}, \mathtt{R}\}$, then the copy of "$\sigma$" that is currently visible on the tape should be replaced by a copy of "τ". The tape head should be moved from the cell that is currently visible, to one that is next to it — either the one to the left if $d = \mathtt{L}$ or the one to the right if $d = \mathtt{R}$ — except when $d = \mathtt{L}$ and the tape head is already pointing to the leftmost cell on the tape. In this last case, the tape head's location should not change. The Turing machine's state should be changed to r.

Definition 2. Let $r_1, r_2 \in Q$, where $r_1 \notin \{q_{\text{accept}}, q_{\text{reject}}\}$, and let $\omega_{1,1}, \omega_{1,2}, \omega_{2,1}, \omega_{2,2} \in \Gamma^\star$. Then configuration $\omega_{1,1}\, r_1\, \omega_{1,2}$ ***yields*** configuration $\omega_{2,1}\, r_2\, \omega_{2,2}$ — written

$$\omega_{1,1}\, r_1\, \omega_{1,2} \vdash \omega_{2,1}\, r_2\, \omega_{2,2}$$

— if configuration $\omega_{2,1}\, r_2\, \omega_{2,1}$ is reached from configuration $\omega_{1,1}\, r_1\, \omega_{2,1}$ by taking a ***single step*** of M.

Configuration $\omega_{1,1}\, r_1\, \omega_{2,1}$ ***derives*** configuration $\omega_{2,1}\, r_2\, \omega_{2,2}$ — written

$$\omega_{1,1}\, r_1\, \omega_{1,2} \vdash^\star \omega_{2,1}\, r_2\, \omega_{2,2}$$

— if configuration $\omega_{2,1}\, r_2\, \omega_{2,2}$ is reached from configuration $\omega_{1,1}\, r_1\, \omega_{1,2}$ by taking ***zero or more*** (but finitely many) steps of M.

Processing a String

Definition 3. Let $\omega \in \Sigma^\star$. Then M ***accepts*** ω if M reaches an ***accepting configuration*** (that is, a configuration including state q_{accept}) after a finite number of steps — that is, if

$$q_0\, \omega \vdash^\star \omega_1\, q_{\text{accept}}\, \omega_2$$

for some strings $\omega_1, \omega_2 \in \Gamma^\star$.

On the other hand, M ***rejects*** ω if M reaches a ***rejecting configuration*** (that is, a configuration including state q_{reject}) after a finite number of steps — that is, if

$$q_0\, \omega \vdash^\star \omega_1\, q_{\text{reject}}\, \omega_2$$

for some strings $\omega_1, \omega_2 \in \Gamma^\star$.

Finally, M ***loops on*** ω if M does not accept ω and M does not reject ω — so that M's computation on ω never halts.

Processing Languages

Definition 4. Let $L \subseteq \Sigma^\star$. Then M ***recognizes*** L if M accepts ω for every string $\omega \in \Sigma^\star$ such that $\omega \in L$ and M either ***rejects*** or ***loops on*** every string $\omega \in \Sigma^\star$ such that $\omega \notin L$.

We say that L is the ***language, $L(M)$, of M*** if M recognizes L.

A language $L \subseteq \Sigma^\star$ is ***Turing-recognizable*** (often just called"*recognizable*") if there exists a Turing machine M that recognizes L.

Definition 5. Let $L \subseteq \Sigma^\star$. Then M ***decides*** L if M accepts ω for every string $\omega \in \Sigma^\star$ such that $\omega \in L$ and M ***rejects*** ω for every string $\omega \in \Sigma^\star$ such that $\omega \notin L$.

A language $L \subseteq \Sigma^\star$ is ***Turing-decidable*** (often just called "*decidable*") if there exists a Turing machine M that decides L.

It follows from these definitions that if L is Turing-decidable then L is also Turing-recognizable, for all $L \subseteq \Sigma^\star$. We will see, soon, that there exist languages that are recognizable, but not decidable. There also exist languages that are not recognizable.