Computer Science 351

Introduction to Turing Machines

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #11

Goals for Today

- Introduction to a more powerful kind of automaton — a ***Turing machine***

Informal Definition of a Turing Machine

A ***Turing machine***

- processes strings over some ***input alphabet*** Σ;
- has a ***finite control*** consisting of a finite set Q of ***states***;
- also has access to a (one-way) infinite ***tape*** whose *cells* store elements of a larger ***tape alphabet*** Γ:
  - $\Sigma \subseteq \Gamma$;
  - Γ also includes a special symbol, "blank" — drawn as $\sqcup$. $\sqcup \notin \Sigma$.
  - Γ can include a finite number of other symbols that are not in $\Sigma \cup \{\sqcup\}$ as well.

Informal Definition of a Turing Machine

- A Turing machine also has a “tape head” which points to exactly one cell on the tape at any time.
- A ***configuration*** of a Turing machine, with a set of states Q and tape alphabet Γ, includes
  - the current state (which belongs to Q),
  - the non-blank part of the tape (a string in $\Gamma^\star$ stored in the leftmost cells of the tape), and
  - the location of the tape head.

Informal Definition of a Turing Machine

- A configuration of a Turing machine, with a set of states Q and tape alphabet Γ, can be represented by a string (of symbols over $Q \cup \Gamma$)

  $$\mu\, q\, \nu$$

  where
  - $\mu \in \Gamma^\star$ is the string stored in cells to the *left* of the tape head,
  - $q \in Q$ is the current state, and
  - $\nu \in \Gamma^\star$ is the string stored in cells *at*, and to the *right* of the tape head (with infinitely many copies of $\sqcup$ on the the tape, to the right of this).

Informal Definition of a Turing Machine

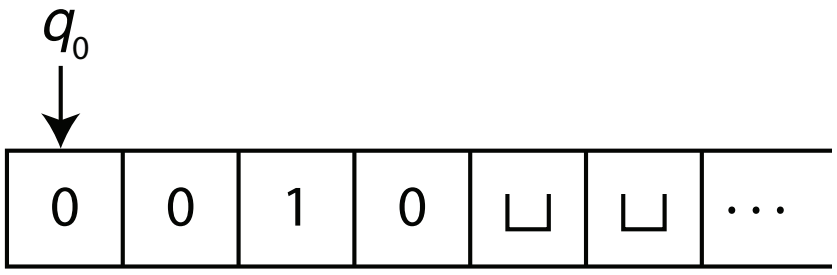
- At the beginning of an execution of this machine on a string
$$\omega = \sigma_1\sigma_2\ldots\sigma_n \in \Sigma^\star,$$
  - the finite control is in the Turing machine's ***start state***;
  - the first $n = |\omega|$ cells of the tape store the symbols $\sigma_1, \sigma_2, \ldots, \sigma_n$ in the string ω (in order);
  - every other cell of the tape stores the "blank" symbol $\sqcup$;
  - the tape head is pointing to the leftmost cell on the tape.

  This is called the ***initial configuration*** (or the ***start configuration*** of the machine for the input ω.

Informal Description of a Turing Machine

Example:

- Consider a Turing machine with input alphabet $\Sigma = \{0, 1\}$ and start state q_0. The ***initial configuration*** on an input string $\omega = 0010 \in \Sigma^\star$ looks like



and is represented by the string

$$q_0 0010$$

Informal Definition of a Turing Machine

- The next "move" — or ***transition*** — that the machine can make will generally depend on *both*
  - its current *state* (an element of Q), and
  - the symbol (in Γ) stored at the current location of its tape head.

Informal Definition of a Turing Machine

- When it makes its next move, the machine will
  - Write a (possibly different) symbol in Γ onto the cell of the tape that is currently visible;
  - change the finite control to a (possibly different) state in Q;
  - move the location of the tape head, either one position to the ***left*** or one position to the ***right***.

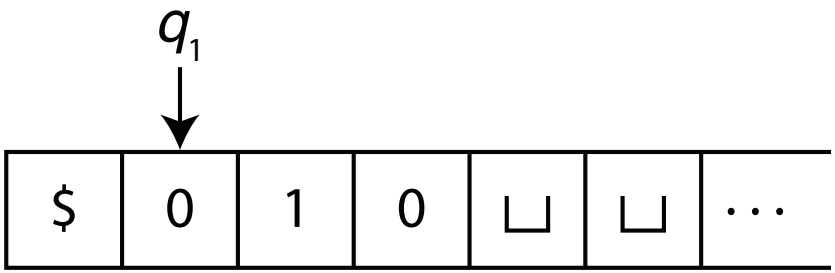
  Exception: If the tape head is already at the leftmost cell and the move would (otherwise) be to the ***left*** then the location of the tape head does not change.
- Transitions are specified using a ***transition function*** — a *partial* function

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}\}.$$

Informal Description of a Turing Machine

Example Continued:

- Suppose the set of states also includes states q_1 and q_2 and that $\Gamma = \Sigma \cup \{\sqcup, \$\}$.
- Suppose, as well, that $\delta(q_0, 0) = (q_1, \$, \mathrm{R})$. Then, after the first step of the Turing machine's execution on the input string $\omega = 0010$, the configuration looks like



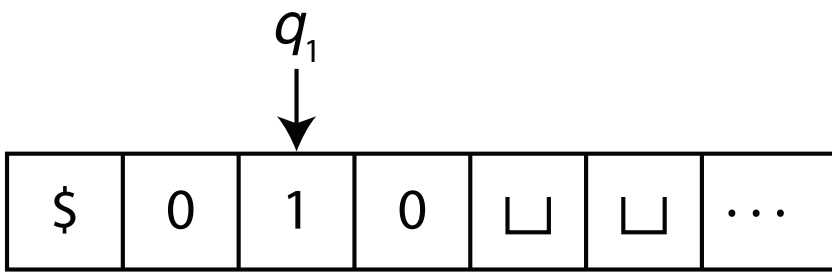
and is represented by the string

$$\$q_1 010$$

Informal Description of a Turing Machine

Example Continued:

- Suppose that $\delta(q_1, \mathtt{0}) = (q_1, \mathtt{0}, \mathtt{R})$. Then, after the second step of the Turing machine's execution on the input string $\omega = \mathtt{0010}$, the configuration looks like



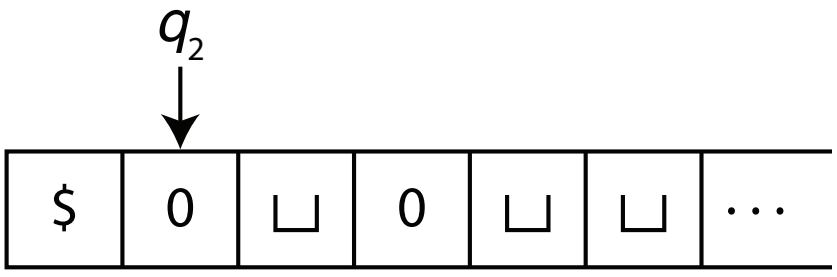
and is represented by the string

$$\$\mathtt{0}q_1\mathtt{10}$$

Informal Description of a Turing Machine

Example Continued:

- Suppose that $\delta(q_1, 1) = (q_2, \sqcup, \mathtt{L})$. Then, after the third step of the Turing machine's execution on the input string $\omega = 0010$, the configuration looks like



  and is represented by the string

$$\$q_2 0 \sqcup 0$$

- A supplemental document has a bit more information about how the transition function should be used.

Informal Definition of a Turing Machine

- The machine has two other special states — both of which might be the start state, but which cannot be the *same* state:
  - q_{accept} — the ***accept state***, and
  - q_{reject}— the ***reject state***.
- Configurations including q_{accept} are called ***accepting configurations***.
- Configurations including q_{reject} are called ***rejecting configurations***.
- Accepting configurations, and rejecting configurations, are called ***halting configurations***.
- All other configurations are called ***non-halting configurations***.

Informal Definition of a Turing Machine

- Transitions should be defined that can be used to continue from all *non-halting configurations* — so $\delta(q, \sigma)$ should be defined whenever $q \in Q \setminus \{q_{\text{accept}}, q_{\text{reject}}\}$ and for all $\sigma \in \Gamma$.
- Computations should end when "halting configurations" are reached — so $\delta(q_{\text{accept}}, \sigma)$ and $\delta(q_{\text{reject}}, \sigma)$ should *not* be defined, for any symbol $\sigma \in \Gamma$.

First Complete Example of a Turing Machine

Consider a Turing machine M with input alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, tape alphabet $\Gamma = \{\mathtt{a}, \mathtt{b}, \sqcup\}$, a set of states

$$Q = \{q_0, q_{\mathtt{a},1}, q_{\mathtt{a},2}, q_{\mathtt{b},1}, q_{\mathtt{b},2}, q_{\text{accept}}, q_{\text{reject}}\},$$

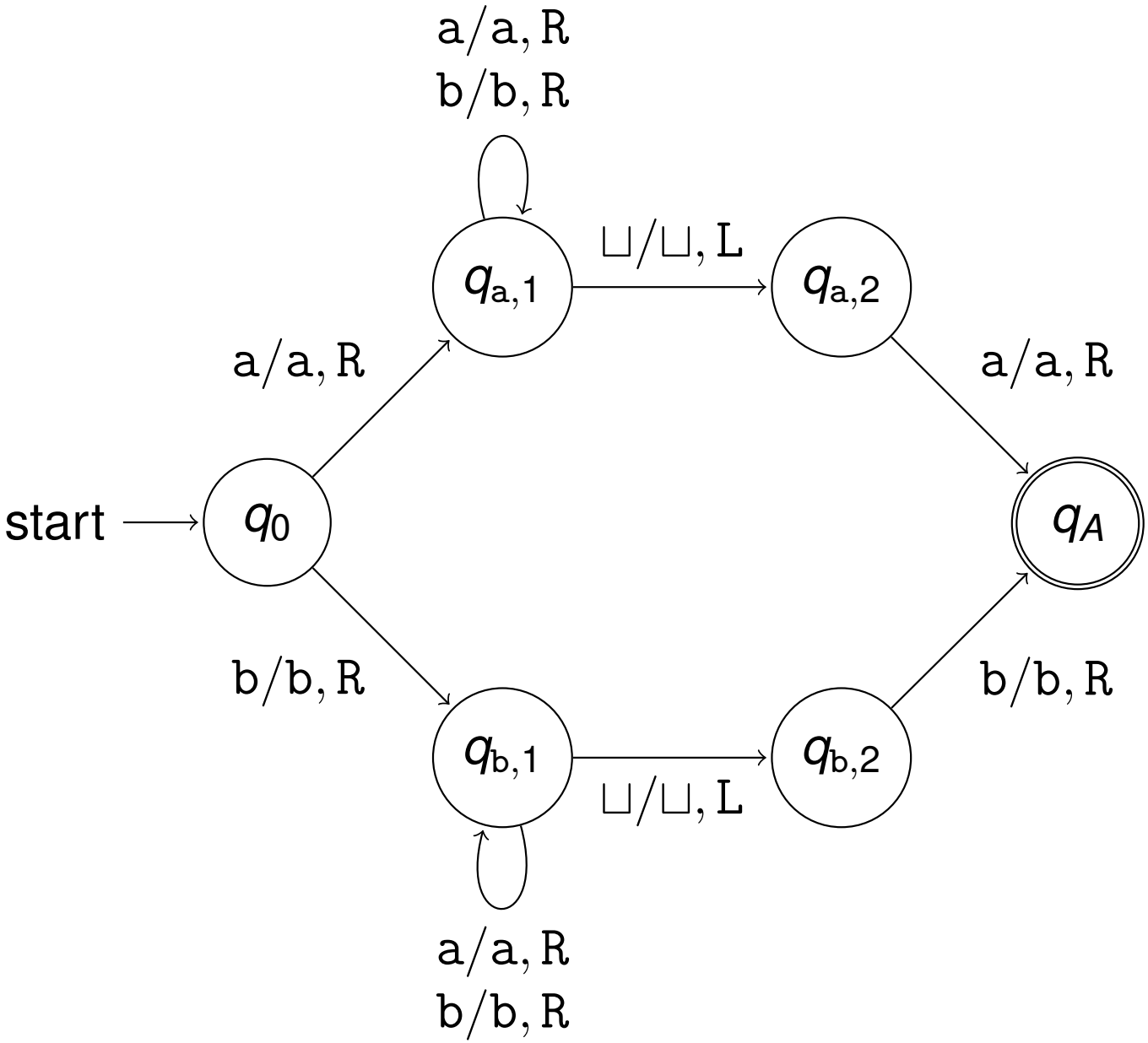
and whose transition function might be shown by a ***transition diagram*** that is as shown on the following slide.

- To keep the picture simple, the accepting state is shown as "q_A" instead of q_{accept}.
- To keep the picture, transitions to the rejecting state, and this state are left out — but

$$\delta(q, \sigma) = (q_{\text{reject}}, \sigma, \mathtt{R})$$

for $q = q_0$ and $\sigma = \sqcup$, for $q = \mathtt{a}, 2$ and either $\sigma = \mathtt{b}$ or $\sigma = \sqcup$, and for $q_{\mathtt{b},2}$ and either $\sigma = \mathtt{a}$ or $\sigma = \sqcup$.

First Complete Example of a Turing Machine



First Complete Example of a Turing Machine

The transition function can also be given using a ***transition table:***

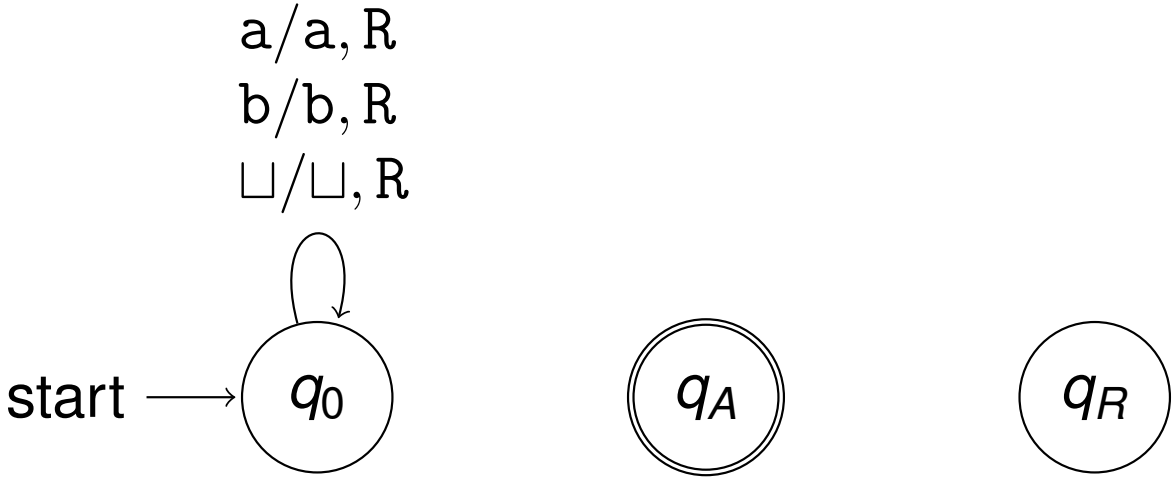
	a	b	⊔
q_0	$(q_{\mathtt{a},1}, \mathtt{a}, \mathtt{R})$	$(q_{\mathtt{b},1}, \mathtt{b}, \mathtt{R})$	$(q_{\text{reject}}, \sqcup, \mathtt{R})$
$q_{\mathtt{a},1}$	$(q_{\mathtt{a},1}, \mathtt{a}, \mathtt{R})$	$(q_{\mathtt{a},1}, \mathtt{b}, \mathtt{R})$	$(q_{\mathtt{a},2}, \sqcup, \mathtt{L})$
$q_{\mathtt{a},2}$	$(q_{\text{accept}}, \mathtt{a}, \mathtt{R})$	$(q_{\text{reject}}, \mathtt{b}, \mathtt{R})$	$(q_{\text{reject}}, \sqcup, \mathtt{R})$
$q_{\mathtt{b},1}$	$q_{\mathtt{b},1}, \mathtt{a}, \mathtt{R})$	$q_{\mathtt{b},1}, \mathtt{b}, \mathtt{R})$	$(q_{\mathtt{b},2}, \sqcup, \mathtt{L})$
$q_{\mathtt{b},2}$	$q_{\text{reject}}, \mathtt{a}, \mathtt{R})$	$(q_{\text{accept}}, \mathtt{b}, \mathtt{R})$	$(q_{\text{reject}}, \sqcup, \mathtt{R})$

Second Complete Example of a Turing Machine

Consider a Turing machine M with input alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, tape alphabet $\Gamma = \{\mathtt{a}, \mathtt{b}, \sqcup\}$, a set of states

$$Q = \{q_0, q_{\text{accept}}, q_{\text{reject}}\},$$

and whose transition function might be shown by a ***transition diagram*** is as follows (with the accepting and rejecting states shown as "q_A" and "q_R"):



Second Complete Example of a Turing Machine

The transition function can also be given using a ***transition table:***

	a	b	⊔
q_0	$(q_0, \mathtt{a}, \mathtt{R})$	$(q_0, \mathtt{b}, \mathtt{R})$	$(q_0, \sqcup, \mathtt{R})$

Formal Definition of a Turing Machine

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***;
2. Σ is the (finite and nonempty) ***input alphabet***. 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, $Q \cap \Gamma = \emptyset$.

Formal Definition of a Turing Machine

4. The ***transition function*** is a *partial* function

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}\}.$$

   - $\delta(q, \sigma)$ 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***.

First Complete Example of a Turing Machine

The first complete example of a Turing machine can be described as

$$M_1 = (Q_1, \Sigma_1, \Gamma_1, \delta_1, q_0, q_{\text{accept}}, q_{\text{reject}})$$

where

- $Q_1 = \{q_0, q_{\text{a},1}, q_{\text{a},2}, q_{\text{b},1}, q_{\text{b},2}, q_{\text{accept}}, q_{\text{reject}}\}$,
- $\Sigma_1 = \{\text{a}, \text{b}\}$,
- $\Gamma_1 = \{\text{a}, \text{b}, \sqcup\}$,
- q_0, q_{accept} and q_{reject} are M_1's start state, accept state and reject state, respectively,
- and the transition function δ_1, is as given on the following slide.

First Complete Example of a Turing Machine

- M_1's transition function is a partial function $\delta_1 : Q_1 \times \Gamma_1 \to Q_1 \times \Gamma_1 \times \{\mathrm{L}, \mathrm{R}\}$, as follows:
  - $\delta_1(q_0, \mathrm{a}) = (q_{\mathrm{a},1}, \mathrm{a}, \mathrm{R})$, $\delta_1(q_0, \mathrm{b}) = (q_{\mathrm{b},1}, \mathrm{b}, \mathrm{R})$, and $\delta_1(q_0, \sqcup) = (q_{\mathrm{reject}}, \sqcup, \mathrm{R})$;
  - $\delta_1(q_{\mathrm{a},1}, \mathrm{a}) = (q_{\mathrm{a},1}, \mathrm{a}, \mathrm{R})$, $\delta_1(q_{\mathrm{a},1}, \mathrm{b}) = (q_{\mathrm{a},1}, \mathrm{b}, \mathrm{R})$, and $\delta_1(q_{\mathrm{a},1}, \sqcup) = (q_{\mathrm{a},2}, \sqcup, \mathrm{L})$;
  - $\delta_1(q_{\mathrm{a},2}, \mathrm{a}) = (q_{\mathrm{accept}}, \mathrm{a}, \mathrm{R})$, $\delta_1(q_{\mathrm{b},2}, \mathrm{b}) = (q_{\mathrm{reject}}, \mathrm{b}, \mathrm{R})$, and $\delta_1(q_{\mathrm{a},2}, \sqcup) = (q_{\mathrm{reject}}, \sqcup, \mathrm{R})$;
  - $\delta_1(q_{\mathrm{b},1}, \mathrm{a}) = (q_{\mathrm{b},1}, \mathrm{a}, \mathrm{R})$, $\delta_1(q_{\mathrm{b},1}, \mathrm{b}) = (q_{\mathrm{b},1}, \mathrm{b}, \mathrm{R})$, and $\delta_1(q_{\mathrm{b},1}, \sqcup) = (q_{\mathrm{b},2}, \sqcup, \mathrm{L})$;
  - $\delta_1(q_{\mathrm{b},2}, \mathrm{a}) = (q_{\mathrm{reject}}, \mathrm{a}, \mathrm{R})$, $\delta_1(q_{\mathrm{b},2}, \mathrm{b}) = (q_{\mathrm{accept}}, \mathrm{b}, \mathrm{R})$, and $\delta_1(q_{\mathrm{b},2}, \sqcup) = (q_{\mathrm{reject}}, \sqcup, \mathrm{R})$;
  - $\delta_1(q_{\mathrm{accept}}, \mathrm{a})$, $\delta_1(q_{\mathrm{accept}}, \mathrm{b})$ and $\delta_1(q_{\mathrm{accept}}, \sqcup)$ are undefined; and
  - $\delta_1(q_{\mathrm{reject}}, \mathrm{a})$, $\delta_1(q_{\mathrm{reject}}, \mathrm{b})$, and $\delta_1(q_{\mathrm{reject}}, \sqcup)$ are undefined.

Second Complete Example of a Turing Machine

The second complete example of a Turing machine can be described as

$$M_2 = (Q_2, \Sigma_2, \Gamma_2, \delta_2, q_0, q_{\text{accept}}, q_{\text{reject}})$$

where

- $Q_2 = \{q_0, q_{\text{accept}}, q_{\text{reject}})$,
- $\Sigma_2 = \{\mathtt{a}, \mathtt{b}\}$,
- $\Gamma_2 = \{\mathtt{a}, \mathtt{b}, \sqcup\}$,
- q_0, q_{accept} and q_{reject} are M_2's start state, accept state and reject state, respectively,
- and the transition function for M_2 is a partial function $\delta_2 : Q_2 \times \Gamma_2 \to Q_2 \times \Gamma_2 \times \{\mathtt{L}, \mathtt{R})$ such that $\delta_2(q_0, \sigma) = (q_0, \sigma, \mathtt{R})$ and $\delta(q_{\text{accept}}, \sigma)$ and $\delta_2(q_{\text{reject}}, \sigma)$ are undefined, for every symbol $\sigma \in \Gamma_2$.

Tracing Executions: Notation

Let $\mu_1, \mu_2, \omega_1, \omega_2 \in \Gamma^\star$ and let $q, r \in Q$ — so that both of the strings

$$\mu_1\, q\, \mu_2 \qquad \text{and} \qquad \omega_1\, r\, \omega_2$$

represent configurations of M.

- $\mu_1\, q\, \mu_2 \vdash \omega_1\, r\, \omega_2$ — "$\mu_1 q \mu_2$ ***yields*** $\omega_1 r \omega_2$" — means that it is possible to go from the configuration of M represented by $\mu_1\, q\, \mu_2$ to the configuration represented by $\omega_1\, r\, \omega_2$, using ***one*** move of M, as defined above, and
- $\mu_1\, q\, \mu_2 \vdash^\star \omega_1\, r\, \omega_2$ — "$\mu_1 q \mu_2$ ***derives*** $\omega_1 r \omega_2$" — means that it is possible to go from the configuration of M represented by $\mu_1\, q\, \mu_2$ to the configuration represented by $\omega_1\, r\, \omega_2$, using ***zero or more*** (but only finitely many) moves of M, as defined above.

Tracing Executions: Notation

Consider the execution of M_1 on the input string $\mathtt{aba}$:

$$
\begin{aligned}
q_0\mathtt{aba} &\vdash \mathtt{a}q_{\mathtt{a},1}\mathtt{ba} && (\text{since } \delta_1(q_0, \mathtt{a}) = (q_{\mathtt{a},1}, \mathtt{a}, \mathtt{R})) \\
&\vdash \mathtt{ab}q_{\mathtt{a},1}\mathtt{a} && (\text{since } \delta_1(q_{\mathtt{a},1}, \mathtt{b}) = (q_{\mathtt{a},1}, \mathtt{b}, \mathtt{R})) \\
&\vdash \mathtt{aba}q_{\mathtt{a},1} && (\text{since } \delta_1(q_{\mathtt{a},1}, \mathtt{b}) = (q_{\mathtt{a},1}, \mathtt{a}, \mathtt{R})) \\
&\vdash \mathtt{ab}q_{\mathtt{a},2}\mathtt{a} && (\text{since } \delta_1(q_{\mathtt{a},1}, \sqcup) = (q_{\mathtt{a},2}, \sqcup, \mathtt{L}) \\
&\vdash \mathtt{aba}q_{\text{accept}} && (\text{since } \delta_1(q_{\mathtt{a},2}, \mathtt{a}) = (q_{\text{accept}}, \mathtt{a}, \mathtt{R})).
\end{aligned}
$$

Thus

$$q_0\mathtt{aba} \vdash^\star \mathtt{aba}q_{\text{accept}}.$$

Accepting, Rejecting, and Looping

Suppose, now, that $\omega \in \Sigma^\star$, where Σ is the input alphabet for a Turing machine M.

- M ***accepts*** ω if $q_0\omega \vdash^\star \mu_1\, q_{\text{accept}}\, \mu_2$ for some strings $\mu_1, \mu_2 \in \Gamma^\star$.
- M ***rejects*** ω if $q_0\omega \vdash^\star \mu_1\, q_{\text{reject}}\, \mu_2$ for some strings $\mu_1, \mu_2 \in \Gamma^\star$.
- M ***loops on*** ω if and only if M neither *accepts* nor *rejects* ω, as defined above.

Recognition and Decision

Let $L \subseteq \Sigma^\star$.

- A Turing machine M, with input alphabet Σ, ***recognizes*** L — and L is called the ***language*** of M — if
  - M ***accepts*** every string $\omega \in L$, and
  - M either ***rejects*** or ***loops*** on every string $\omega \in \Sigma^\star$ such that $\omega \notin L$.

  A language is ***Turing-recognizable*** — or simply ***recognizable*** — if it is the language of some Turing machine.

Recognition and Decision

Examples, Continued:

- The Turing machine in the first example ***recognizes*** the language

$$\{\omega \in \Sigma^\star \mid |\omega| \geq 1 \text{ and } \omega \text{ begins and ends with the same symbol}\}.$$

- The Turing machine in the second example ***recognizes*** the language $\emptyset$.

Recognition and Decision

Let $L \subseteq \Sigma^\star$.

- A Turing machine M, with input alphabet Σ, ***decides*** L if
  - M ***accepts*** every string $\omega \in L$, and
  - M ***rejects*** every string $\omega \in \Sigma^\star$ such that $\omega \notin L$.

  A language is ***Turing-decidable*** — or simply ***decidable*** — if there is a Turing machine that decides it.

Recognition and Decision

Examples, Continued:

- The Turing machine in the first example either accepts, or rejects, every string in $\Sigma_1^\star$ — so it also ***decides*** the language

$$\{\omega \in \Sigma^\star \mid |\omega| \geq 1 \text{ and } \omega \text{ begins and ends with the same symbol}\}.$$

- The Turing machine in the second example does not ***decide*** any language, at all — because it loops on some strings in $\Sigma_2^\star$.

Other Things To Think About

Other Things To Think About:

- What will Turing machines be used for?
- How can we *prove* things about Turing machines?
- Can Turing machines compute ***functions***? *How*?