Languages and Machines

L8: Turing machines

Jorge A. Pérez

Bernoulli Institute for Math, Computer Science, and AI
University of Groningen, Groningen, the Netherlands

May 21, 2025

Languages and Their Machines

Regular	$\leftrightarrow$	Finite State Machines (FSMs)
Context-free	$\leftrightarrow$	Pushdown Machines
Context-sensitive	$\leftrightarrow$	Linearly-bounded Machines
Decidable	$\leftrightarrow$	**Always-terminating Turing Machines**
Semi-decidable	$\leftrightarrow$	**Turing Machines**

Languages and Machines, Up to Here

- Finite state machines (FSMs), in different flavors, fully characterize regular languages.

Languages and Machines, Up to Here

- Finite state machines (FSMs), in different flavors, fully characterize regular languages.
- There are, however, languages that are not regular. Example:

$$L_1 = \{a^n\, b^n \mid n \geq 0\}$$

Hence, FSMs cannot recognize languages such as L_1.

Languages and Machines, Up to Here

- Finite state machines (FSMs), in different flavors, fully characterize regular languages.
- There are, however, languages that are not regular. Example:

$$L_1 = \{a^n\, b^n \mid n \geq 0\}$$

  Hence, FSMs cannot recognize languages such as L_1.
- Context-free languages strictly include regular languages, but also languages such as L_1.
  Pushdown machines, in different flavors, use a stack to fully characterize context-free languages (including L_1).

Languages and Machines, Up to Here

- Finite state machines (FSMs), in different flavors, fully characterize regular languages.
- There are, however, languages that are not regular. Example:

  $$L_1 = \{a^n\, b^n \mid n \geq 0\}$$

  Hence, FSMs cannot recognize languages such as L_1.
- Context-free languages strictly include regular languages, but also languages such as L_1.
  Pushdown machines, in different flavors, use a stack to fully characterize context-free languages (including L_1).
- There are, however, languages that are not context-free. Example:

  $$L_2 = \{a^n\, b^n\, c^n \mid n \geq 0\}$$

Languages and Machines, Up to Here

- Finite state machines (FSMs), in different flavors, fully characterize regular languages.
- There are, however, languages that are not regular. Example:

  $$L_1 = \{a^n\ b^n \mid n \geq 0\}$$

  Hence, FSMs cannot recognize languages such as L_1.
- Context-free languages strictly include regular languages, but also languages such as L_1.
  Pushdown machines, in different flavors, use a stack to fully characterize context-free languages (including L_1).
- There are, however, languages that are not context-free. Example:

  $$L_2 = \{a^n\ b^n\ c^n \mid n \geq 0\}$$

- What kind of machines do we need to recognize L_2?

Turing Machines (TMs)

Turing Machines (TMs)

- A Turing machine (TM) may access and modify any memory position, using a sequence of elementary operations
- No limitation on the space/time available for a computation
- A finite state machine equipped with a **tape**, divided into **squares**, which can be written on as a result of a transition
- The **head** of the machine can move to the right or to the left, allowing the TM to read and manipulate the input as desired

Turing Machines (TMs)

- A Turing machine (TM) may access and modify any memory position, using a sequence of elementary operations
- No limitation on the space/time available for a computation
- A finite state machine equipped with a **tape**, divided into **squares**, which can be written on as a result of a transition
- The **head** of the machine can move to the right or to the left, allowing the TM to read and manipulate the input as desired

In other words, a transition:

- changes the state
- writes a symbol on the square scanned by the head
- moves the head

Turing Machines (TMs)

A (simple) **Turing machine** M is a quintuple $(Q, \Sigma, \Gamma, \delta, q_0)$ where

- Q is a set of **states**
- $q_0 \in Q$ is the **start state**
- Γ is the **tape alphabet**, a set of symbols disjoint from Q. Contains a **blank symbol** $\mathtt{B}$, not in Σ
- $\Sigma \subseteq \Gamma \setminus \{\mathtt{B}\}$ is the **input alphabet**
- The transition function δ is a partial function such that

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

If $\delta(q, X)$ is undefined then $\delta(q, X) = \perp$.

Turing Machines (TMs)

A (simple) **Turing machine** M is a quintuple $(Q, \Sigma, \Gamma, \delta, q_0)$ where

- Q is a set of **states**
- $q_0 \in Q$ is the **start state**
- Γ is the **tape alphabet**, a set of symbols disjoint from Q. Contains a **blank symbol** $\mathtt{B}$, not in Σ
- $\Sigma \subseteq \Gamma \setminus \{\mathtt{B}\}$ is the **input alphabet**
- The transition function δ is a partial function such that

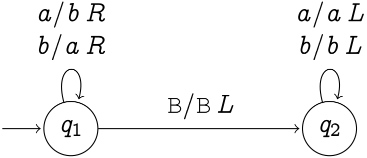
$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$$

  If $\delta(q, X)$ is undefined then $\delta(q, X) = \perp$.

A set of accepting states $F \subseteq Q$ is possible but not indispensable for defining acceptance (see later).

Example 1

A TM that reads the input string and interchanges symbols a and b:

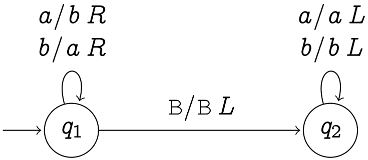


In state q_1, label '$a/b\ R$' indicates:

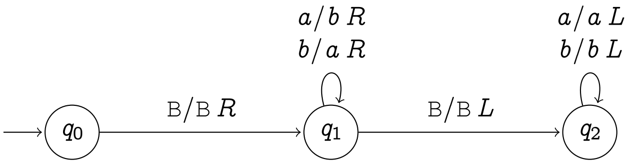
- symbol a is rewritten into b, and
- the head moves right (R).

Example 1

A TM that reads the input string and interchanges symbols a and b:



A slightly more general machine:



Turing Machines (TMs)

The global state of the TM is determined by the state $q \in Q$, the contents of the tape (a string in Γ^*) and the position of the head

- A **configuration** of the TM is a string uqv in $\Gamma^* Q \Gamma^*$, in which:
  - u is a string on the tape to the left of the head
  - q is the **current** state
  - v is a string on the tape that begins under the head
- The initial configuration is $q_0 w$, where $w \in \Sigma^*$ is the input string
- The first symbol of $v\mathtt{B}^\infty$ is called the **current** symbol

Turing Machines (TMs)

Suppose X, Y, Z are tape symbols (in Γ).
Moving to the next configuration:

$$\begin{aligned}
\delta(q, X) = (r, Y, R) \quad &\Rightarrow \quad u\,Z\,q\,X\,v \vdash u\,Z\,Y\,r\,v \\
\delta(q, X) = (r, Y, L) \quad &\Rightarrow \quad u\,Z\,q\,X\,v \vdash u\,r\,Z\,Y\,v \\
\delta(q, X) = \perp \quad &\Rightarrow \quad u\,q\,X\,v \vdash \perp
\end{aligned}$$

Turing Machines (TMs)

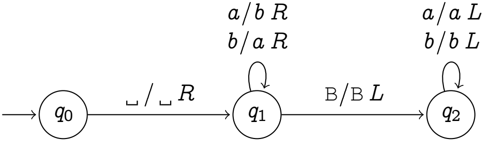
Suppose X, Y, Z are tape symbols (in Γ).
Moving to the next configuration:

$$\begin{array}{rcl} \delta(q,X) = (r,Y,R) & \Rightarrow & u\,Z\,q\,X\,v \vdash u\,Z\,Y\,r\,v \\ \delta(q,X) = (r,Y,L) & \Rightarrow & u\,Z\,q\,X\,v \vdash u\,r\,Z\,Y\,v \\ \delta(q,X) = \perp & \Rightarrow & u\,q\,X\,v \vdash \perp \end{array}$$

- A computation is a sequence of steps, as defined by $\vdash$
- A TM **computes a function** f
  - if starting in $q_0 w$, the final tape upon termination is always $\mathtt{B}^\infty u \mathtt{B}^\infty$, with $u = f(w)$.

Example 1, Revisited

Suppose there is a special symbol '␣' preceding the input string. We can adapt the previous machine as follows:



Computation for input $abab$:

$$
\begin{array}{l}
\rightarrow\ [q_0\rangle\ \sqcup\ a\ b\ a\ b\ \mathrm{B} \\
\vdash \sqcup\ [q_1\rangle\ a\ b\ a\ b\ \mathrm{B} \\
\vdash \sqcup\ b\ [q_1\rangle\ b\ a\ b\ \mathrm{B} \\
\vdash \sqcup\ b\ a\ [q_1\rangle\ a\ b\ \mathrm{B} \\
\vdash \sqcup\ b\ a\ b\ [q_1\rangle\ b\ \mathrm{B} \\
\vdash \sqcup\ b\ a\ b\ a\ [q_1\rangle\ \mathrm{B}
\end{array}
$$

$$
\begin{array}{l}
\vdash \sqcup\ b\ a\ b\ [q_2\rangle\ a\ \mathrm{B} \\
\vdash \sqcup\ b\ a\ [q_2\rangle\ b\ a\ \mathrm{B} \\
\vdash \sqcup\ b\ [q_2\rangle\ a\ b\ a\ \mathrm{B} \\
\vdash \sqcup\ [q_2\rangle\ b\ a\ b\ a\ \mathrm{B} \\
\vdash\ [q_2\rangle\ \sqcup\ b\ a\ b\ a\ \mathrm{B}
\end{array}
$$

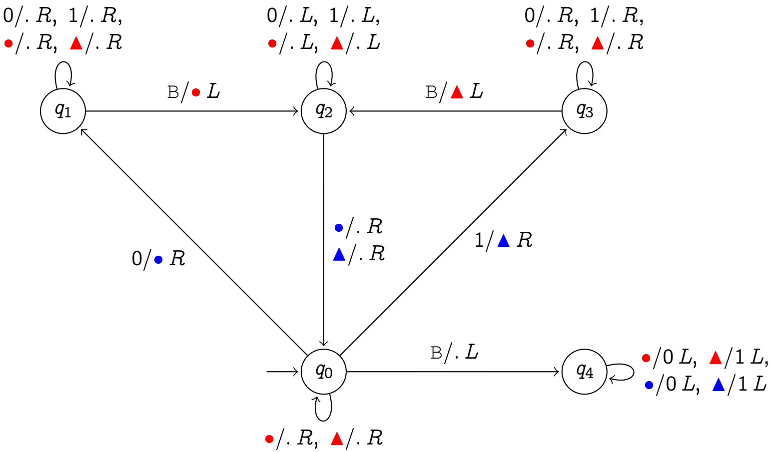
Example 2

A TM that duplicates the input string $w \in \{0, 1\}^*$.

- Before: A tape with the string w
- After: The tape contains the string $w\,w$
- What is your (programming) strategy?

Example 2

A TM that duplicates the input string $w \in \{0, 1\}^*$.



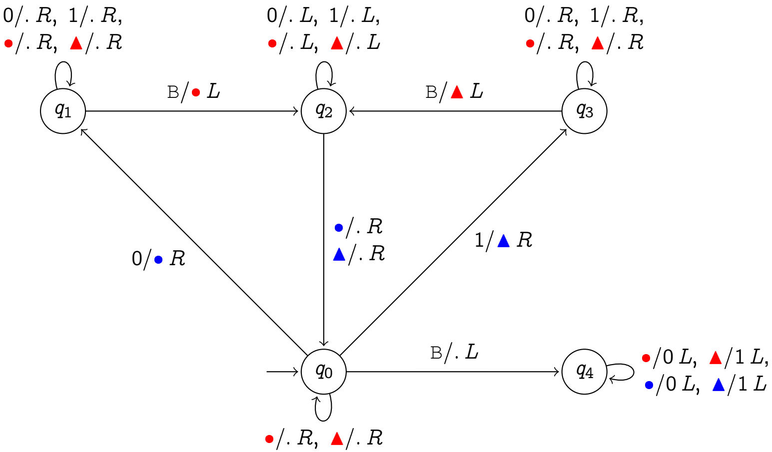
↓

B^∞	0	1	B	B	B^∞

(State = q_0)

Example 2

A TM that duplicates the input string $w \in \{0, 1\}^*$.

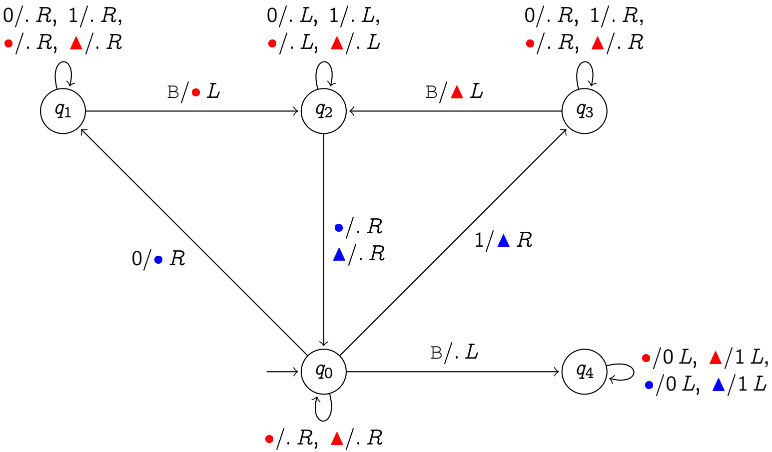


B^∞	●	1	B	B	B^∞

(State = q_1)

Example 2

A TM that duplicates the input string $w \in \{0, 1\}^*$.

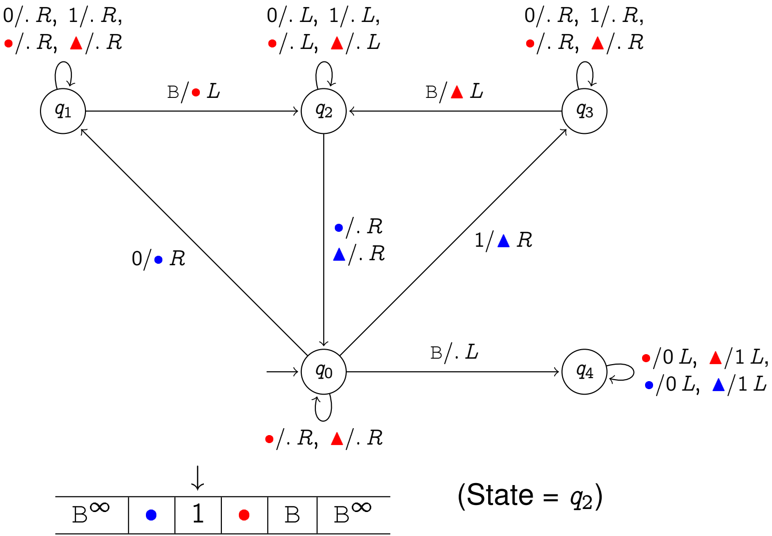


			↓		
B^∞	●	1	B	B	B^∞

(State = q_1)

Example 2

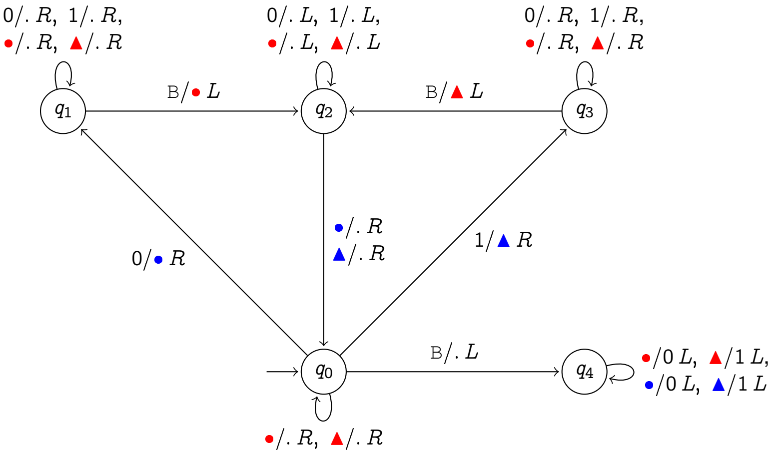
A TM that duplicates the input string $w \in \{0,1\}^*$.

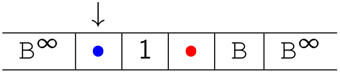


(State = q_2)

Example 2

A TM that duplicates the input string $w \in \{0, 1\}^*$.

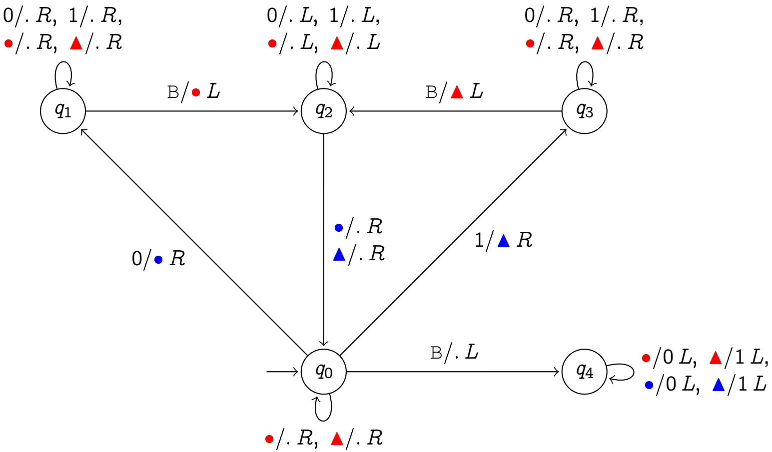




(State = q_2)

Example 2

A TM that duplicates the input string $w \in \{0, 1\}^*$.

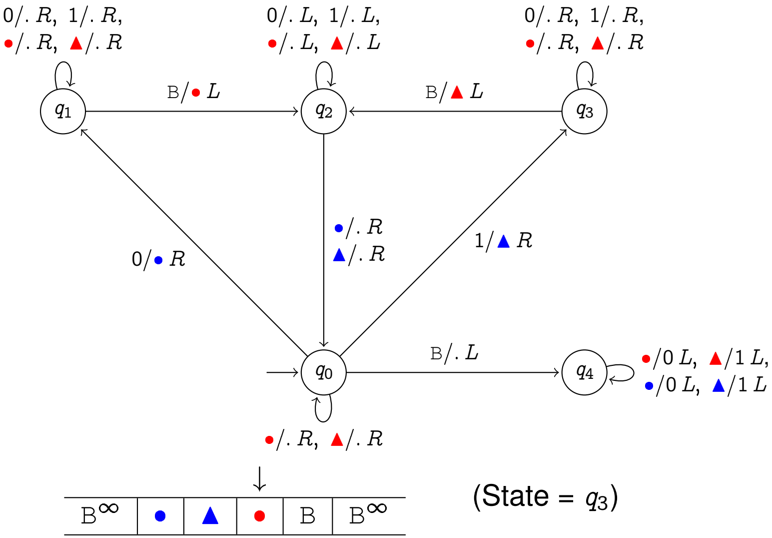


$\mathtt{B}^\infty$	$\bullet$	1	$\bullet$	$\mathtt{B}$	$\mathtt{B}^\infty$

(State = q_0)

Example 2

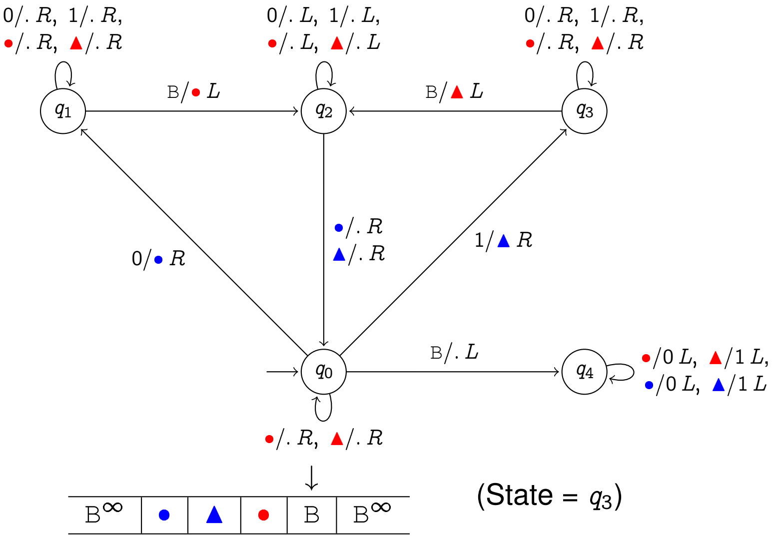
A TM that duplicates the input string $w \in \{0, 1\}^*$.



(State = q_3)

Example 2

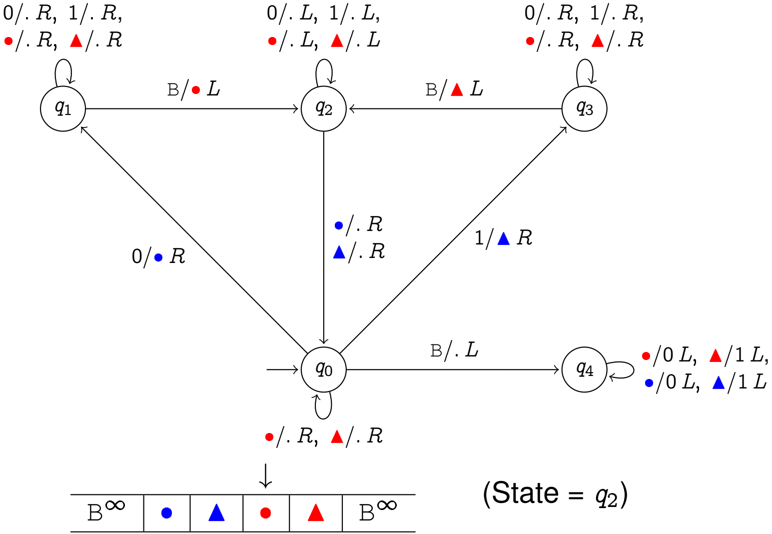
A TM that duplicates the input string $w \in \{0, 1\}^*$.



(State = q_3)

Example 2

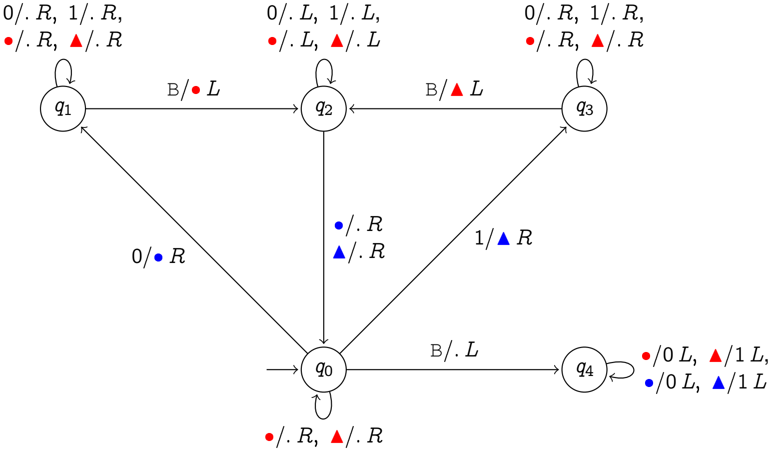
A TM that duplicates the input string $w \in \{0,1\}^*$.

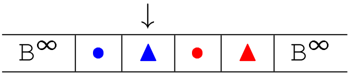


(State = q_2)

Example 2

A TM that duplicates the input string $w \in \{0, 1\}^*$.

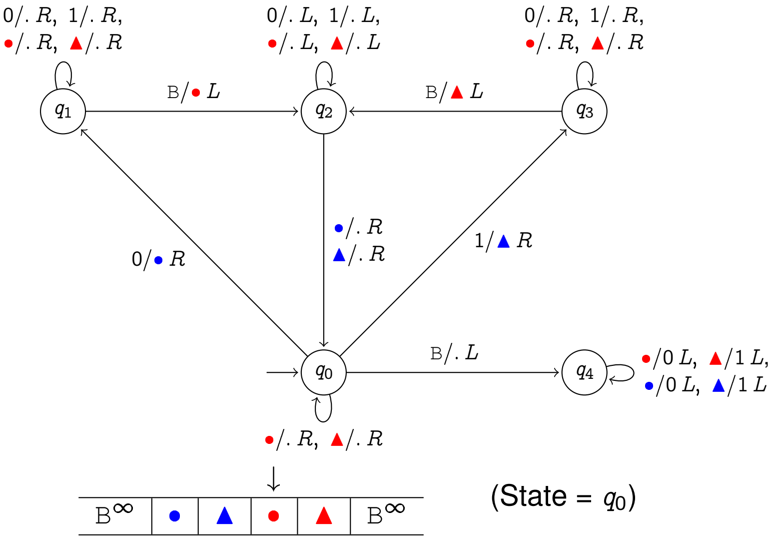




(State = q_2)

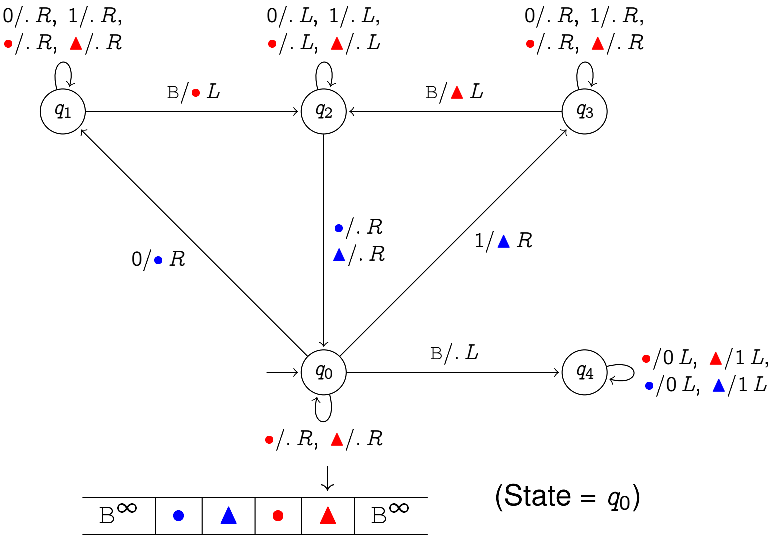
Example 2

A TM that duplicates the input string $w \in \{0, 1\}^*$.



Example 2

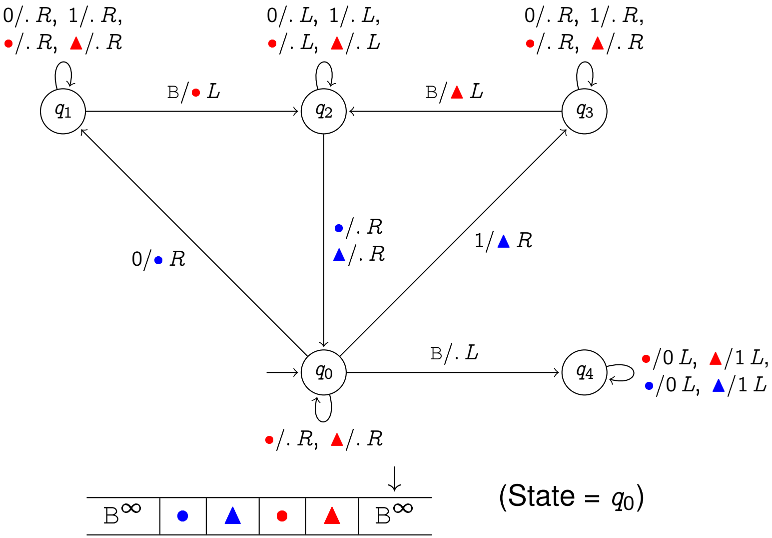
A TM that duplicates the input string $w \in \{0, 1\}^*$.



(State = q_0)

Example 2

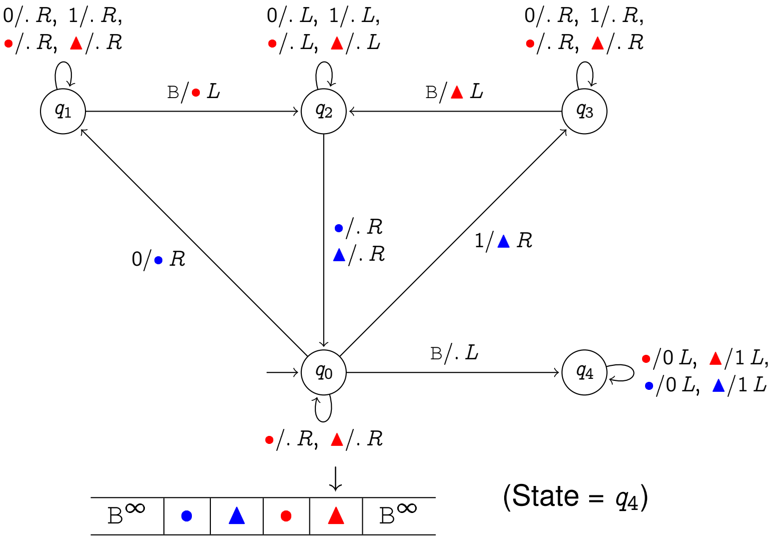
A TM that duplicates the input string $w \in \{0,1\}^*$.



(State = q_0)

Example 2

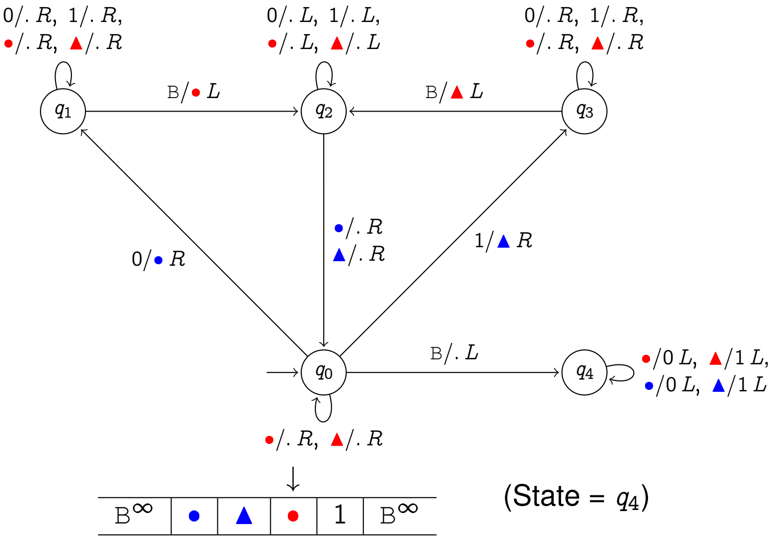
A TM that duplicates the input string $w \in \{0,1\}^*$.



(State = q_4)

Example 2

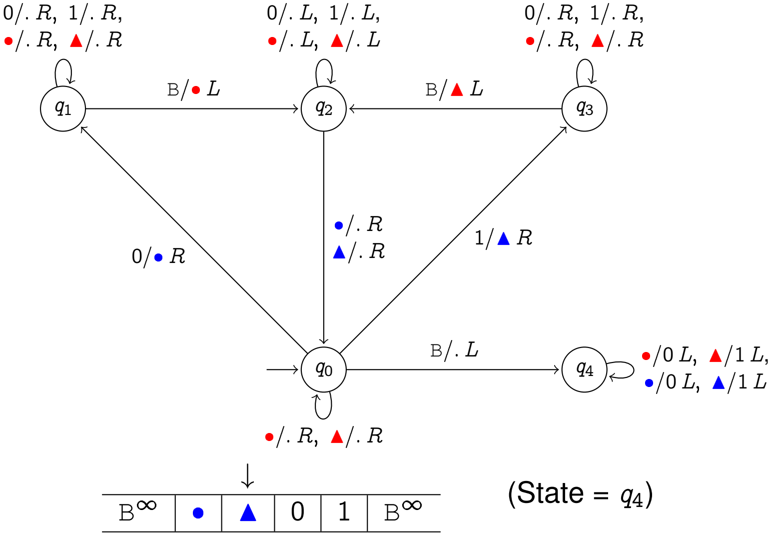
A TM that duplicates the input string $w \in \{0, 1\}^*$.



(State = q_4)

Example 2

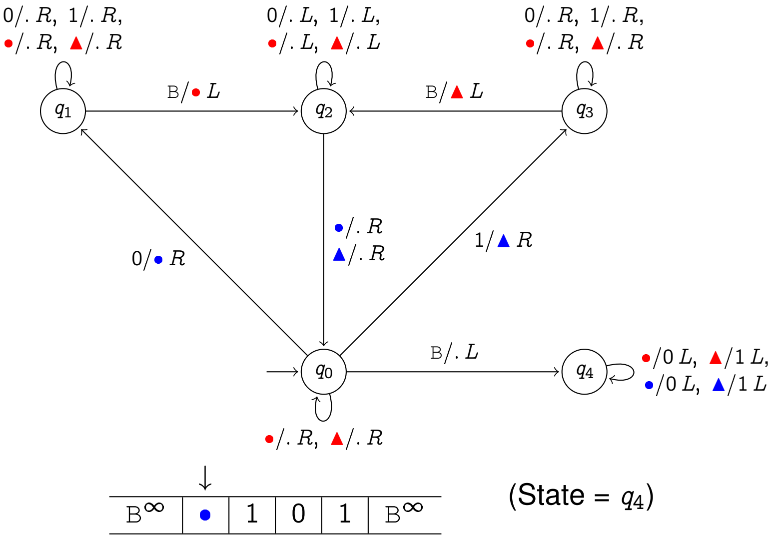
A TM that duplicates the input string $w \in \{0,1\}^*$.



(State = q_4)

Example 2

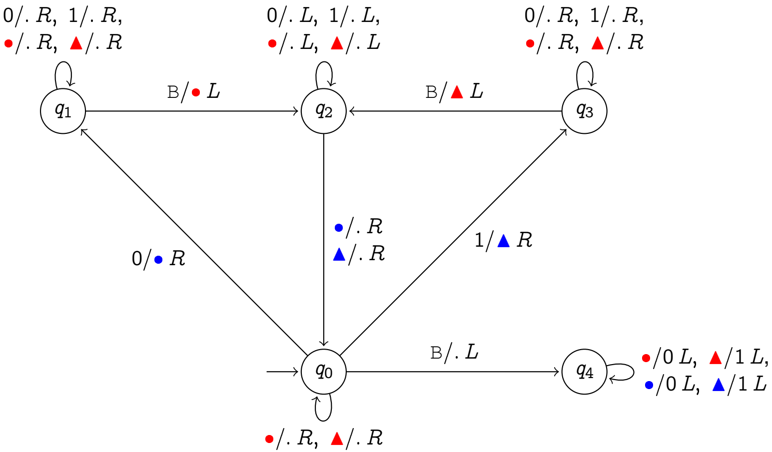
A TM that duplicates the input string $w \in \{0, 1\}^*$.

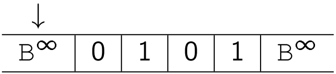


(State = q_4)

Example 2

A TM that duplicates the input string $w \in \{0, 1\}^*$.

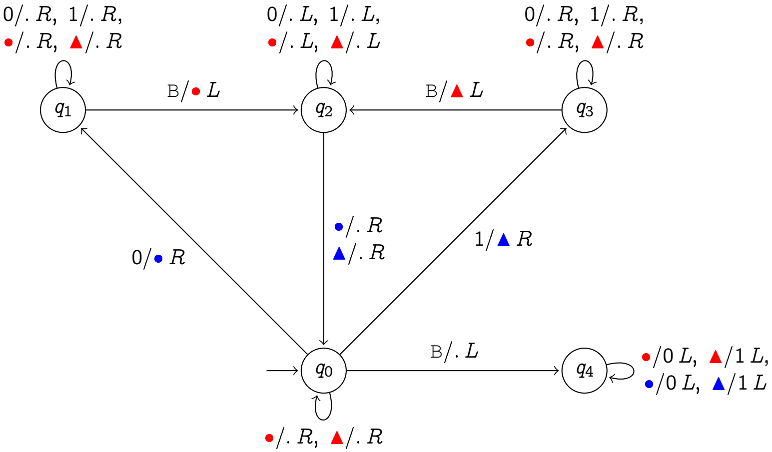




(State = q_4)

Example 2

A TM that duplicates the input string $w \in \{0,1\}^*$.



What does each state/transition represent?

Acceptance

The set $L(M)$ can be defined in two different ways.

1. A TM M **accepts by termination** the language of the input strings w for which it terminates:

$$L(M) = \{w \in \Sigma^* \mid q_0 w \vdash^* \perp\}$$

No need for accepting states.

Acceptance

The set $L(M)$ can be defined in two different ways.

1. A TM M **accepts by termination** the language of the input strings w for which it terminates:

   $$L(M) = \{w \in \Sigma^* \mid q_0 w \vdash^* \perp\}$$

   No need for accepting states.

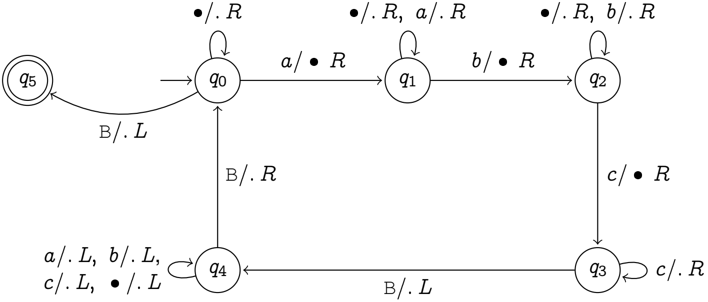
2. $L(M)$ can also be defined by **termination in an accepting state**, extending M with a set $F \subseteq Q$:

   $$L(M) = \{w \in \Sigma^* \mid \exists q_f \in F,\ u, v \in \Gamma^* : q_0 w \vdash^* u\, q_f\, v \vdash \perp\}$$

   This definition can be reduced to the first one by letting $F = Q$. In fact, both definitions are equivalent.

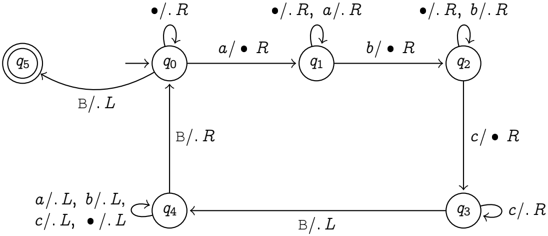
Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$

A TM with accepting state(s):



How does it work?

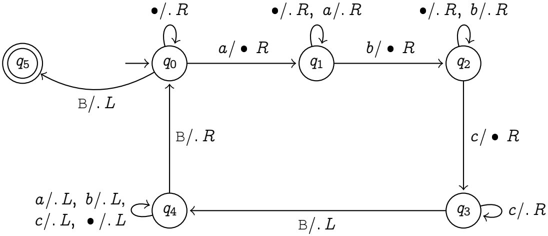
Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$



Computation for input $aabbcc$:

$$\rightarrow\ \mathtt{B}\ [q_0\rangle\ a\ a\ b\ b\ c\ c\ \mathtt{B}$$

Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$

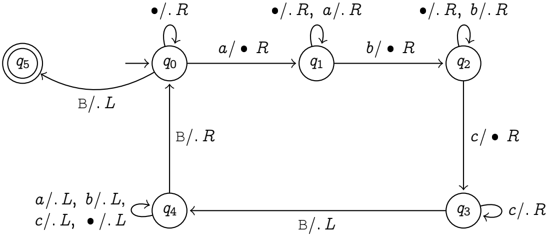


Computation for input $aabbcc$:

$$\rightarrow\ \mathtt{B}\ [q_0\rangle\ a\ a\ b\ b\ c\ c\ \mathtt{B}$$

$$\vdash \mathtt{B}\ \bullet\ [q_1\rangle\ a\ b\ b\ c\ c\ \mathtt{B}$$

Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$



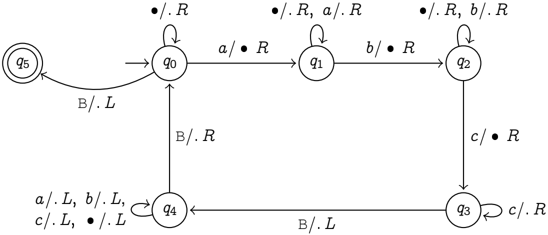
Computation for input $aabbcc$:

$$\rightarrow\ \mathtt{B}\ [q_0\rangle\ a\ a\ b\ b\ c\ c\ \mathtt{B}$$

$$\vdash \mathtt{B}\ \bullet\ [q_1\rangle\ a\ b\ b\ c\ c\ \mathtt{B}$$

$$\vdash \mathtt{B}\ \bullet\ a\ [q_1\rangle\ b\ b\ c\ c\ \mathtt{B}$$

Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$



Computation for input $aabbcc$:

$$\rightarrow \; \mathtt{B}\,[q_0\rangle\; a\; a\; b\; b\; c\; c\; \mathtt{B}$$

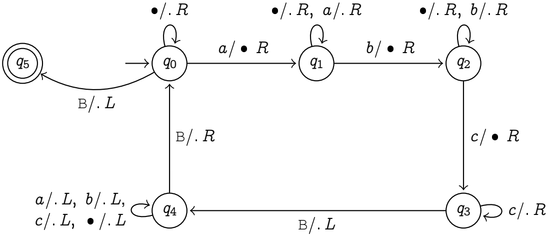
$$\vdash \mathtt{B}\; \bullet\; [q_1\rangle\; a\; b\; b\; c\; c\; \mathtt{B}$$

$$\vdash \mathtt{B}\; \bullet\; a\; [q_1\rangle\; b\; b\; c\; c\; \mathtt{B}$$

$$\vdash \mathtt{B}\; \bullet\; a\; \bullet\; [q_2\rangle\; b\; c\; c\; \mathtt{B}$$

$$\vdash \mathtt{B}\; \bullet\; a\; \bullet\; b\; [q_2\rangle\; c\; c\; \mathtt{B}$$

Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$



Computation for input $aabbcc$:

$\rightarrow\ \mathrm{B}\ [q_0\rangle\ a\ a\ b\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ [q_1\rangle\ a\ b\ b\ c\ c\ \mathrm{B}$

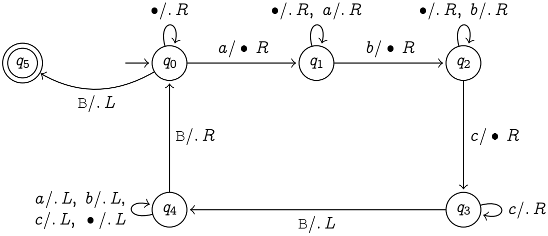
$\vdash \mathrm{B}\ \bullet\ a\ [q_1\rangle\ b\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ [q_2\rangle\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ [q_2\rangle\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ [q_3\rangle\ c\ \mathrm{B}$

Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$



Computation for input $aabbcc$:

$\rightarrow\ \mathrm{B}\ [q_0\rangle\ a\ a\ b\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ [q_1\rangle\ a\ b\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ [q_1\rangle\ b\ b\ c\ c\ \mathrm{B}$

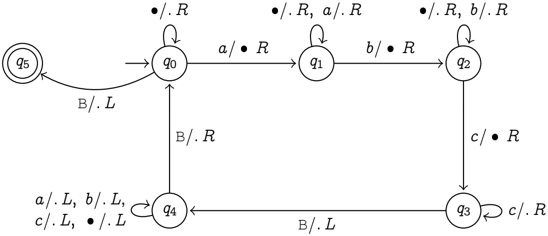
$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ [q_2\rangle\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ [q_2\rangle\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ [q_3\rangle\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ c\ [q_3\rangle\ \mathrm{B}$

Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$



Computation for input $aabbcc$:

$\rightarrow\ \mathrm{B}\ [q_0\rangle\ a\ a\ b\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ [q_1\rangle\ a\ b\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ [q_1\rangle\ b\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ [q_2\rangle\ b\ c\ c\ \mathrm{B}$

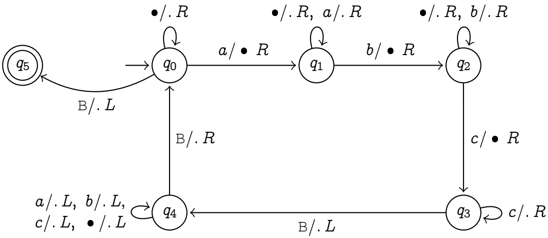
$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ [q_2\rangle\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ [q_3\rangle\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ c\ [q_3\rangle\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ [q_4\rangle\ c\ \mathrm{B}$

Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$



Computation for input $aabbcc$:

$$\rightarrow\ \mathrm{B}\ [q_0\rangle\ a\ a\ b\ b\ c\ c\ \mathrm{B}$$

$$\vdash \mathrm{B}\ \bullet\ [q_1\rangle\ a\ b\ b\ c\ c\ \mathrm{B}$$

$$\vdash \mathrm{B}\ \bullet\ a\ [q_1\rangle\ b\ b\ c\ c\ \mathrm{B}$$

$$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ [q_2\rangle\ b\ c\ c\ \mathrm{B}$$

$$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ [q_2\rangle\ c\ c\ \mathrm{B}$$

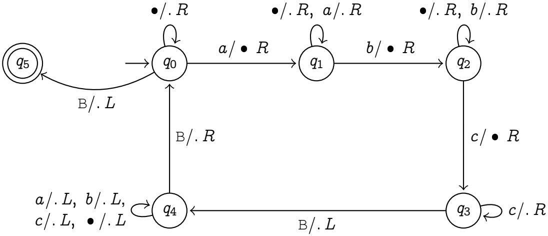
$$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ [q_3\rangle\ c\ \mathrm{B}$$

$$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ c\ [q_3\rangle\ \mathrm{B}$$

$$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ [q_4\rangle\ c\ \mathrm{B}$$

$$\vdash^*\ [q_4\rangle\ \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ c\ \mathrm{B}$$

Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$



Computation for input *aabbcc*:

$\rightarrow\ \mathrm{B}\ [q_0\rangle\ a\ a\ b\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ [q_1\rangle\ a\ b\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ [q_1\rangle\ b\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ [q_2\rangle\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ [q_2\rangle\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ [q_3\rangle\ c\ \mathrm{B}$

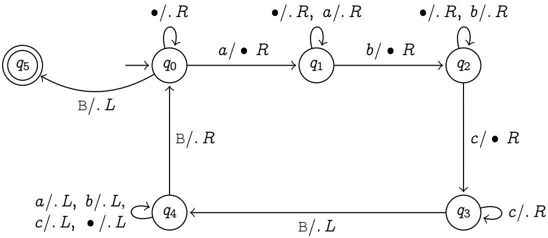
$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ c\ [q_3\rangle\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ [q_4\rangle\ c\ \mathrm{B}$

$\vdash^*\ [q_4\rangle\ \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ c\ \mathrm{B}$

$\vdash^*\ \mathrm{B}\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ [q_3\rangle\ \mathrm{B}$

Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$

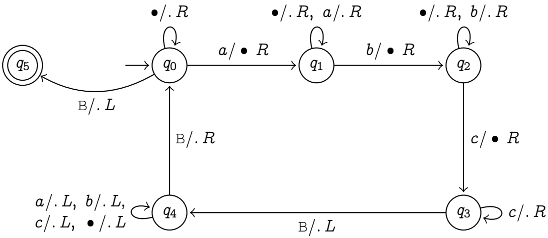


Computation for input $aabbcc$:

$$\to\ \mathrm{B}\ [q_0\rangle\ a\ a\ b\ b\ c\ c\ \mathrm{B}$$
$$\vdash \mathrm{B}\ \bullet\ [q_1\rangle\ a\ b\ b\ c\ c\ \mathrm{B}$$
$$\vdash \mathrm{B}\ \bullet\ a\ [q_1\rangle\ b\ b\ c\ c\ \mathrm{B}$$
$$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ [q_2\rangle\ b\ c\ c\ \mathrm{B}$$
$$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ [q_2\rangle\ c\ c\ \mathrm{B}$$
$$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ [q_3\rangle\ c\ \mathrm{B}$$
$$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ c\ [q_3\rangle\ \mathrm{B}$$

$$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ [q_4\rangle\ c\ \mathrm{B}$$
$$\vdash^*\ [q_4\rangle\ \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ c\ \mathrm{B}$$
$$\vdash^*\ \mathrm{B}\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ [q_3\rangle\ \mathrm{B}$$
$$\vdash^*\ [q_4\rangle\ \mathrm{B}\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \mathrm{B}$$

Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$



Computation for input $aabbcc$:

$\rightarrow\ \mathrm{B}\,[q_0\rangle\ a\ a\ b\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ [q_1\rangle\ a\ b\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ [q_1\rangle\ b\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ [q_2\rangle\ b\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ [q_2\rangle\ c\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ [q_3\rangle\ c\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ c\ [q_3\rangle\ \mathrm{B}$

$\vdash \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ [q_4\rangle\ c\ \mathrm{B}$

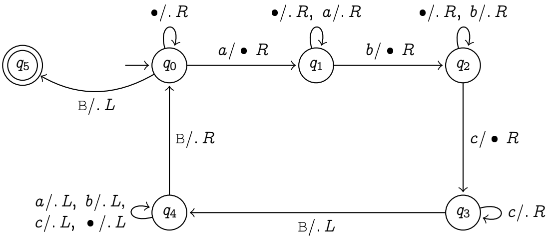
$\vdash^*\ [q_4\rangle\ \mathrm{B}\ \bullet\ a\ \bullet\ b\ \bullet\ c\ \mathrm{B}$

$\vdash^*\ \mathrm{B}\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ [q_3\rangle\ \mathrm{B}$

$\vdash^*\ [q_4\rangle\ \mathrm{B}\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \mathrm{B}$

$\vdash\ \mathrm{B}[q_0\rangle\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \mathrm{B}$

Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$



Computation for input $aabbcc$:

$\rightarrow\ B\,[q_0\rangle\ a\ a\ b\ b\ c\ c\ B$

$\vdash B\ \bullet\ [q_1\rangle\ a\ b\ b\ c\ c\ B$

$\vdash B\ \bullet\ a\ [q_1\rangle\ b\ b\ c\ c\ B$

$\vdash B\ \bullet\ a\ \bullet\ [q_2\rangle\ b\ c\ c\ B$

$\vdash B\ \bullet\ a\ \bullet\ b\ [q_2\rangle\ c\ c\ B$

$\vdash B\ \bullet\ a\ \bullet\ b\ \bullet\ [q_3\rangle\ c\ B$

$\vdash B\ \bullet\ a\ \bullet\ b\ \bullet\ c\ [q_3\rangle\ B$

$\vdash B\ \bullet\ a\ \bullet\ b\ \bullet\ [q_4\rangle\ c\ B$

$\vdash^*\ [q_4\rangle\ B\ \bullet\ a\ \bullet\ b\ \bullet\ c\ B$

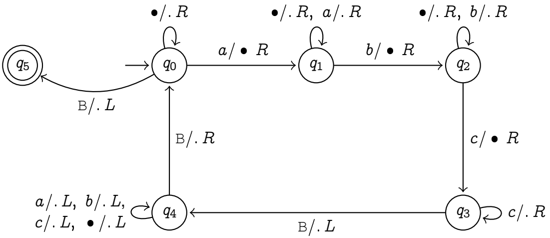
$\vdash^*\ B\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ [q_3\rangle\ B$

$\vdash^*\ [q_4\rangle\ B\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ B$

$\vdash\ B[q_0\rangle\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ B$

$\vdash^*\ B\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ [q_0\rangle\ B$

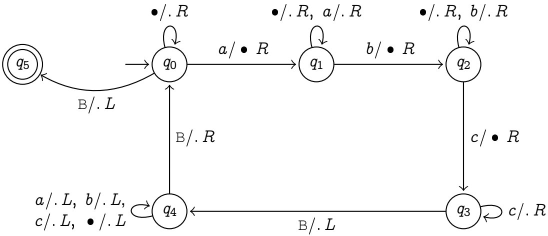
Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$



Computation for input $aabbcc$:

$$\begin{array}{ll}
\to \mathrm{B}\ [q_0\rangle\ a\ a\ b\ b\ c\ c\ \mathrm{B} & \vdash \mathrm{B} \bullet a \bullet b \bullet [q_4\rangle\ c\ \mathrm{B} \\
\vdash \mathrm{B} \bullet [q_1\rangle\ a\ b\ b\ c\ c\ \mathrm{B} & \vdash^* [q_4\rangle\ \mathrm{B} \bullet a \bullet b \bullet c\ \mathrm{B} \\
\vdash \mathrm{B} \bullet a\ [q_1\rangle\ b\ b\ c\ c\ \mathrm{B} & \vdash^* \mathrm{B} \bullet \bullet \bullet \bullet \bullet \bullet [q_3\rangle\ \mathrm{B} \\
\vdash \mathrm{B} \bullet a \bullet [q_2\rangle\ b\ c\ c\ \mathrm{B} & \vdash^* [q_4\rangle\ \mathrm{B} \bullet \bullet \bullet \bullet \bullet \bullet \mathrm{B} \\
\vdash \mathrm{B} \bullet a \bullet b\ [q_2\rangle\ c\ c\ \mathrm{B} & \vdash \mathrm{B}[q_0\rangle \bullet \bullet \bullet \bullet \bullet \bullet \mathrm{B} \\
\vdash \mathrm{B} \bullet a \bullet b \bullet [q_3\rangle\ c\ \mathrm{B} & \vdash^* \mathrm{B} \bullet \bullet \bullet \bullet \bullet \bullet [q_0\rangle\ \mathrm{B} \\
\vdash \mathrm{B} \bullet a \bullet b \bullet c\ [q_3\rangle\ \mathrm{B} & \vdash \mathrm{B} \bullet \bullet \bullet \bullet \bullet [q_5\rangle \bullet \mathrm{B}
\end{array}$$

Example 5.1.2: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$



Consider now the computation for $aabcc$: where does it get stuck?

Further Terminology

A TM is **always terminating** if it terminates for every input.

Let L be a language.

- L is **semi-decidable** (or **recursively enumerable, RE**) if there exists a TM M such that $L = L(M)$.
- L is **decidable** (or **recursive**) if there is an always terminating TM that accepts L by termination in an accepting state.
- If L is decidable, then it is also semi-decidable. The converse doesn't hold!

Taking Stock

This lecture (Sections 5.1 and 5.2):

- Turing machines
- Key terminology for TM-accepted languages

Next Lecture (Sections 5.3–5.8)

- Further examples of TMs
- Variants of TMs: multiple-track, multiple-tape, non-deterministic