Turing Machine

Turing machine

Finite automata (regular languages):
limited memory

Turing machine

Finite automata (regular languages):
limited memory

Push-down automata (context-free languages):
unlimited memory, but restricted access

Turing machine

Finite automata (regular languages):
limited memory

Push-down automata (context-free languages):
unlimited memory, but restricted access

How about unlimited memory, *and* unrestricted access

Turing machine

Finite automata (regular languages):
limited memory

Push-down automata (context-free languages):
unlimited memory, but restricted access

How about unlimited memory, *and* unrestricted access

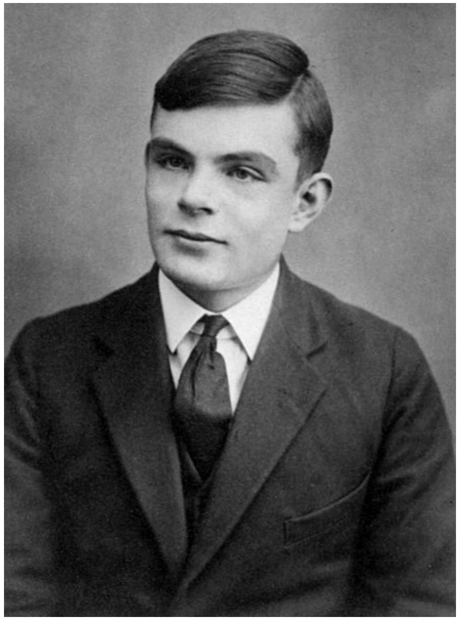
This is the model of Turing machines:
by Alan Turing (1936)

A Turing machine can do everything a real computer can do

Turing machine

However, (we'll see) even a Turing machine cannot solve certain problems

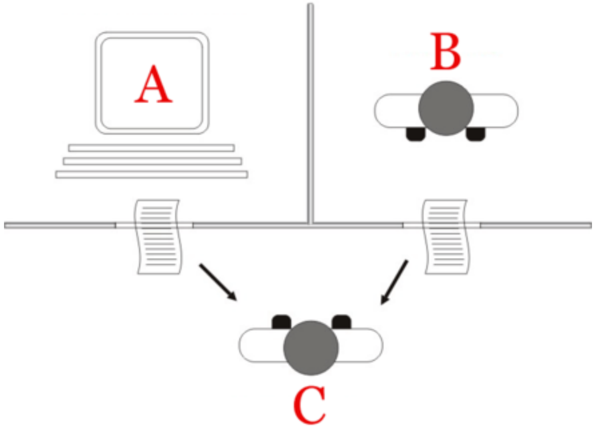
Alan Turing (1912~1954)



Turing machine: formalization of *algorithm* and *computation*

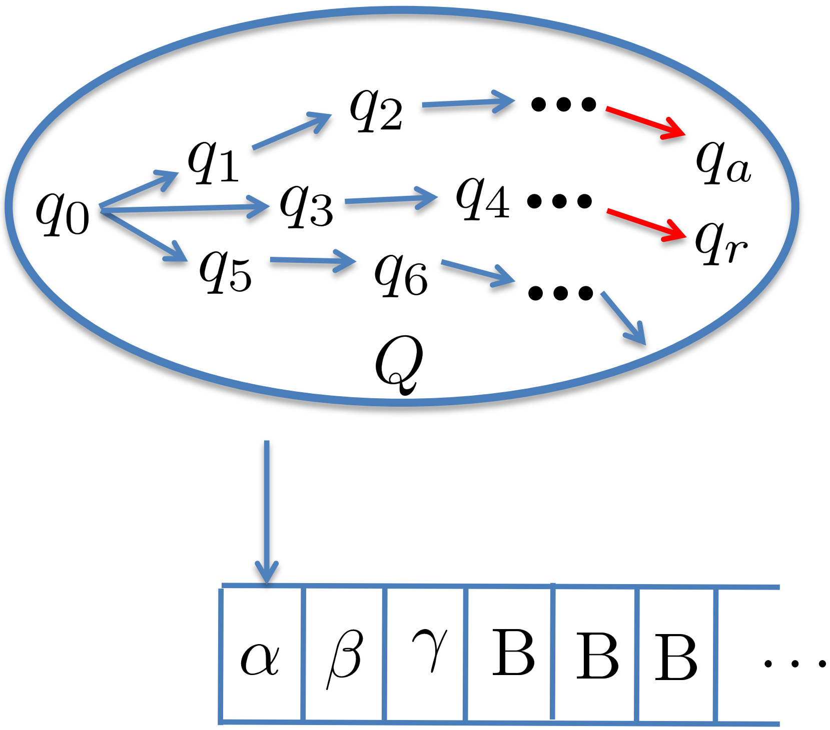
Decode secret messages encoded by Enigma machine (movie: The Imitation Game (2014))

Turing test:

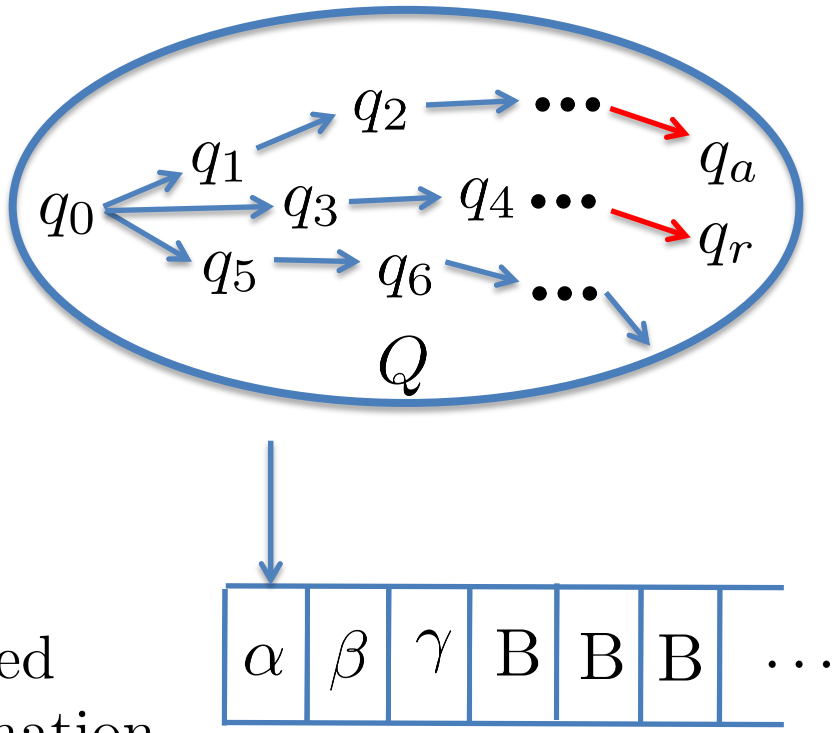


Turing award: Nobel prize of computing

Turing machine

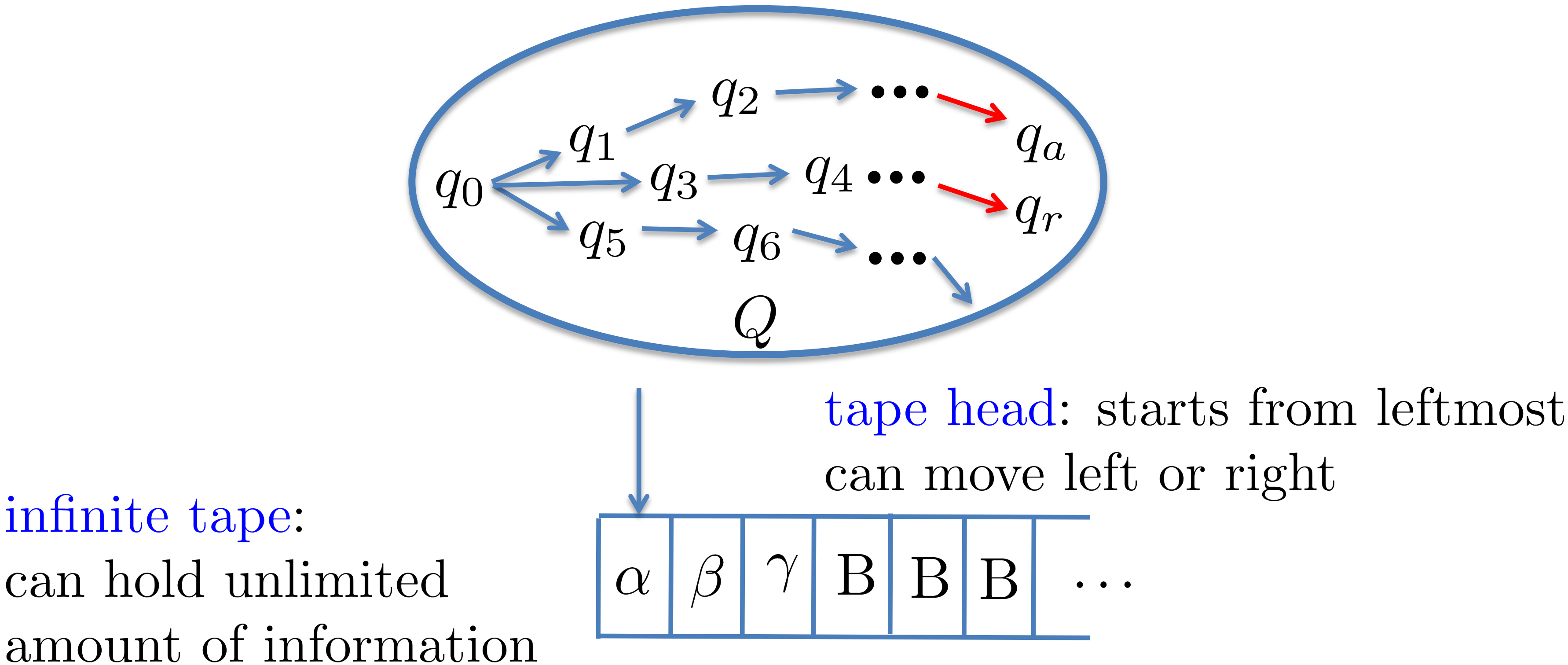


Turing machine

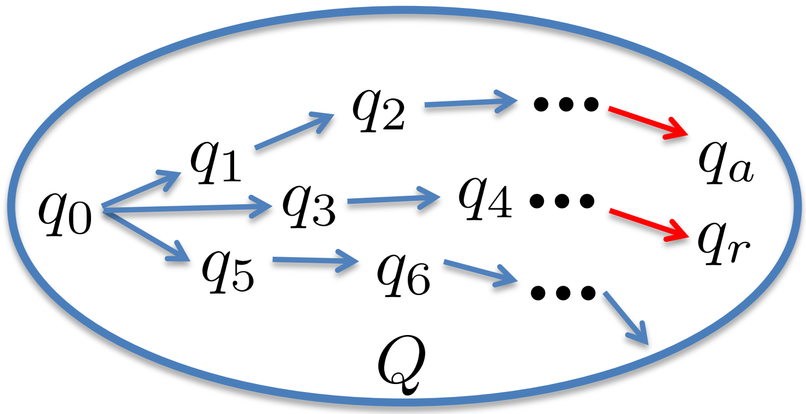


infinite tape:
can hold unlimited
amount of information

Turing machine

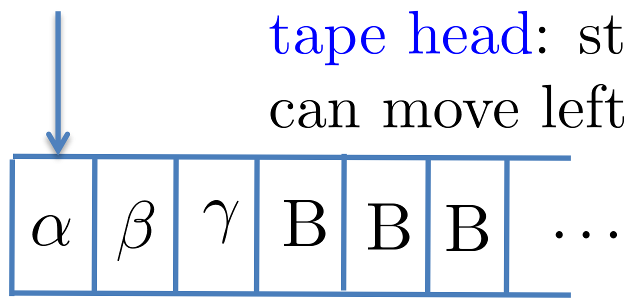


Turing machine



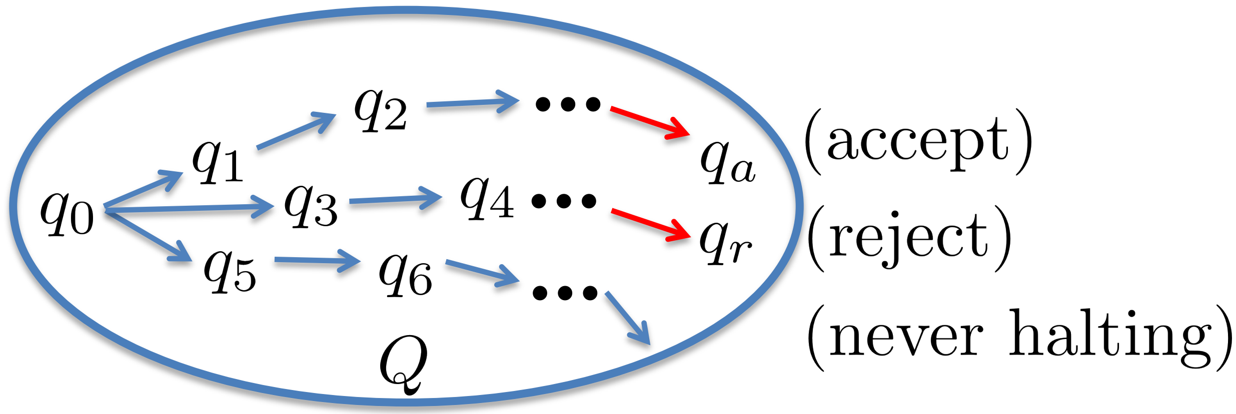
tape head: starts from leftmost;
can move left or right

infinite tape:
can hold unlimited
amount of information



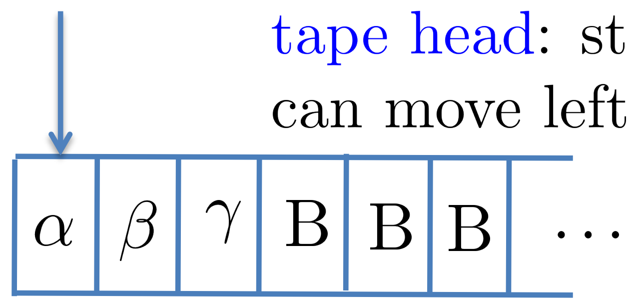
input string stored from leftmost;
followed by B (blank symbol)

Turing machine



tape head: starts from leftmost;
can move left or right

infinite tape:
can hold unlimited
amount of information



input string stored from leftmost;
followed by B (blank symbol)

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$0\ 1\ 1\ 0\ 0\ 0\ \#\ 0\ 1\ 1\ 0\ 0\ 0\ \mathrm{B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\downarrow$$

$$\text{x } 1\ 1\ 0\ 0\ 0\ \#\ 0\ 1\ 1\ 0\ 0\ 0\ \text{B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x } 1\ 1\ 0\ 0\ 0\ \#\ 0\ 1\ 1\ 0\ 0\ 0\ \text{B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\downarrow$$

$$\text{x } 1\ 1\ 0\ 0\ 0\ \#\ 0\ 1\ 1\ 0\ 0\ 0\ \text{B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

x 1 1 0 0 0 # x 1 1 0 0 0 B $\cdots$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

↓

x 1 1 0 0 0 # x 1 1 0 0 0 B $\cdots$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\downarrow$$

$$\text{x}\ 1\ 1\ 0\ 0\ 0\ \#\ \text{x}\ 1\ 1\ 0\ 0\ 0\ \text{B}\ \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x } \overset{\downarrow}{1} \text{ 0 0 0 \# x 1 1 0 0 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x 1 0 0 0 \# x 1 1 0 0 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x 1 0 0 0 \# x x 1 0 0 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

↓

x x 1 0 0 0 # x x 1 0 0 0 B $\cdots$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x 1 0 0 0 \# x x 1 0 0 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

x x x 0 0 0 # x x 1 0 0 0 B $\cdots$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x x 0 0 0 \# x x 1 0 0 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x x 0 0 0 \# x x x 0 0 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\downarrow$$

$$\text{x x x 0 0 0 \# x x x 0 0 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\downarrow$$

$$\text{x x x 0 0 0 \# x x x 0 0 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

↓

x x x x 0 0 # x x x 0 0 0 B ⋯

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x x x 0 0 \# x x x 0 0 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x x x 0 0 \# x x x x 0 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\downarrow$$

$$\text{x x x x 0 0 \# x x x x 0 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x x x } \downarrow 0 \text{ } 0 \ \# \text{ x x x x } 0 \ 0 \text{ B } \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x x x x } \underset{\uparrow}{0} \text{ } \# \text{ x x x x 0 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x x x x 0 \# x x x x 0 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

x x x x x 0 # x x x x x 0 B $\cdots$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x x x x 0 \# x x x x x 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\downarrow$$

$$\mathrm{x\ x\ x\ x\ x\ 0\ \#\ x\ x\ x\ x\ x\ 0\ B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x x x x x } \# \text{ x x x x x 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x x x x x \# x x x x x 0 B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

x x x x x x # x x x x x x B $\cdots$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\downarrow$$

$$\text{x x x x x x \# x x x x x x B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x x x x x \# x x x x x x B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

e.g. $s = 011000\#011000$

$$\text{x x x x x x \# x x x x x x B} \cdots$$

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

M_1 = "On input string s:

1) Zig-zag across tape to corresponding positions on both sides of #, and check if these positions contain the same symbols. If they do not, or if no # found, then reject. Cross off symbols that are checked.

2) When all symbols to the left of # are crossed off, check for any remaining symbols to the right of #. If any symbols remain, then reject. Otherwise, accept."

Example

Consider $L = \{w\#w \mid w \in \{0,1\}^*\}$ (not context-free)

Design a Turing machine to recognize L:
given any string s, accept s iff $s \in L$; reject s iff $s \notin L$

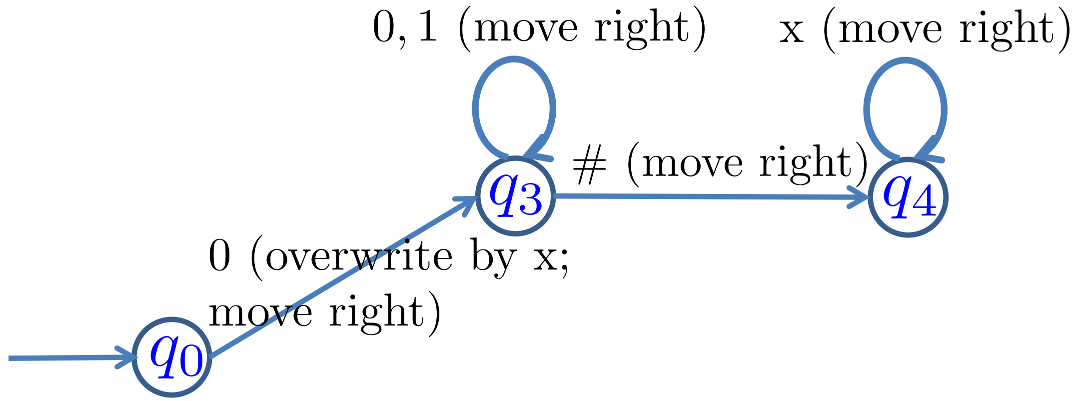
M_1 = "On input string s:

1) Zig-zag across tape to corresponding positions on both sides of #, and check if these positions contain the same symbols. If they do not, or if no # found, then reject. Cross off symbols that are checked.

2) When all symbols to the left of # are crossed off, check for any remaining symbols to the right of #. If any symbols remain, then reject. Otherwise, accept."

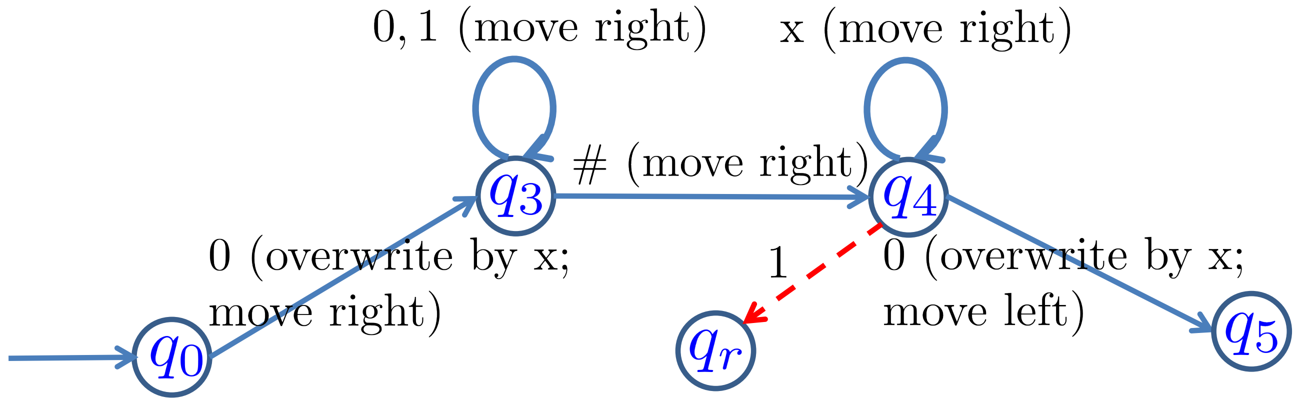
Example

e.g. $s = 011000\#011000$



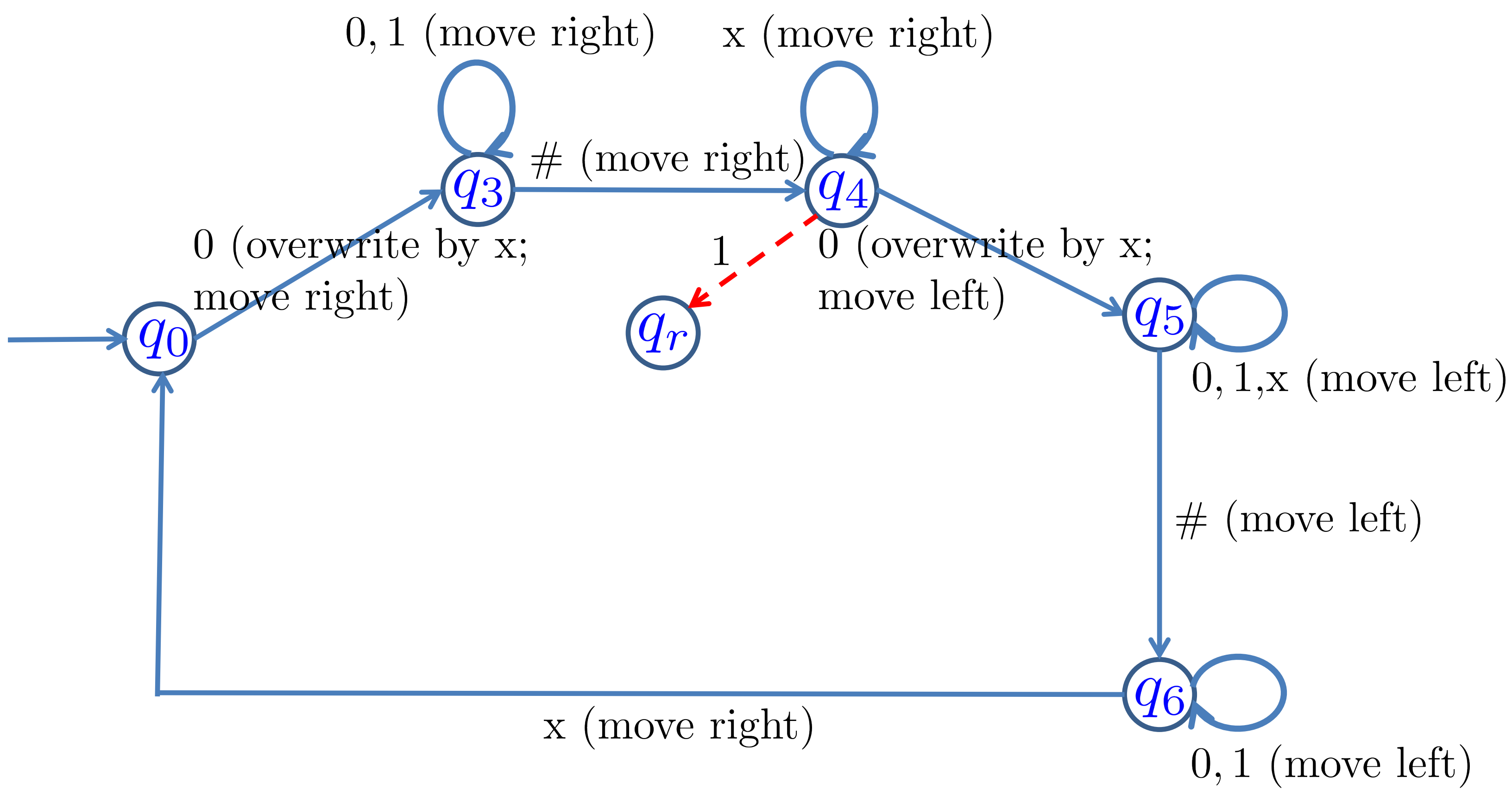
Example

e.g. $s = 011000\#011000$



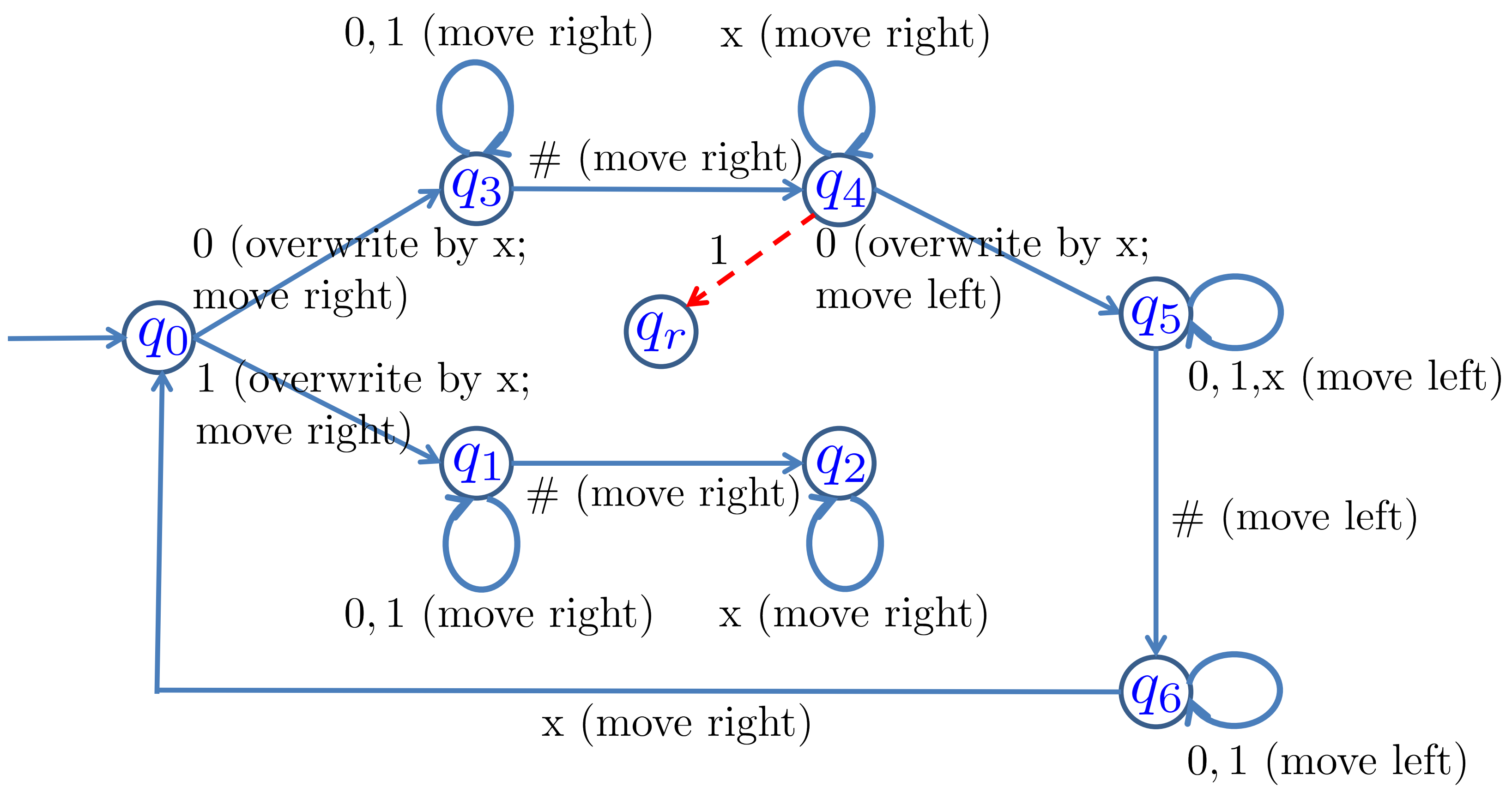
Example

e.g. $s = \cancel{0}11000\#011000$



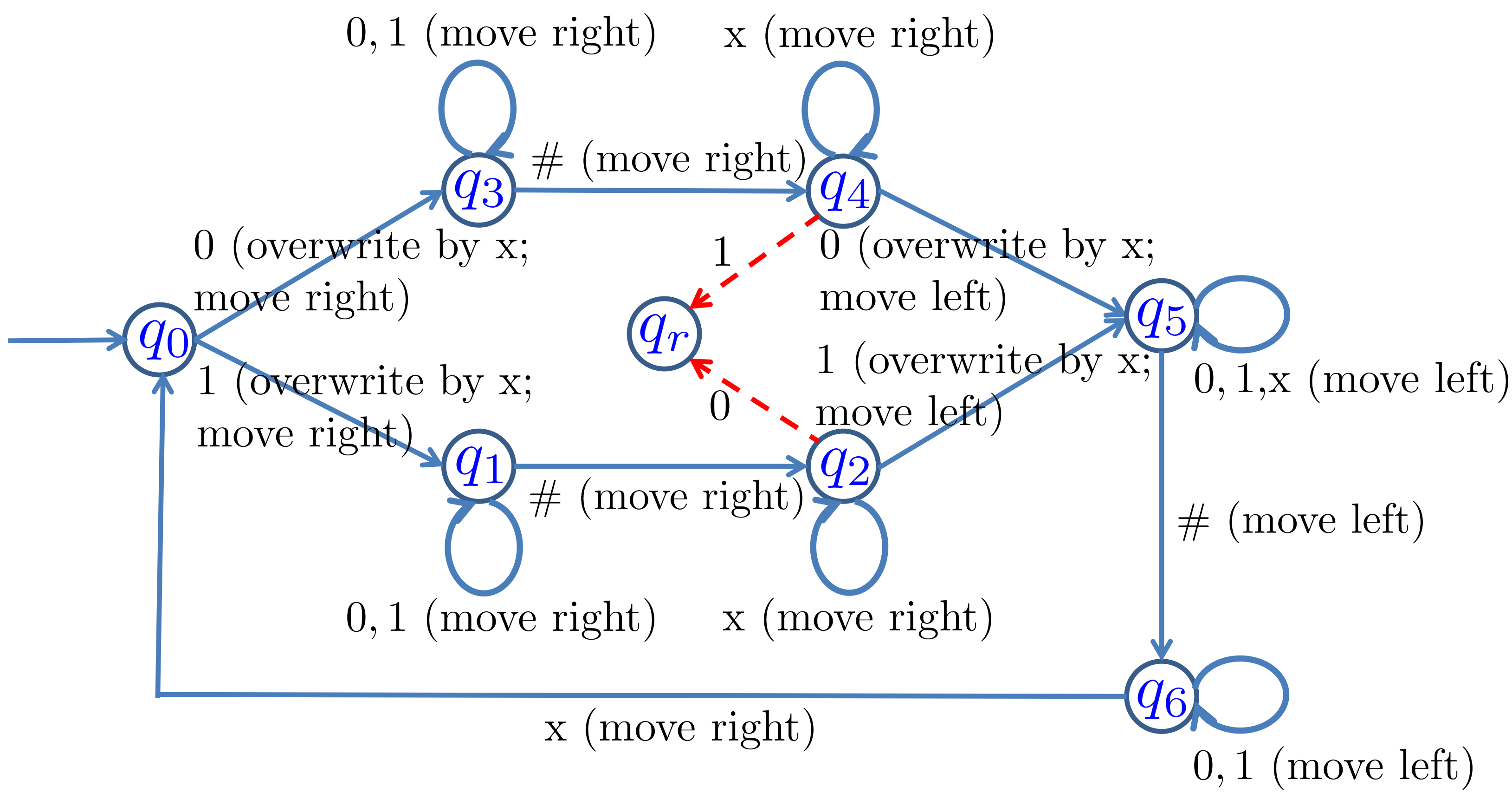
Example

e.g. $s =$ 011000#011000



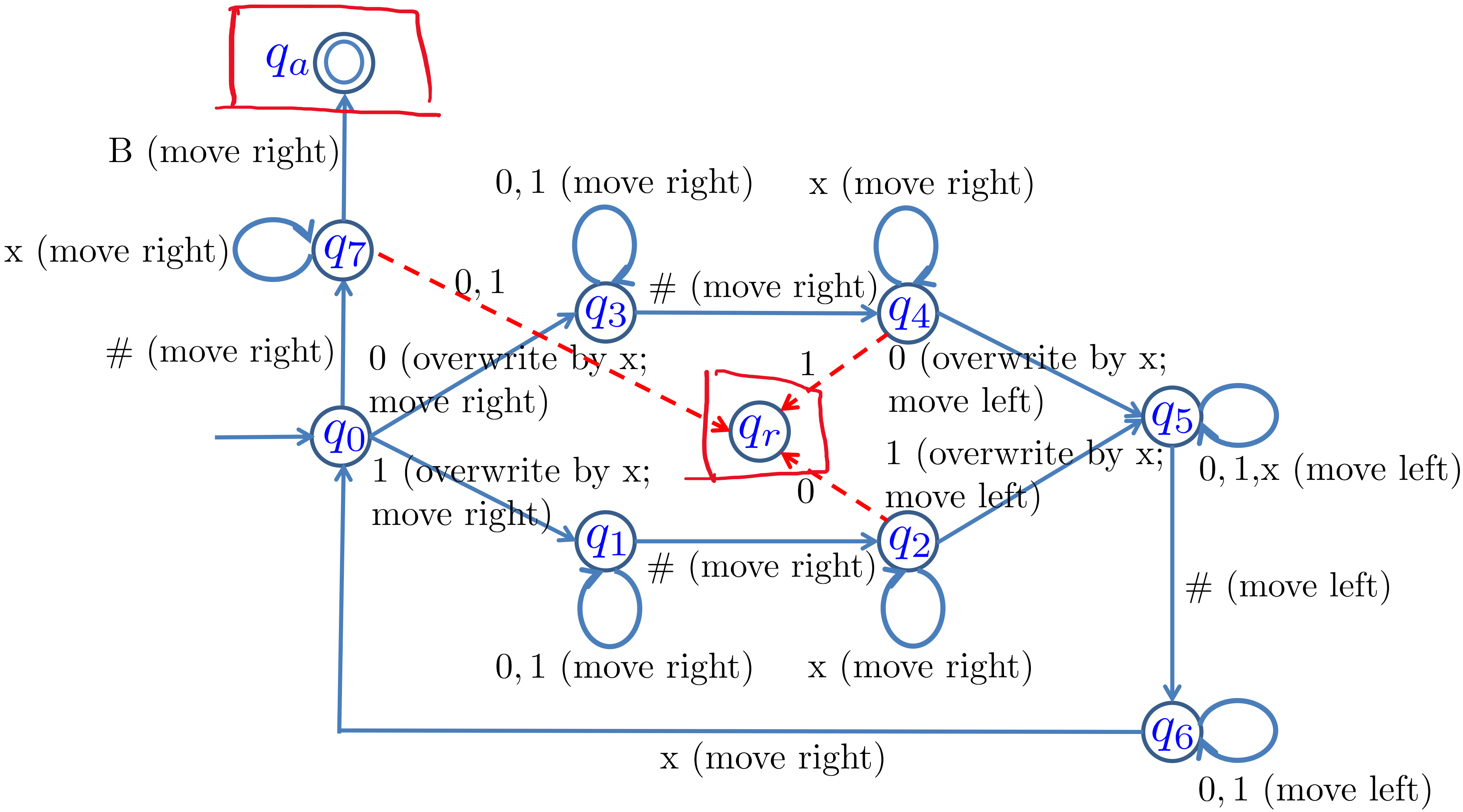
Example

e.g. $s = 011000\#011000$



Example

e.g. $s = 011000\#011000$

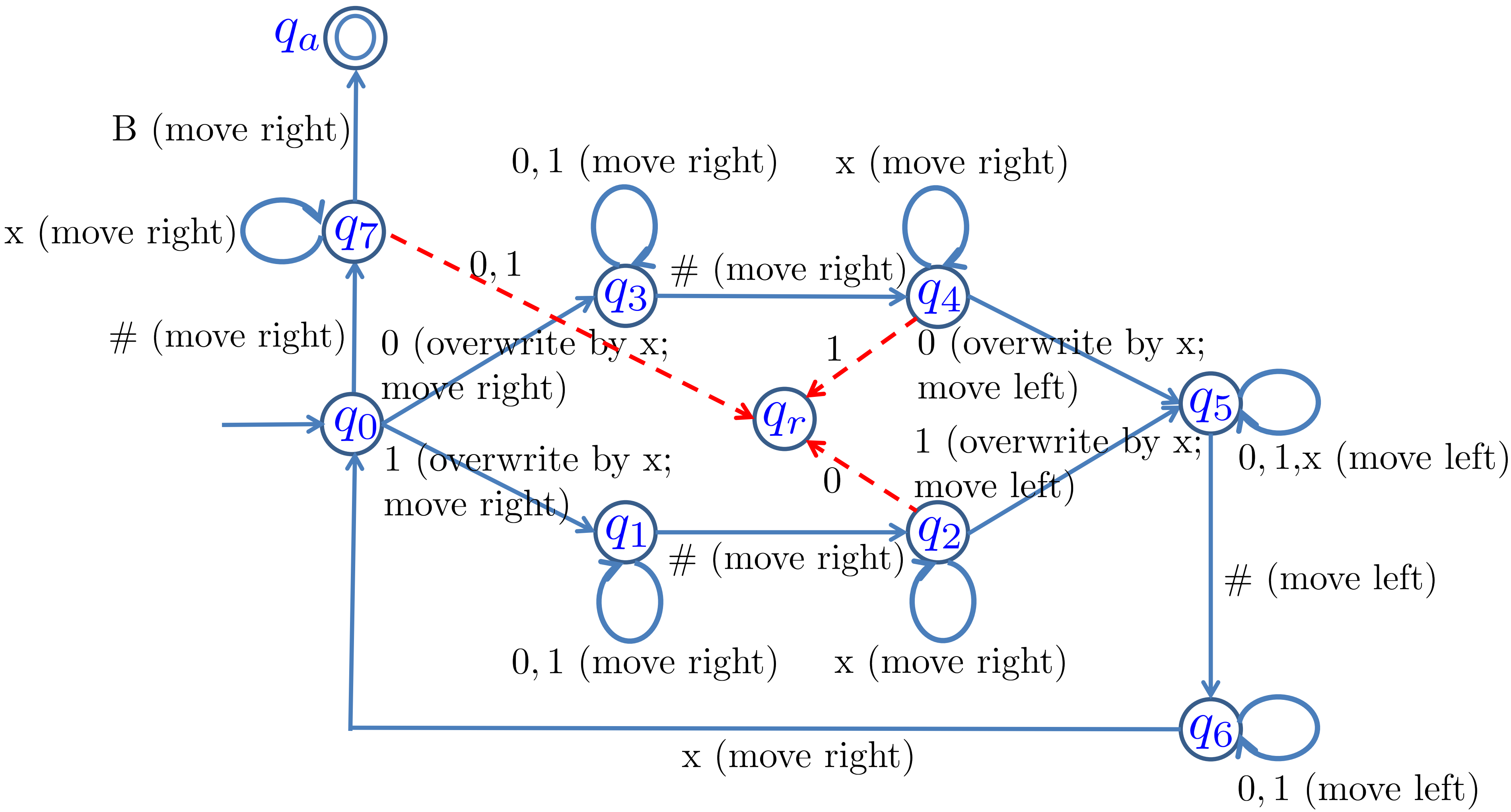


Example

e.g. $s = 0010\#0110$

$q_0\ q_3\ q_4\ q_5\ q_6\ q_0\ q_3\ q_4\ q_r$

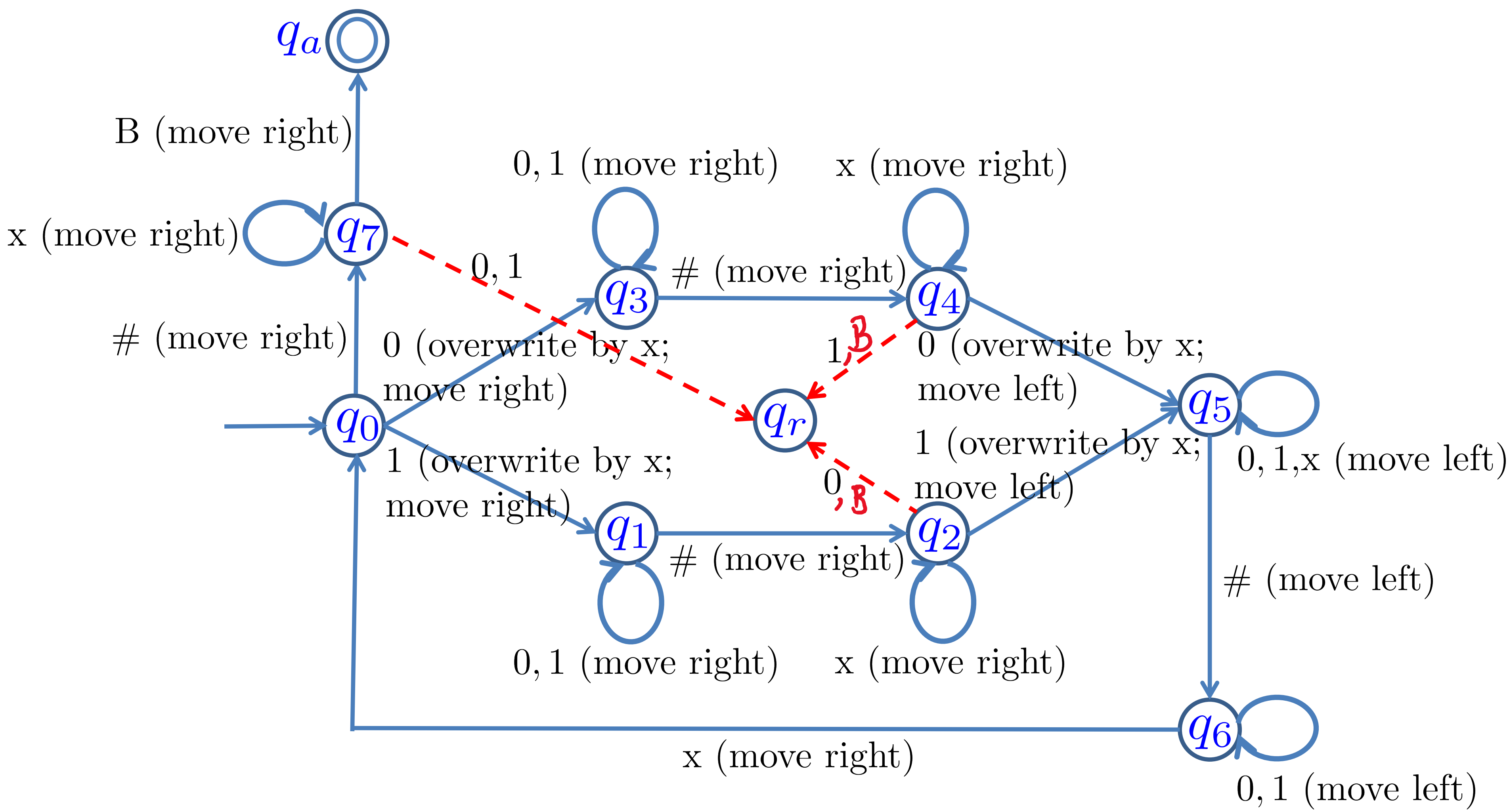
reject ✓



Example

e.g. $s = 010\#01$

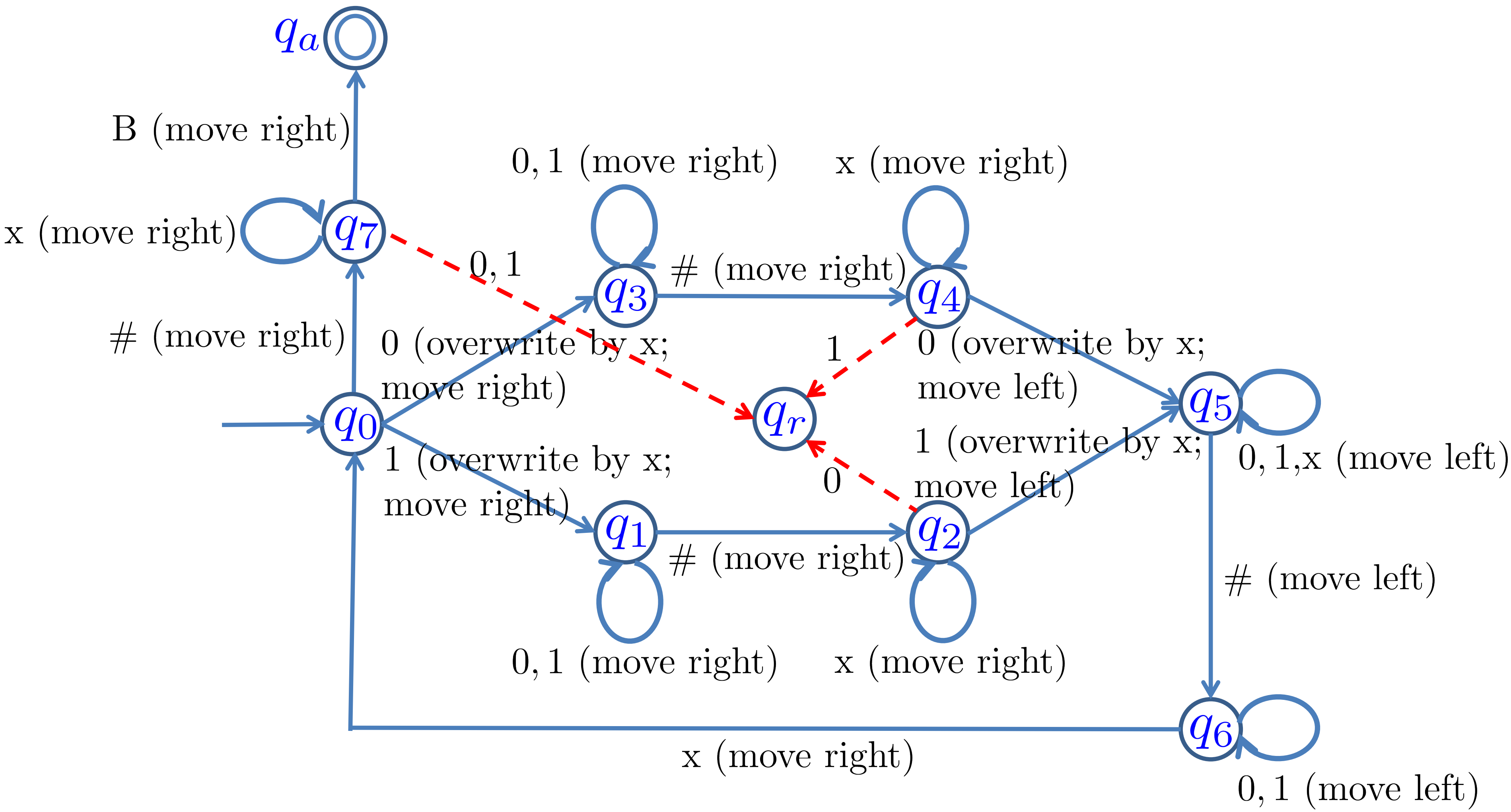
$q_0\ q_3\ q_4\ q_5\ q_6\ q_0\ q_1\ q_2\ q_5\ q_6\ q_0$
$q_3\ q_4\ q_r$ = "reject"



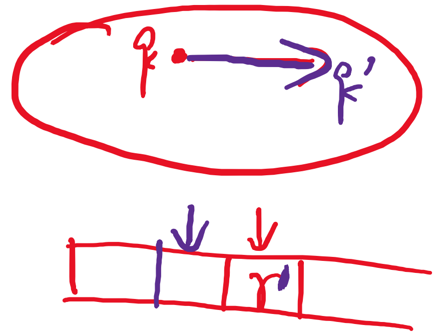
Example

$q_0 q_3 q_4 q_5 q_6 q_0 q_1 q_2 q_5 q_6 q_0 q_3 q_4 q_5 q_6$
$q_0 q_7 \underline{q_r}$ "reject"

e.g. $s = 010\#0101$



$\delta(q_k, r) = (q_k', r', L)$



Turing machine

A (deterministic) Turing machine (TM) **M** is a 7-tuple $\mathbf{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$, where

Q : state set; a finite set of states

Σ : alphabet; a finite set of symbols; B$\notin \Sigma$

Γ : tape alphabet; a finite set of symbols used in the tape; B$\in \Gamma$ and $\Sigma \subseteq \Gamma$

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

$q_0 \in Q$: initial state

$q_a \in Q$: accept state

$q_r \in Q$: reject state $(q_r \neq q_a)$

Example

$Q = \{q_0, \dots, q_7, q_a, q_r\}$

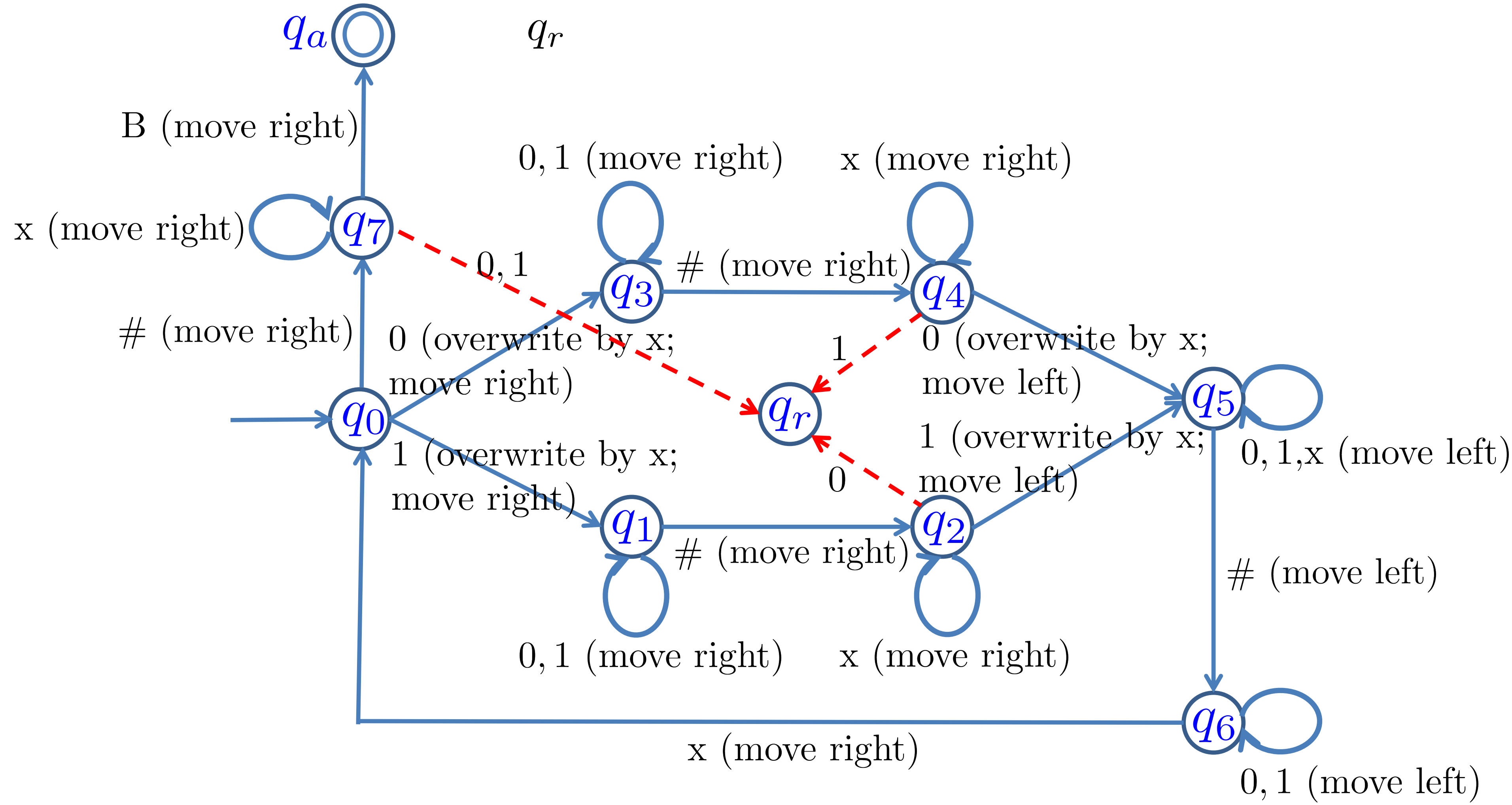
$\Sigma = \{0, 1, \#\}$

$\Gamma = \{B, 0, 1, \#, x\}$

q_0

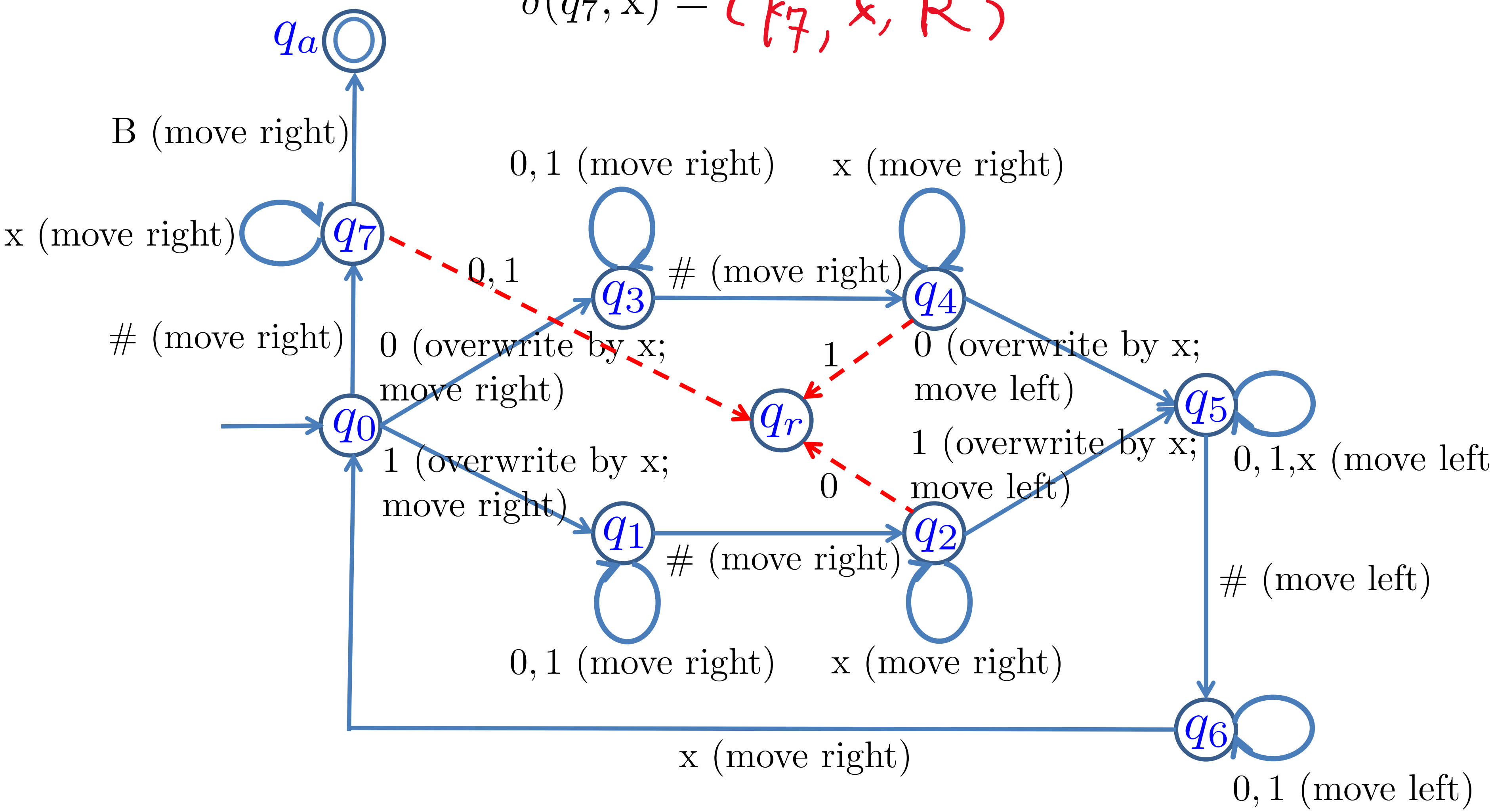
q_a

q_r



Example

$\delta(q_0, 0) = (q_3, \text{x}, \text{R})$
$\delta(q_5, \#) = (q_6, \#, L)$
$\delta(q_7, \text{x}) = (q_7, x, R)$



Turing machine

A (deterministic) Turing machine (TM) $\mathbf{M}$
is a 7-tuple $\mathbf{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

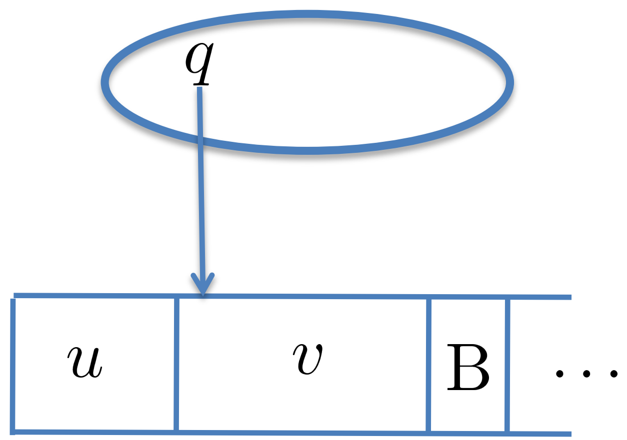
$\delta : Q \times \Gamma \to Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}\}$: transition function

$\delta(q, \gamma) = (q', \gamma', \mathrm{R})$:
before transition: state q, tape head γ
after transition: state q', overwrite by γ', head moves right

Turing machine

A (deterministic) Turing machine (TM) $\mathbf{M}$
is a 7-tuple $\mathbf{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

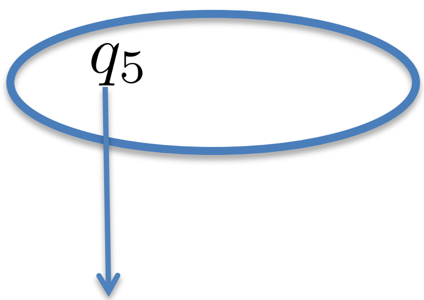
Let $q \in Q$ and $u, v \in \Gamma^*$.
Then uqv is a configuration:
state q, tape content uv, head at first symbol of v



Turing machine

A (deterministic) Turing machine (TM) $\mathbf{M}$
is a 7-tuple $\mathbf{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

Let $q \in Q$ and $u, v \in \Gamma^*$.
Then uqv is a configuration:
state q, tape content uv, head at first symbol of v



e.g. $0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ \mathrm{B} \cdots$

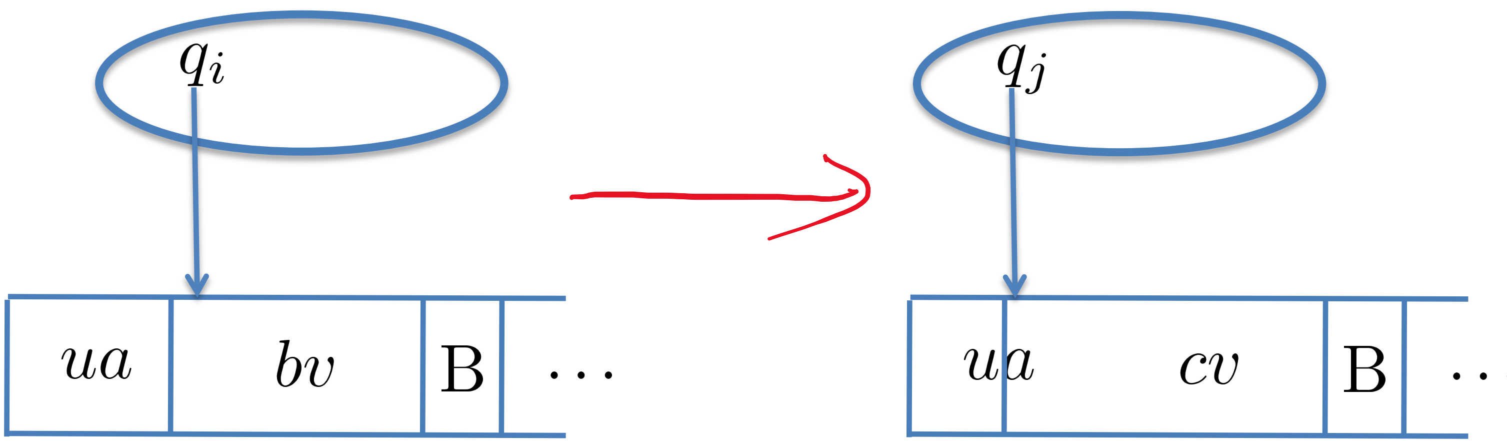
u v

$(01100)q_5(01000)$

Turing machine

A (deterministic) Turing machine (TM) $\mathbf{M}$
is a 7-tuple $\mathbf{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

Let $q_i, q_j \in Q$, $a, b, c \in \Gamma$, and $u, v \in \Gamma^*$.
Consider configuration uaq_ibv and transition $\delta(q_i, b) = (q_j, c, L)$.
Then the resulting configuration is uq_jacv



Turing machine

A (deterministic) Turing machine (TM) $\mathbf{M}$
is a 7-tuple $\mathbf{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

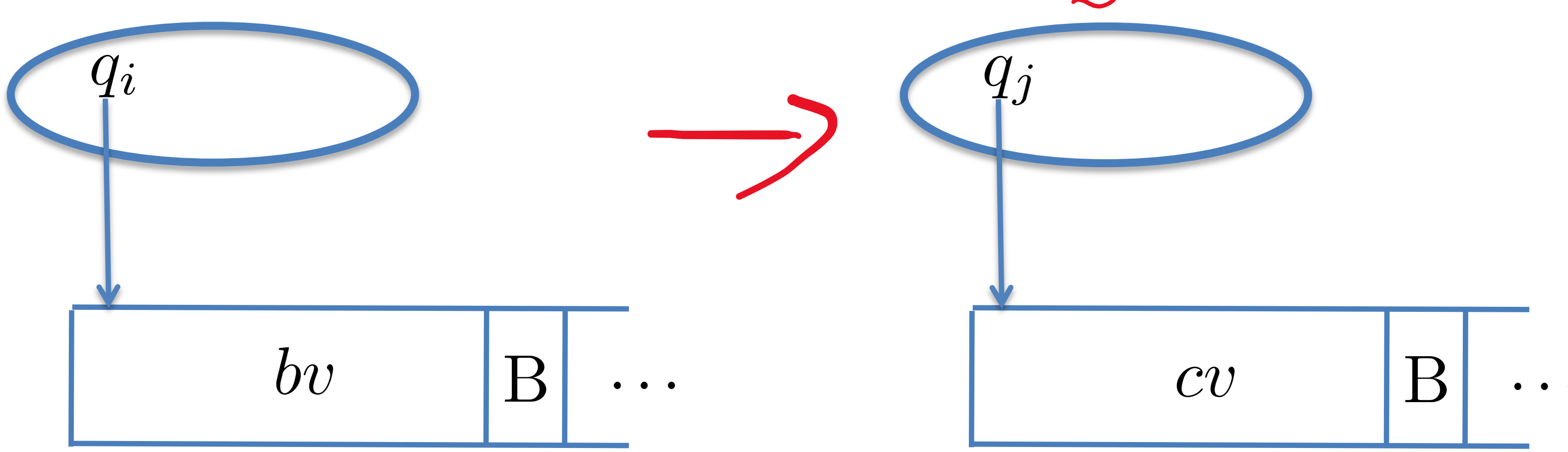
Let $q_i, q_j \in Q$, $a, b, c \in \Gamma$, and $u, v \in \Gamma^*$.
Consider configuration uaq_ibv and transition $\delta(q_i, b) = (q_j, c, L)$.
Then the resulting configuration is uq_jacv

Say configuration uaq_ibv *derives* configuration uq_jacv

Turing machine

A (deterministic) Turing machine (TM) $\mathbf{M}$
is a 7-tuple $\mathbf{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

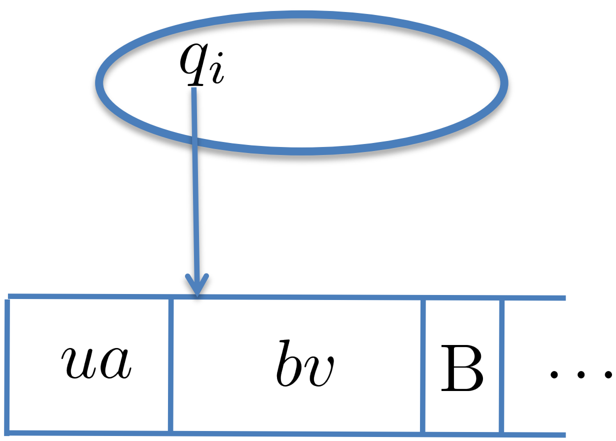
Let $q_i, q_j \in Q$, $a, b, c \in \Gamma$, and $u, v \in \Gamma^*$.
Consider configuration $q_i bv$ and transition $\delta(q_i, b) = (q_j, c, L)$.
Then the derived configuration is $q_j cv$

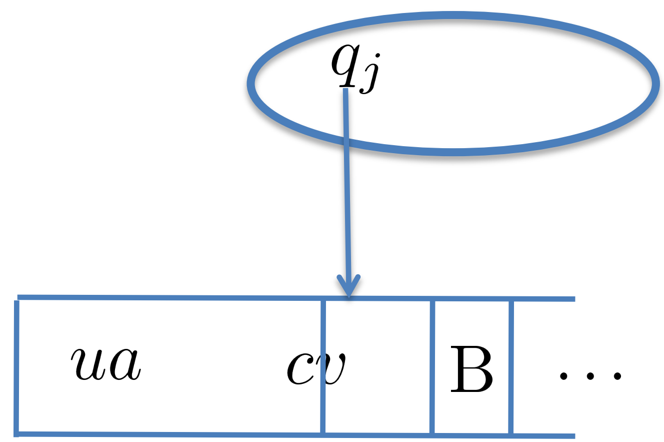


Turing machine

A (deterministic) Turing machine (TM) $\mathbf{M}$
is a 7-tuple $\mathbf{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

Let $q_i, q_j \in Q$, $a, b, c \in \Gamma$, and $u, v \in \Gamma^*$.
Consider configuration uaq_ibv and transition $\delta(q_i, b) = (q_j, c, R)$.
Then the derived configuration is $uacq_jv$

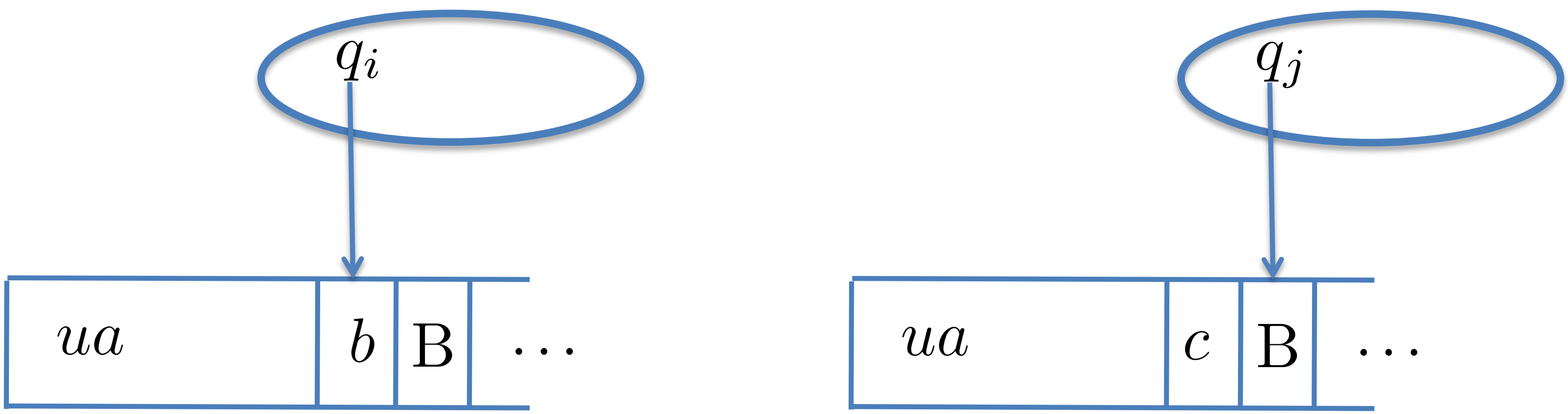




Turing machine

A (deterministic) Turing machine (TM) $\mathbf{M}$
is a 7-tuple $\mathbf{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

Let $q_i, q_j \in Q$, $a, b, c \in \Gamma$, and $u, v \in \Gamma^*$.
Consider configuration uaq_ib and transition $\delta(q_i, b) = (q_j, c, R)$.
Then the derived configuration is $uacq_j$



Turing machine

A (deterministic) Turing machine (TM) $\mathbf{M}$
is a 7-tuple $\mathbf{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

Three special configurations:
Start configuration: $q_0 s$, where s is the input string
Accept configuration: $u q_a v$
Reject configuration: $u q_r v$

Turing machine

A (deterministic) Turing machine (TM) $\mathbf{M}$
is a 7-tuple $\mathbf{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

Three special configurations:
Start configuration: $q_0 s$, where s is the input string
Accept configuration: $u q_a v$
Reject configuration: $u q_r v$

(halting configurations)

The (configuration) transition of TM can halt
by reaching a halting configuration.

Turing machine

A (deterministic) Turing machine (TM) $\mathbf{M}$
is a 7-tuple $\mathbf{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

Three special configurations:
Start configuration: $q_0 s$, where s is the input string
Accept configuration: $uq_a v$
Reject configuration: $uq_r v$
(halting configurations)

The (configuration) transition of TM can halt
by reaching a halting configuration.

If this never happens, TM loops forever.