Lecture 4

Examples of TMs and Computers vs Turing machines

Turing Machine for Parity

Turing Machine for Parity

Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of $1\text{s}\}$

Turing Machine for Parity

Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Turing Machine for Parity

Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea:

Turing Machine for Parity

Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$.

Turing Machine for Parity

Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing Machine for Parity

Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

Turing Machine for Parity

Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.

Turing Machine for Parity

Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$

Turing Machine for Parity

Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$

Turing Machine for Parity

Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$ →

Turing Machine for Parity

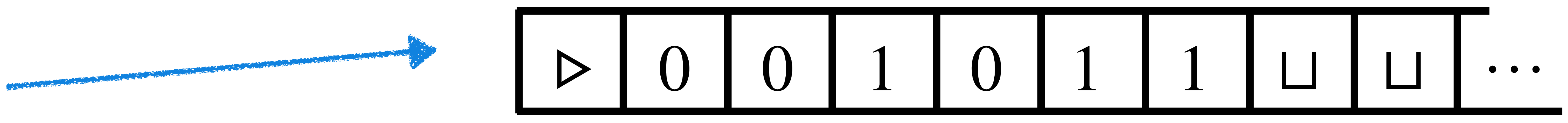
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$



Turing Machine for Parity

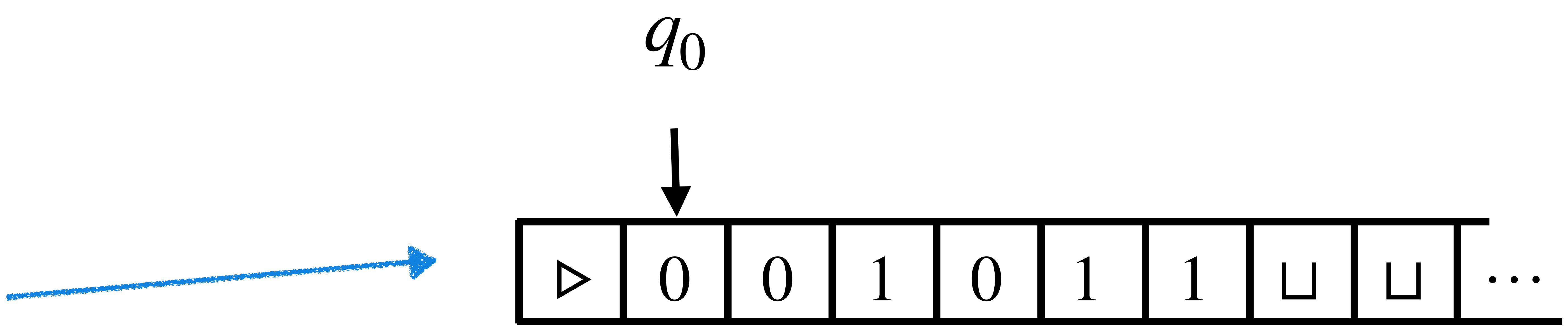
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$



Turing Machine for Parity

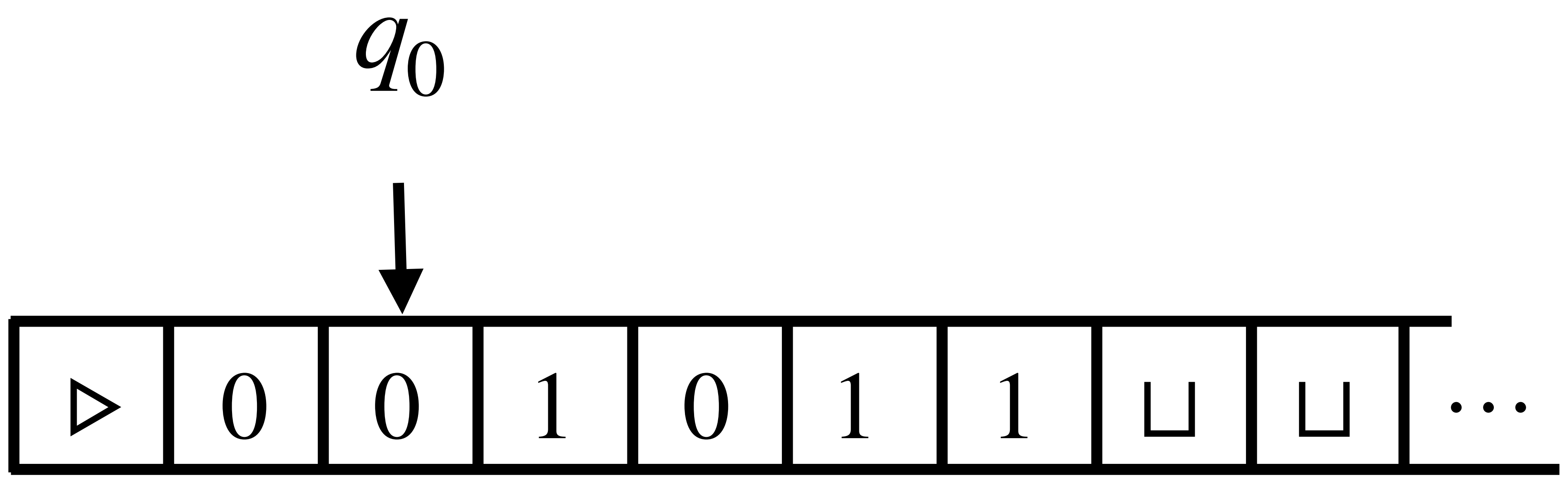
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of $1\text{s}\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$



Turing Machine for Parity

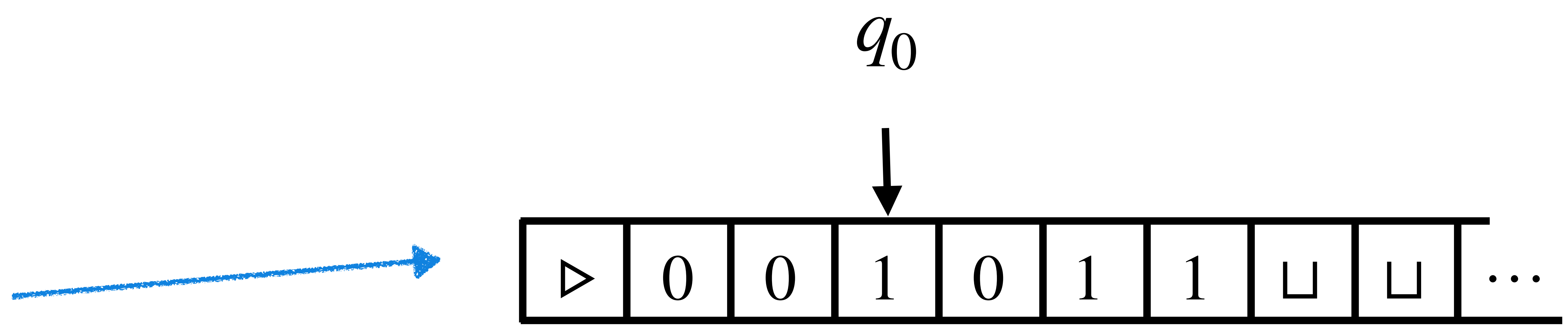
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$



Turing Machine for Parity

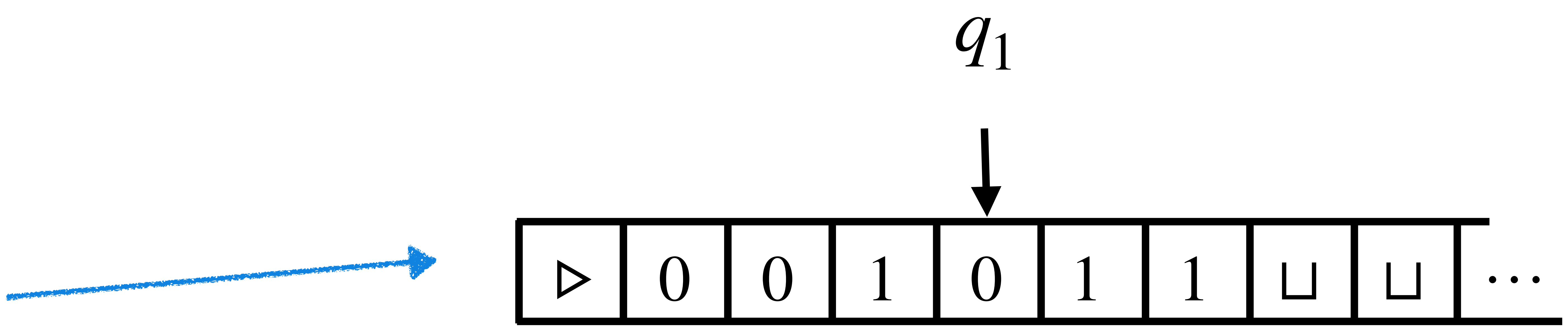
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$



Turing Machine for Parity

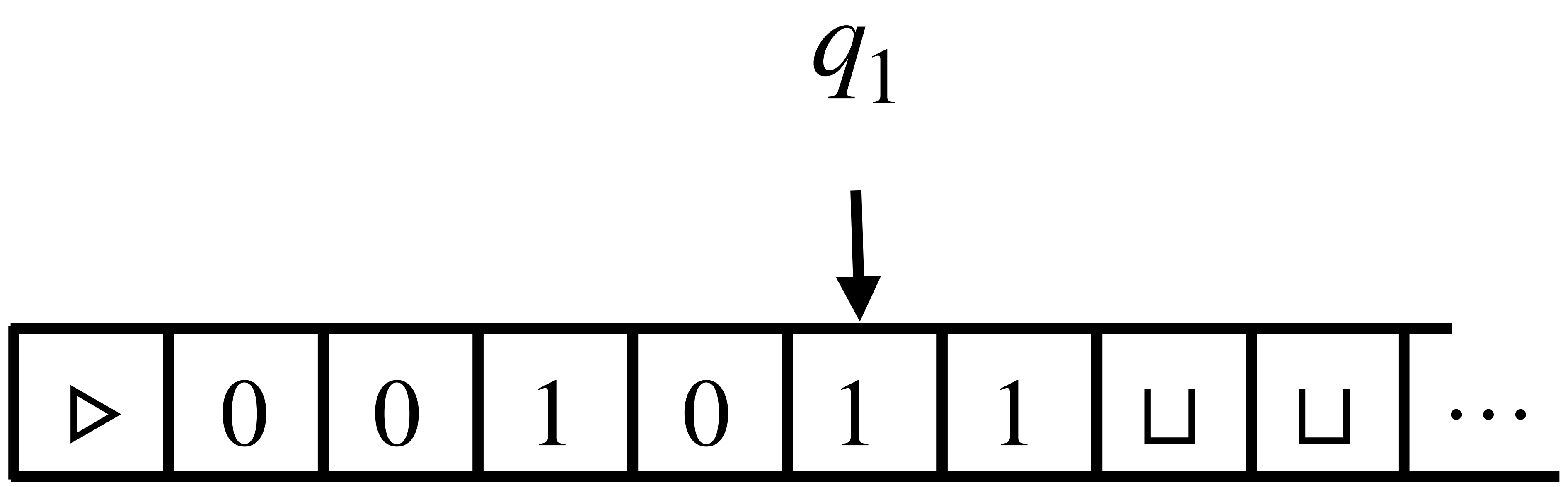
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$



Turing Machine for Parity

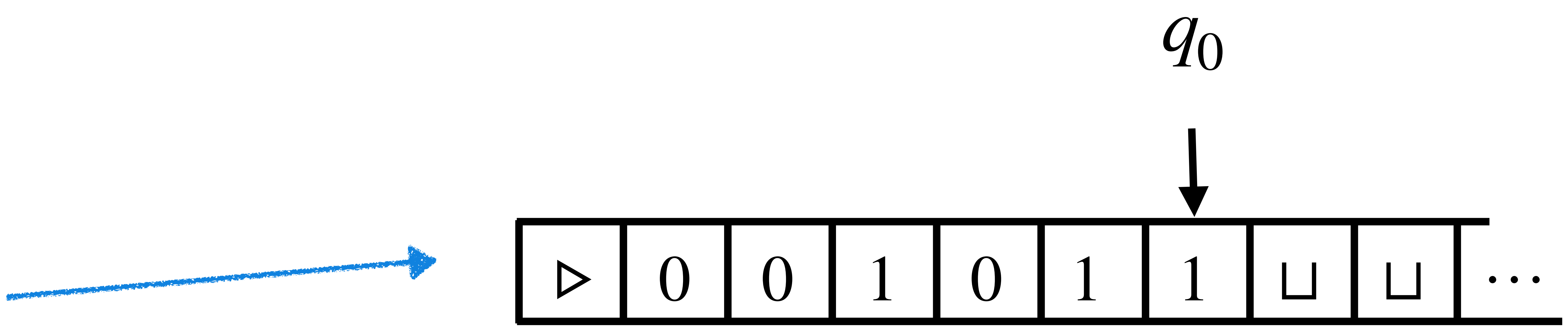
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$



Turing Machine for Parity

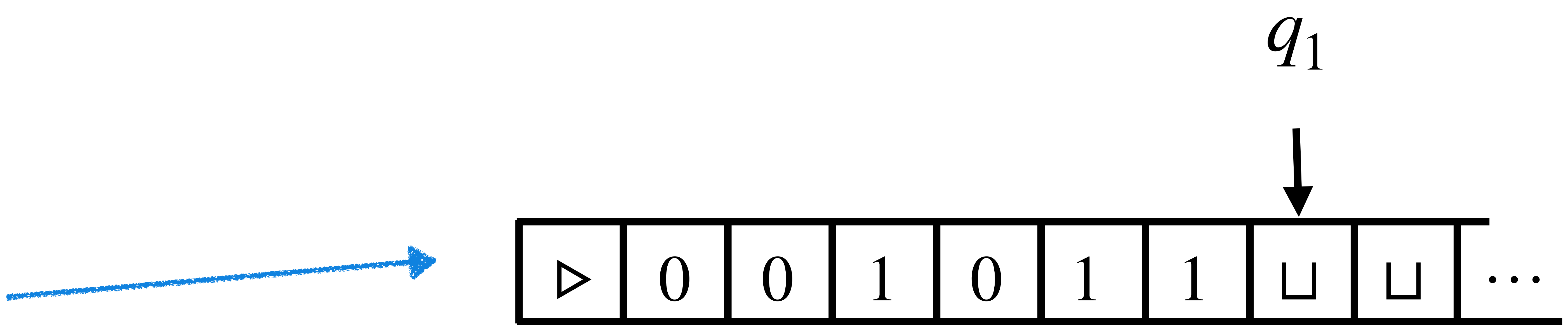
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$



Turing Machine for Parity

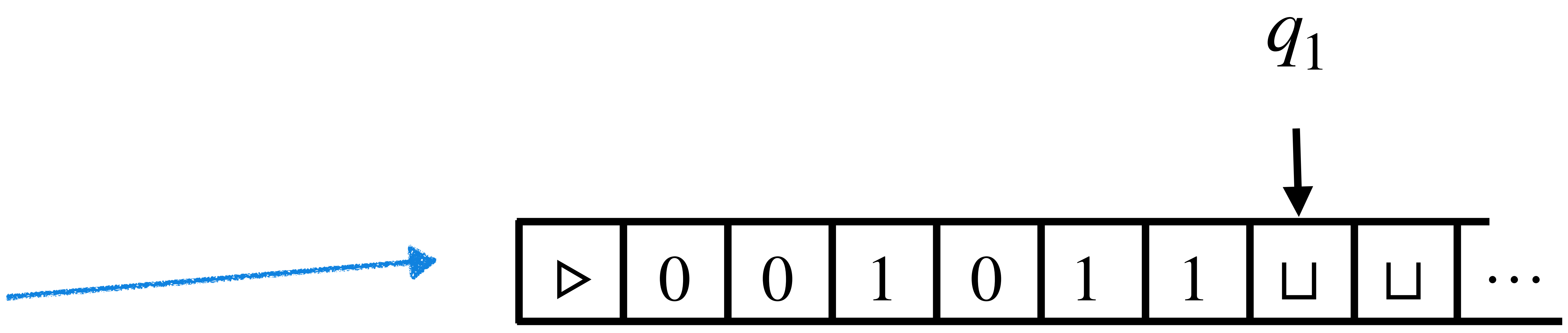
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$
- δ :



Turing Machine for Parity

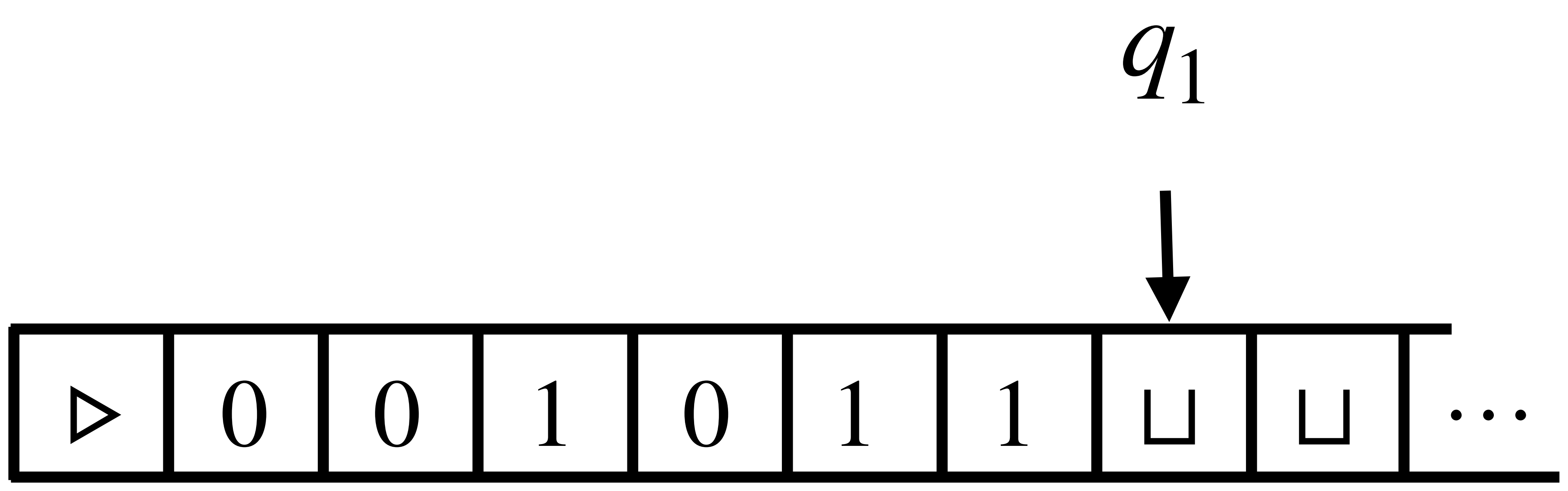
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$
- $\delta: (q_{start}, \triangleright, \triangleright) = (q_0, \triangleright, R, R)$



Turing Machine for Parity

Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

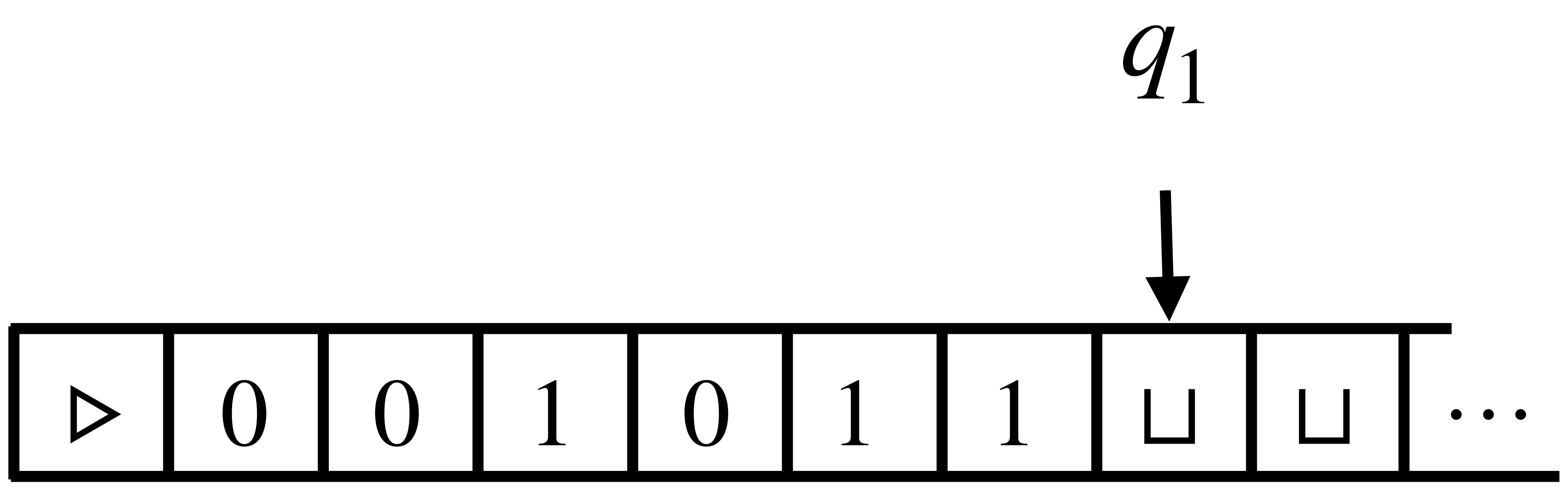
Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$
- $\delta:\ (q_{start}, \triangleright, \triangleright) = (q_0, \triangleright, R, R)$

  $(q_0, 0, \sqcup) = (q_0, \sqcup, R, S)$



Turing Machine for Parity

Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

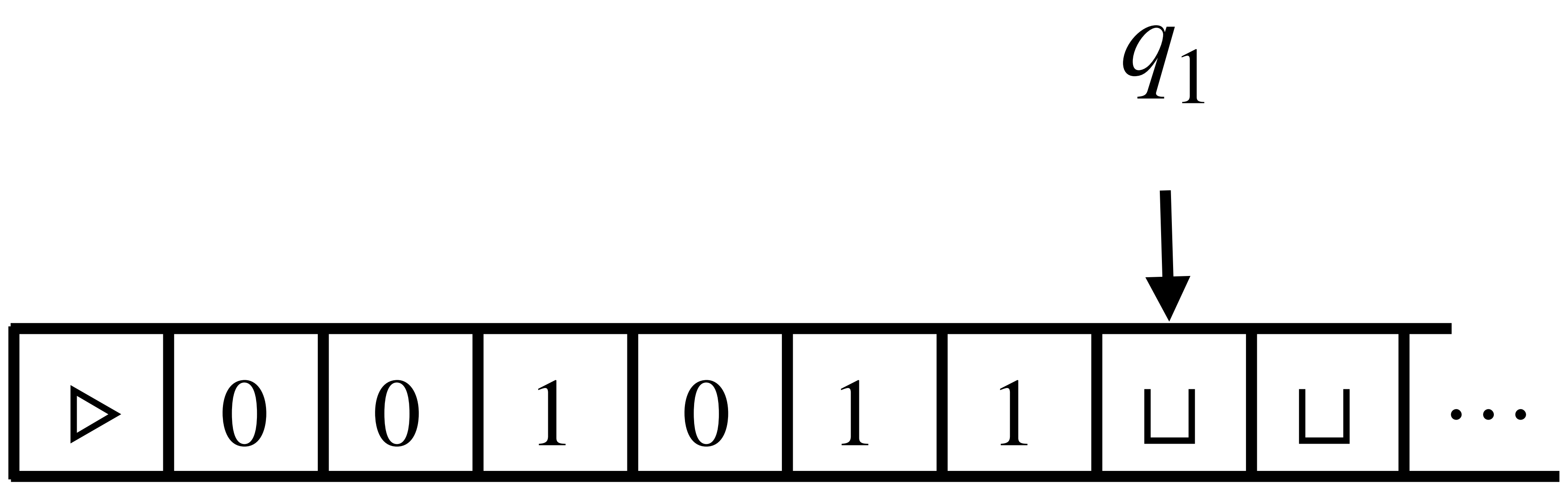
Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$
- $\delta : (q_{start}, \triangleright, \triangleright) = (q_0, \triangleright, R, R)$

$(q_0, 0, \sqcup) = (q_0, \sqcup, R, S)$ $(q_0, 1, \sqcup) = (q_1, \sqcup, R, S)$



Turing Machine for Parity

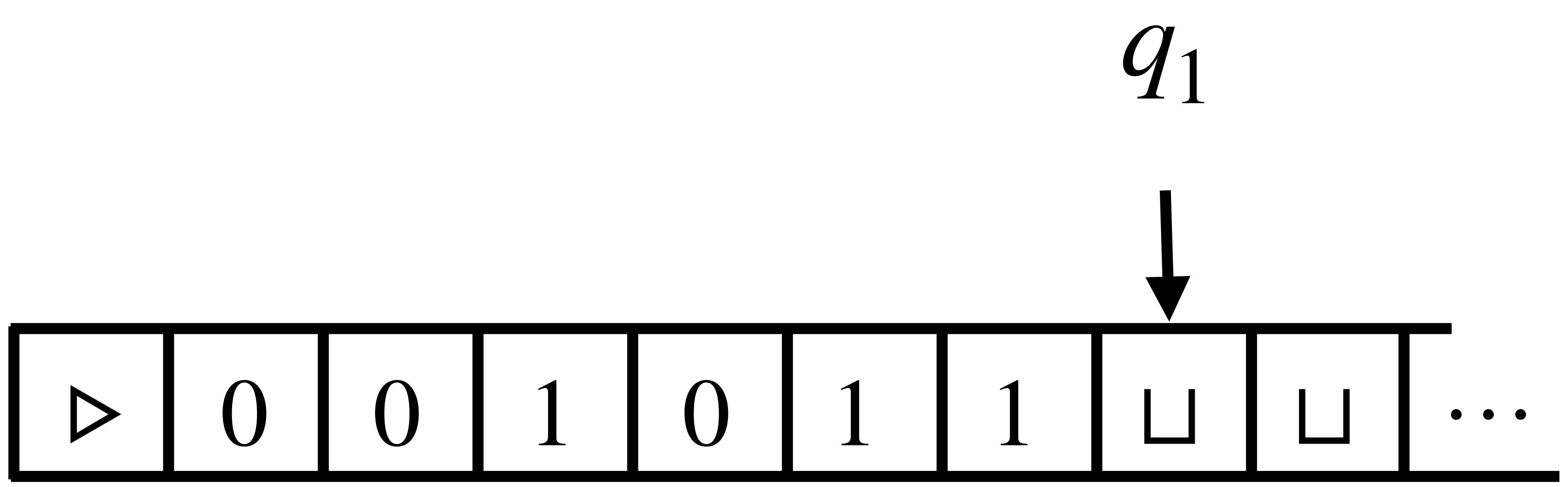
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$



- $\delta: \ (q_{start}, \triangleright, \triangleright) = (q_0, \triangleright, R, R)$

$(q_0, 0, \sqcup) = (q_0, \sqcup, R, S) \quad (q_0, 1, \sqcup) = (q_1, \sqcup, R, S) \quad (q_1, 0, \sqcup) = (q_1, \sqcup, R, S)$

Turing Machine for Parity

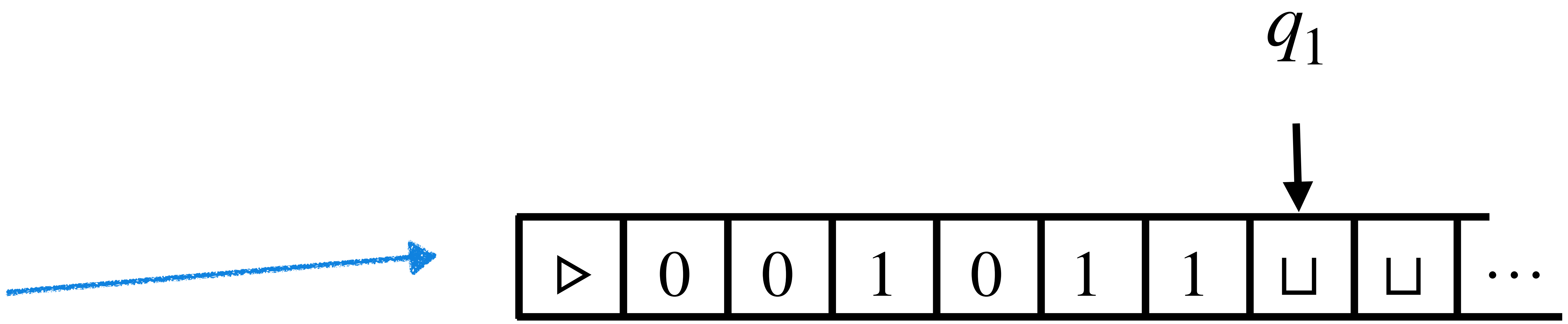
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$
- $\delta :\ (q_{start}, \triangleright, \triangleright) = (q_0, \triangleright, R, R)$



$(q_0, 0, \sqcup) = (q_0, \sqcup, R, S)$ $(q_0, 1, \sqcup) = (q_1, \sqcup, R, S)$ $(q_1, 0, \sqcup) = (q_1, \sqcup, R, S)$ $(q_1, 1, \sqcup) = (q_0, \sqcup, R, S)$

Turing Machine for Parity

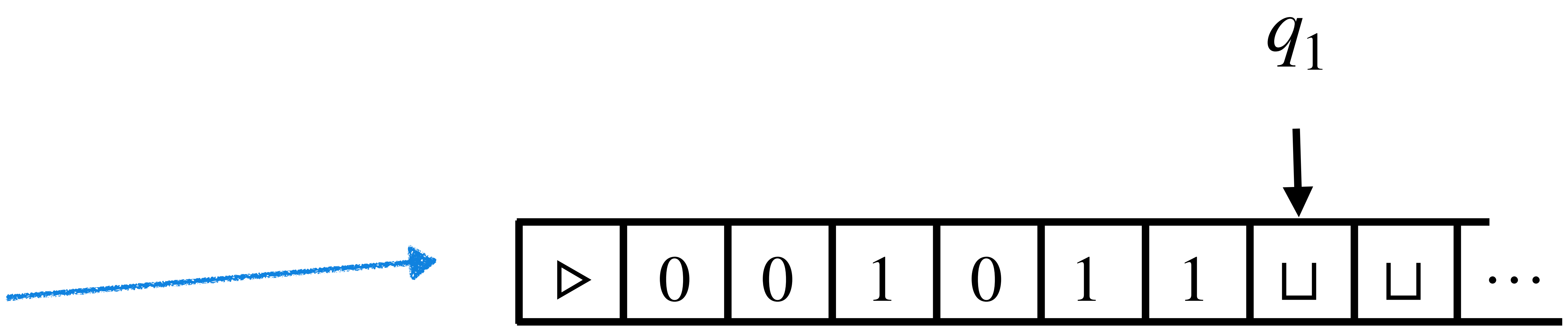
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$
- $\delta:\ (q_{start}, \triangleright, \triangleright) = (q_0, \triangleright, R, R)$



$(q_0, 0, \sqcup) = (q_0, \sqcup, R, S)$ $(q_0, 1, \sqcup) = (q_1, \sqcup, R, S)$ $(q_1, 0, \sqcup) = (q_1, \sqcup, R, S)$ $(q_1, 1, \sqcup) = (q_0, \sqcup, R, S)$

$(q_0, \sqcup, \sqcup) = (q_{halt}, 0, S, S)$

Turing Machine for Parity

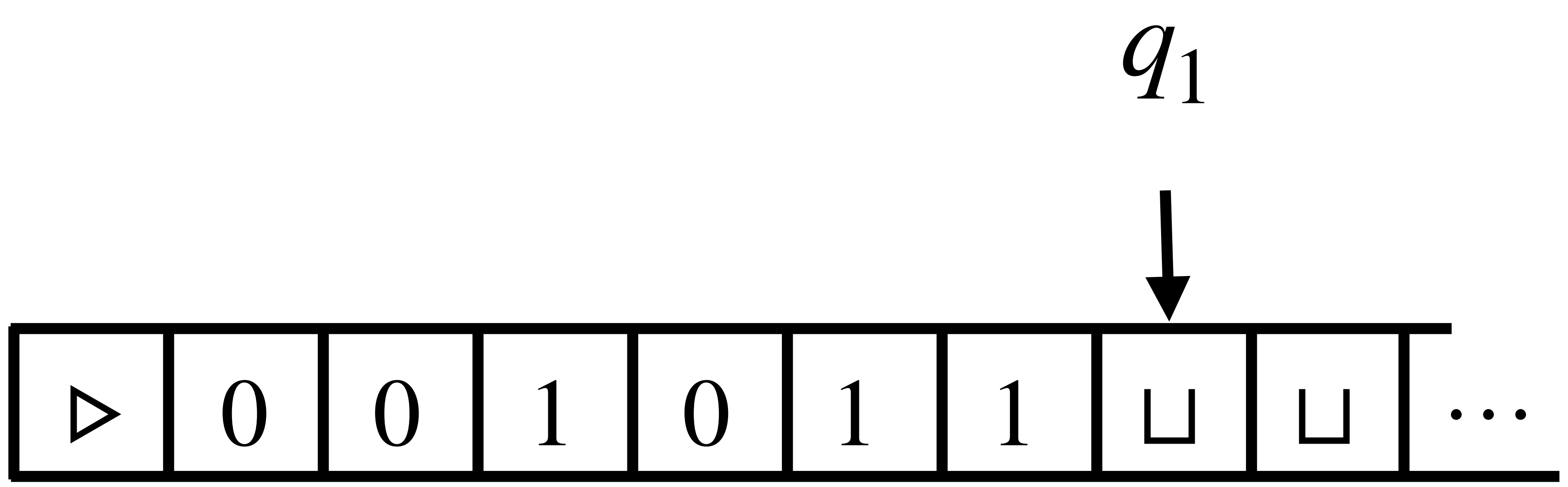
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$



- δ : $(q_{start}, \triangleright, \triangleright) = (q_0, \triangleright, R, R)$

$(q_0, 0, \sqcup) = (q_0, \sqcup, R, S)$ $(q_0, 1, \sqcup) = (q_1, \sqcup, R, S)$ $(q_1, 0, \sqcup) = (q_1, \sqcup, R, S)$ $(q_1, 1, \sqcup) = (q_0, \sqcup, R, S)$

$(q_0, \sqcup, \sqcup) = (q_{halt}, 0, S, S)$ $(q_1, \sqcup, \sqcup) = (q_{halt}, 1, S, S)$

Turing Machine for Parity

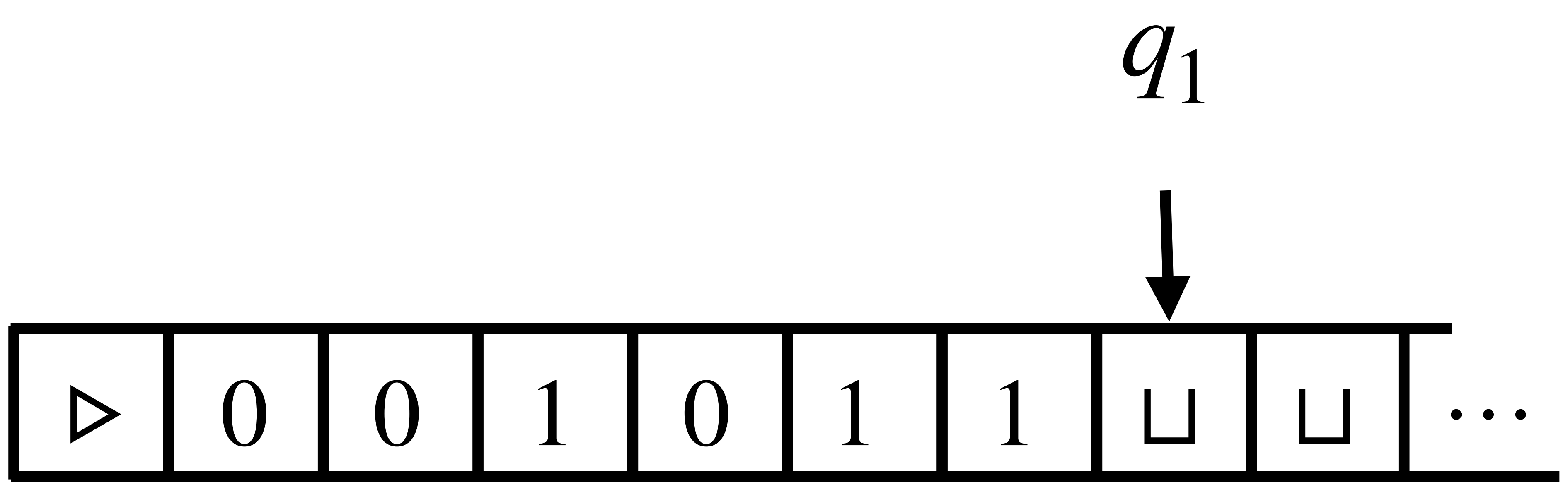
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$
- $\delta:\ (q_{start}, \triangleright, \triangleright) = (q_0, \triangleright, R, R)$



$(q_0, 0, \sqcup) = (q_0, \sqcup, R, S)$ $(q_0, 1, \sqcup) = (q_1, \sqcup, R, S)$ $(q_1, 0, \sqcup) = (q_1, \sqcup, R, S)$ $(q_1, 1, \sqcup) = (q_0, \sqcup, R, S)$

$(q_0, \sqcup, \sqcup) = (q_{halt}, 0, S, S)$ $(q_1, \sqcup, \sqcup) = (q_{halt}, 1, S, S)$ $\ldots (q_{start}/q_0/q_1, \_, \_) = (\_, \_, \_, \_) \ldots$

Turing Machine for Parity

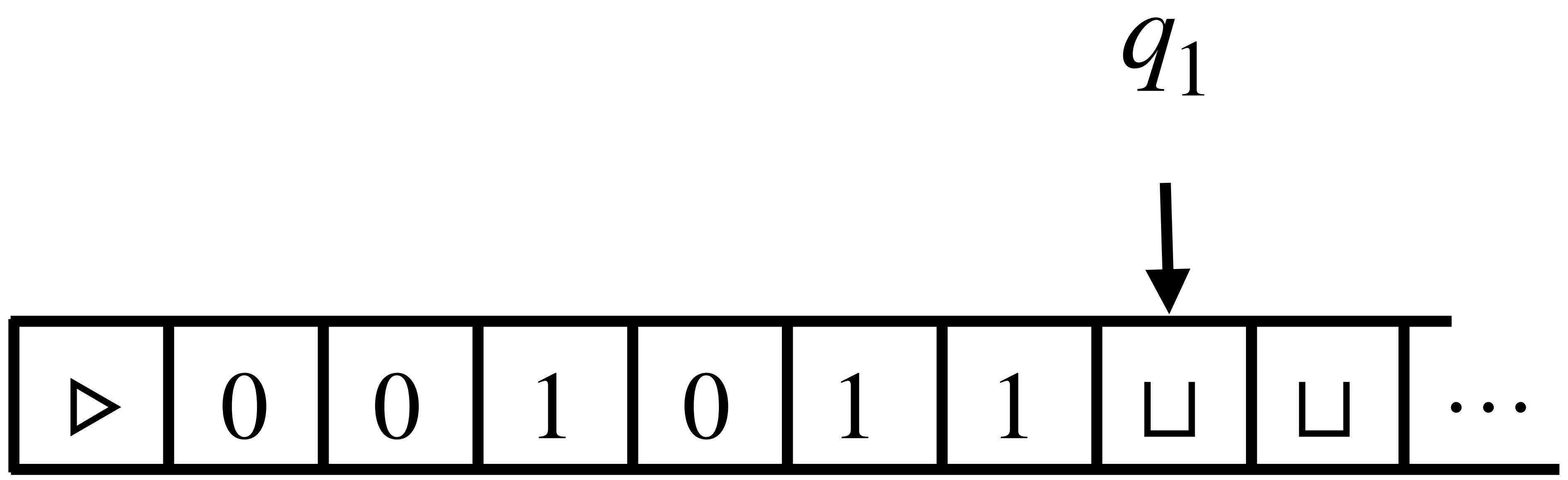
Example: Construct a TM for *PARITY* = $\{x \mid x$ is a binary string with odd number of 1s$\}$

Solution:

Idea: Start with $parity = 0$. Scan x from left to right and flip the $parity$ when you see a 1.

Turing machine:

- Two tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_0, q_1, q_{halt}\}$
- $\delta:\ (q_{start}, \triangleright, \triangleright) = (q_0, \triangleright, R, R)$



$(q_0,0,\sqcup) = (q_0, \sqcup, R, S)$ $(q_0,1,\sqcup) = (q_1, \sqcup, R, S)$ $(q_1,0,\sqcup) = (q_1, \sqcup, R, S)$ $(q_1,1,\sqcup) = (q_0, \sqcup, R, S)$

$(q_0, \sqcup, \sqcup) = (q_{halt}, 0, S, S)$ $(q_1, \sqcup, \sqcup) = (q_{halt}, 1, S, S)$ $\ldots\ (q_{start}/q_0/q_1, \_, \_) = (\_, \_, \_, \_)\ \ldots$ $\blacksquare$

Turing Machine for Palindrome

...

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

…

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

…

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

Idea:

…

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

Idea: Copy the input tape to a work tape.

. . .

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit

...

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit and the work tape head to the rightmost bit.

…

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit and the work tape head to the rightmost bit. Start comparing bits while appropriately

...

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x \text{ is a palindromic binary string}\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit and the work tape head to the rightmost bit. Start comparing bits while appropriately shifting tape heads.

…

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit and the work tape head to the rightmost bit. Start comparing bits while appropriately shifting tape heads.

Turing machine:

. . .

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit and the work tape head to the rightmost bit. Start comparing bits while appropriately shifting tape heads.

Turing machine:

- Three tapes.

...

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x \text{ is a palindromic binary string}\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit and the work tape head to the rightmost bit. Start comparing bits while appropriately shifting tape heads.

Turing machine:

- Three tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$

…

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x \text{ is a palindromic binary string}\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit and the work tape head to the rightmost bit. Start comparing bits while appropriately shifting tape heads.

Turing machine:

- Three tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_{copy}, q_{comp}, q_{left}, q_{halt}\}$

…

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit and the work tape head to the rightmost bit. Start comparing bits while appropriately shifting tape heads.

Turing machine:

- Three tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_{copy}, q_{comp}, q_{left}, q_{halt}\}$

…

Turing Machine for Palindrome

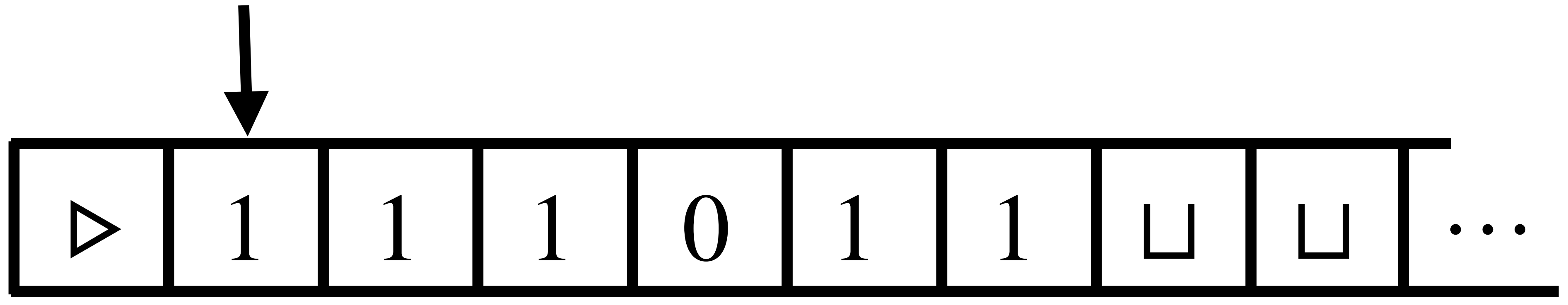
Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit and the work tape head to the rightmost bit. Start comparing bits while appropriately shifting tape heads.

Turing machine:

- Three tapes.
- $\Gamma = \{0,1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_{copy}, q_{comp}, q_{left}, q_{halt}\}$



…

Turing Machine for Palindrome

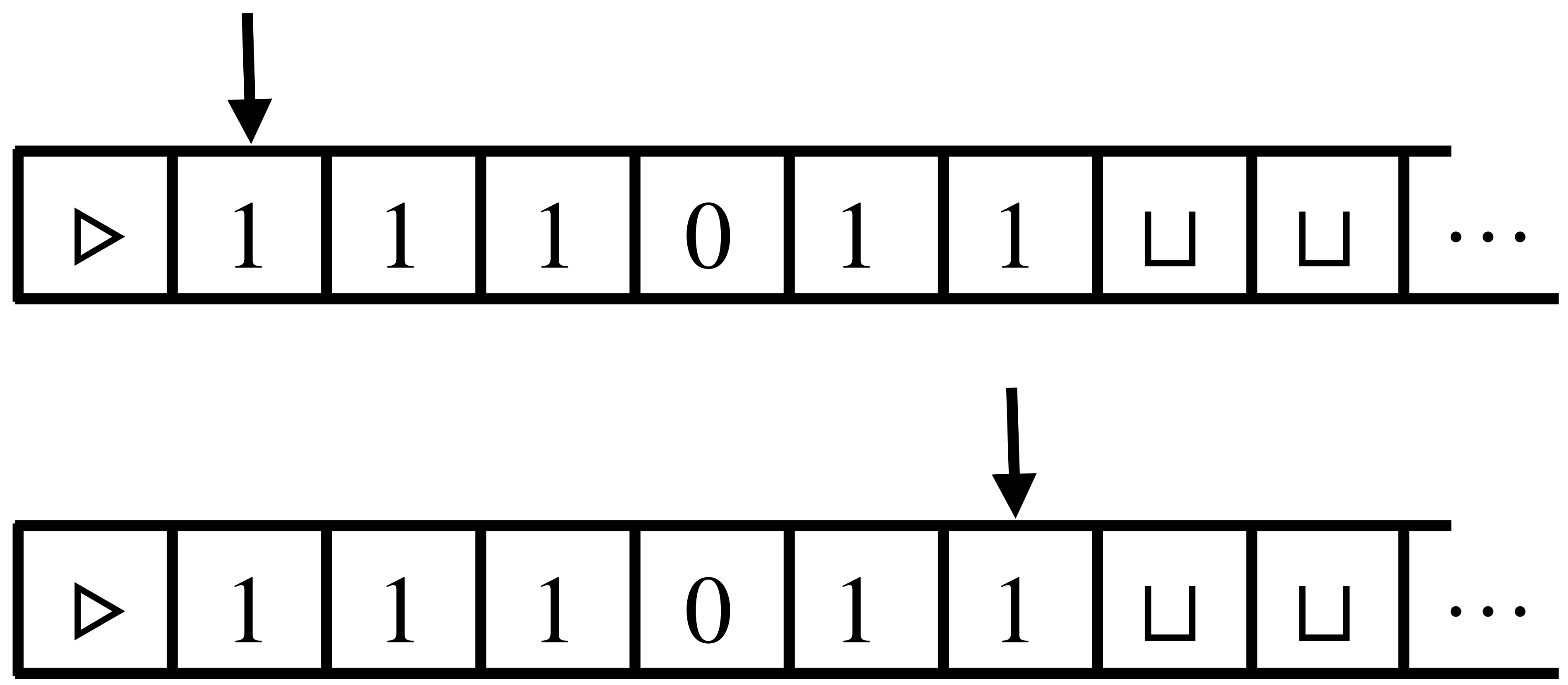
Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit and the work tape head to the rightmost bit. Start comparing bits while appropriately shifting tape heads.

Turing machine:

- Three tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_{copy}, q_{comp}, q_{left}, q_{halt}\}$



…

Turing Machine for Palindrome

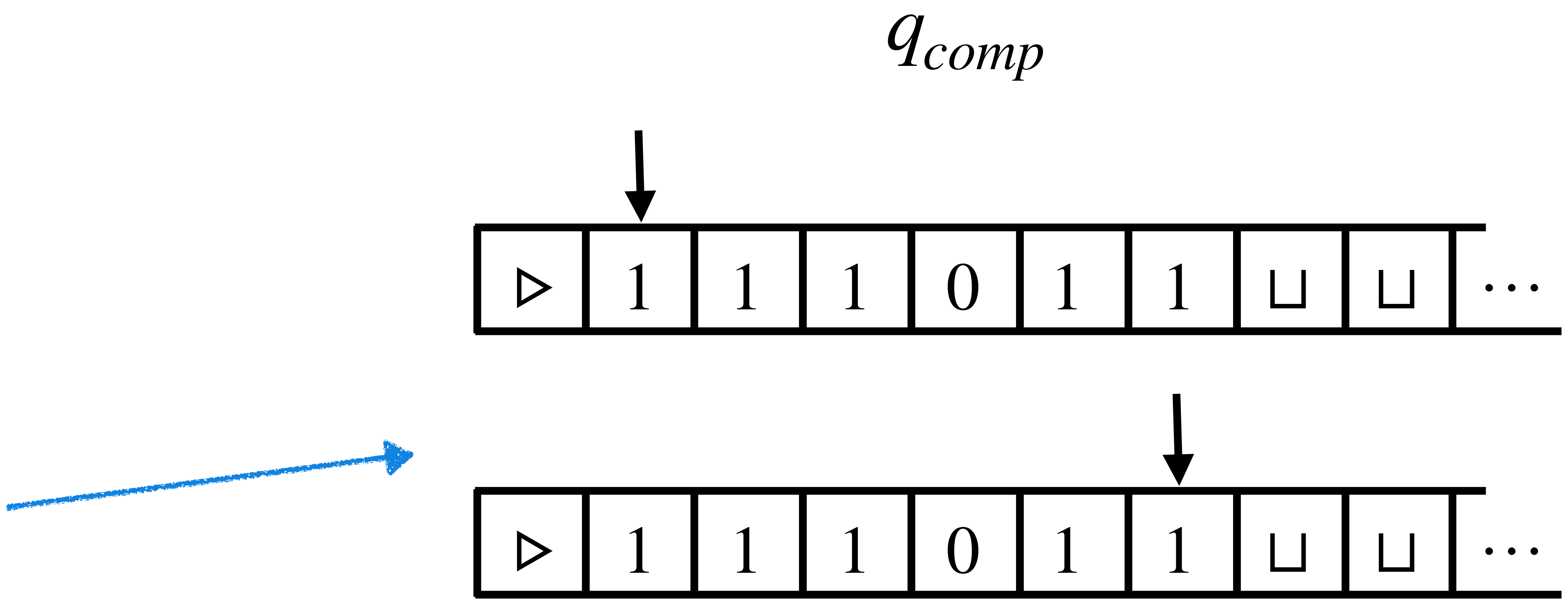
Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit and the work tape head to the rightmost bit. Start comparing bits while appropriately shifting tape heads.

Turing machine:

- Three tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_{copy}, q_{comp}, q_{left}, q_{halt}\}$



…

Turing Machine for Palindrome

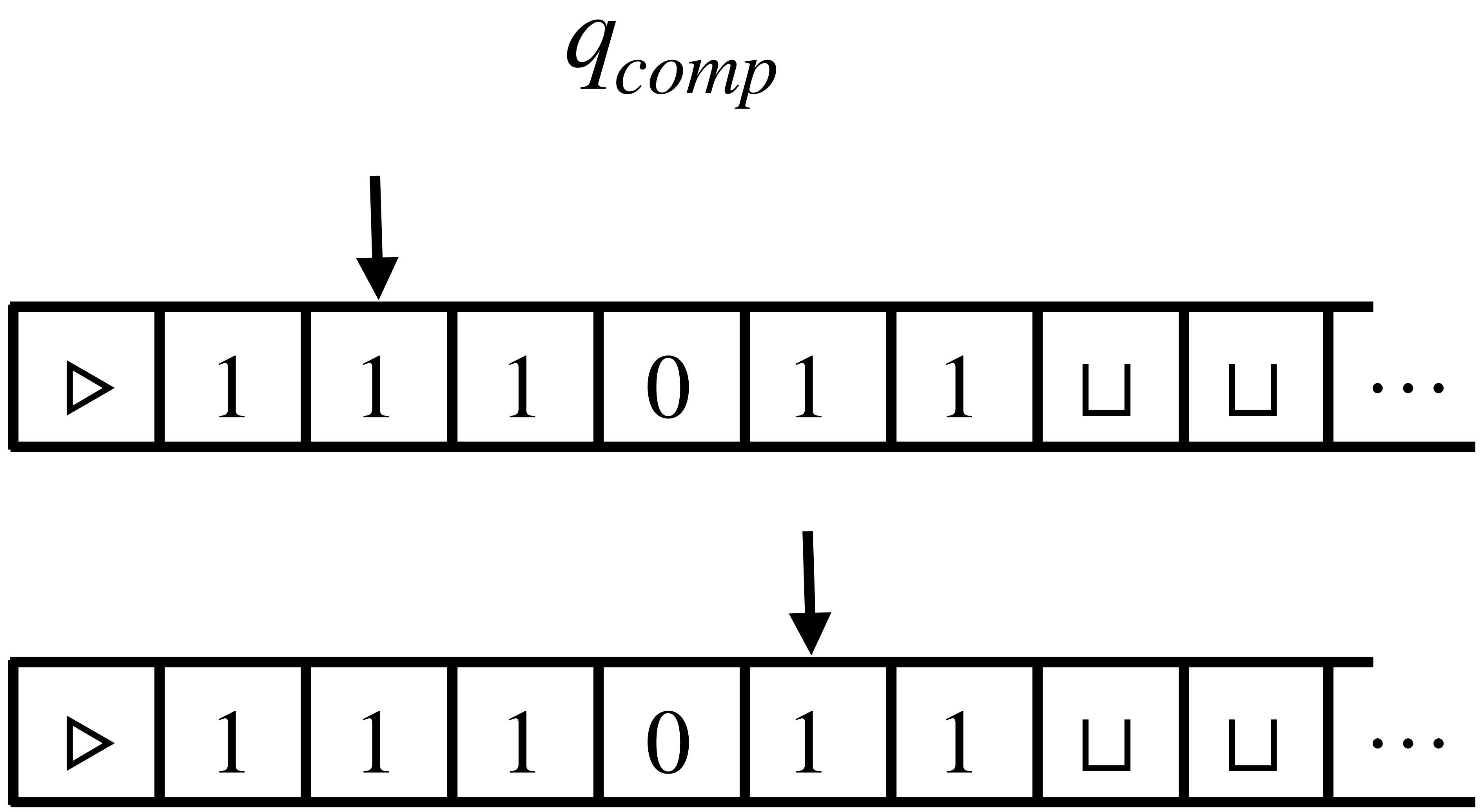
Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit and the work tape head to the rightmost bit. Start comparing bits while appropriately shifting tape heads.

Turing machine:

- Three tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_{copy}, q_{comp}, q_{left}, q_{halt}\}$



…

Turing Machine for Palindrome

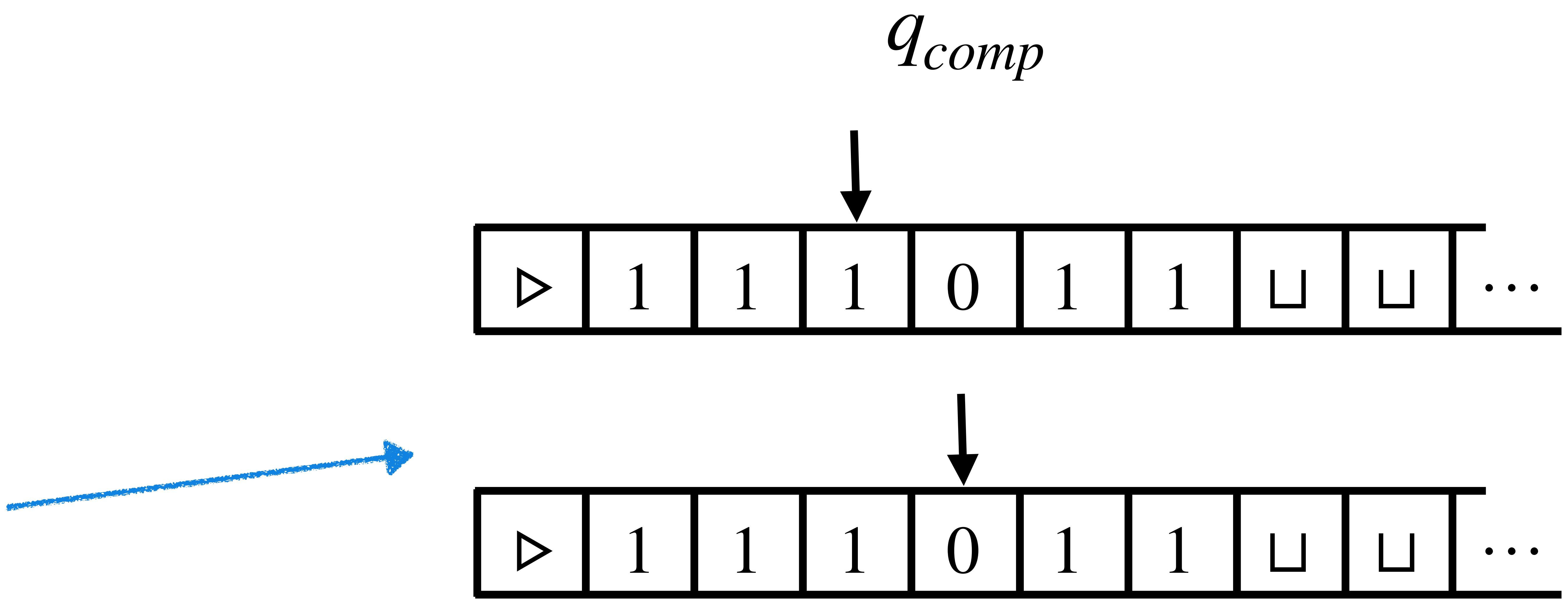
Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:

Idea: Copy the input tape to a work tape. Keep the input tape head to the leftmost bit and the work tape head to the rightmost bit. Start comparing bits while appropriately shifting tape heads.

Turing machine:

- Three tapes.
- $\Gamma = \{0, 1, \triangleright, \sqcup\}$
- $Q = \{q_{start}, q_{copy}, q_{comp}, q_{left}, q_{halt}\}$



…

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution:
- δ :

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution: • $\delta : (q_{start}, \triangleright, \triangleright, \triangleright) = (q_{copy}, \triangleright, \triangleright, R, R, R)$

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution: • $\delta : (q_{start}, \triangleright, \triangleright, \triangleright) = (q_{copy}, \triangleright, \triangleright, R, R, R)$

$$(q_{copy}, 0, \sqcup, \sqcup) = (q_{copy}, 0, \sqcup, R, R, S)$$

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution: • $\delta : (q_{start}, \triangleright, \triangleright, \triangleright) = (q_{copy}, \triangleright, \triangleright, R, R, R)$

$(q_{copy}, 0, \sqcup, \sqcup) = (q_{copy}, 0, \sqcup, R, R, S)$ $(q_{copy}, 1, \sqcup, \sqcup) = (q_{copy}, 1, \sqcup, R, R, S)$

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution: • $\delta : (q_{start}, \triangleright, \triangleright, \triangleright) = (q_{copy}, \triangleright, \triangleright, R, R, R)$

$(q_{copy}, 0, \sqcup, \sqcup) = (q_{copy}, 0, \sqcup, R, R, S)$ $\quad (q_{copy}, 1, \sqcup, \sqcup) = (q_{copy}, 1, \sqcup, R, R, S)$

$(q_{copy}, \sqcup, \sqcup, \sqcup) = (q_{left}, \sqcup, \sqcup, L, L, S)$

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution: • $\delta : (q_{start}, \triangleright, \triangleright, \triangleright) = (q_{copy}, \triangleright, \triangleright, R, R, R)$

$(q_{copy}, 0, \sqcup, \sqcup) = (q_{copy}, 0, \sqcup, R, R, S)$ $\quad (q_{copy}, 1, \sqcup, \sqcup) = (q_{copy}, 1, \sqcup, R, R, S)$

$(q_{copy}, \sqcup, \sqcup, \sqcup) = (q_{left}, \sqcup, \sqcup, L, L, S)$

$(q_{left}, 0/1, 0/1, \sqcup) = (q_{left}, 0/1, \sqcup, L, S, S)$

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution: • $\delta : (q_{start}, \triangleright, \triangleright, \triangleright) = (q_{copy}, \triangleright, \triangleright, R, R, R)$

$(q_{copy}, 0, \sqcup, \sqcup) = (q_{copy}, 0, \sqcup, R, R, S)$ $\quad (q_{copy}, 1, \sqcup, \sqcup) = (q_{copy}, 1, \sqcup, R, R, S)$

$(q_{copy}, \sqcup, \sqcup, \sqcup) = (q_{left}, \sqcup, \sqcup, L, L, S)$

$(q_{left}, 0/1, 0/1, \sqcup) = (q_{left}, 0/1, \sqcup, L, S, S)$ $\quad (q_{left}, \triangleright, 0/1, \sqcup) = (q_{comp}, 0/1, \sqcup, R, S, S)$

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution: • $\delta : (q_{start}, \triangleright, \triangleright, \triangleright) = (q_{copy}, \triangleright, \triangleright, R, R, R)$

$(q_{copy}, 0, \sqcup, \sqcup) = (q_{copy}, 0, \sqcup, R, R, S)$ $\quad (q_{copy}, 1, \sqcup, \sqcup) = (q_{copy}, 1, \sqcup, R, R, S)$

$(q_{copy}, \sqcup, \sqcup, \sqcup) = (q_{left}, \sqcup, \sqcup, L, L, S)$

$(q_{left}, 0/1, 0/1, \sqcup) = (q_{left}, 0/1, \sqcup, L, S, S)$ $\quad (q_{left}, \triangleright, 0/1, \sqcup) = (q_{comp}, 0/1, \sqcup, R, S, S)$

$(q_{comp}, 0, 0, \sqcup) = (q_{comp}, 0, \sqcup, R, L, S)$

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution: • $\delta : (q_{start}, \triangleright, \triangleright, \triangleright) = (q_{copy}, \triangleright, \triangleright, R, R, R)$

$(q_{copy}, 0, \sqcup, \sqcup) = (q_{copy}, 0, \sqcup, R, R, S)$ $\quad (q_{copy}, 1, \sqcup, \sqcup) = (q_{copy}, 1, \sqcup, R, R, S)$

$(q_{copy}, \sqcup, \sqcup, \sqcup) = (q_{left}, \sqcup, \sqcup, L, L, S)$

$(q_{left}, 0/1, 0/1, \sqcup) = (q_{left}, 0/1, \sqcup, L, S, S)$ $\quad (q_{left}, \triangleright, 0/1, \sqcup) = (q_{comp}, 0/1, \sqcup, R, S, S)$

$(q_{comp}, 0, 0, \sqcup) = (q_{comp}, 0, \sqcup, R, L, S)$ $\quad (q_{comp}, 1, 1, \sqcup) = (q_{comp}, 1, \sqcup, R, L, S)$

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution: • $\delta: (q_{start}, \triangleright, \triangleright, \triangleright) = (q_{copy}, \triangleright, \triangleright, R, R, R)$

$(q_{copy}, 0, \sqcup, \sqcup) = (q_{copy}, 0, \sqcup, R, R, S)$ $\quad (q_{copy}, 1, \sqcup, \sqcup) = (q_{copy}, 1, \sqcup, R, R, S)$

$(q_{copy}, \sqcup, \sqcup, \sqcup) = (q_{left}, \sqcup, \sqcup, L, L, S)$

$(q_{left}, 0/1, 0/1, \sqcup) = (q_{left}, 0/1, \sqcup, L, S, S)$ $\quad (q_{left}, \triangleright, 0/1, \sqcup) = (q_{comp}, 0/1, \sqcup, R, S, S)$

$(q_{comp}, 0, 0, \sqcup) = (q_{comp}, 0, \sqcup, R, L, S)$ $\quad (q_{comp}, 1, 1, \sqcup) = (q_{comp}, 1, \sqcup, R, L, S)$

$(q_{comp}, 1, 0, \sqcup) = (q_{halt}, 0, 0, S, S, S)$

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution: • $\delta : (q_{start}, \triangleright, \triangleright, \triangleright) = (q_{copy}, \triangleright, \triangleright, R, R, R)$

$(q_{copy}, 0, \sqcup, \sqcup) = (q_{copy}, 0, \sqcup, R, R, S)$ $\quad (q_{copy}, 1, \sqcup, \sqcup) = (q_{copy}, 1, \sqcup, R, R, S)$

$(q_{copy}, \sqcup, \sqcup, \sqcup) = (q_{left}, \sqcup, \sqcup, L, L, S)$

$(q_{left}, 0/1, 0/1, \sqcup) = (q_{left}, 0/1, \sqcup, L, S, S)$ $\quad (q_{left}, \triangleright, 0/1, \sqcup) = (q_{comp}, 0/1, \sqcup, R, S, S)$

$(q_{comp}, 0, 0, \sqcup) = (q_{comp}, 0, \sqcup, R, L, S)$ $\quad (q_{comp}, 1, 1, \sqcup) = (q_{comp}, 1, \sqcup, R, L, S)$

$(q_{comp}, 1, 0, \sqcup) = (q_{halt}, 0, 0, S, S, S)$ $\quad (q_{comp}, 0, 1, \sqcup) = (q_{halt}, 1, 0, S, S, S)$

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution: • $\delta: (q_{start}, \triangleright, \triangleright, \triangleright) = (q_{copy}, \triangleright, \triangleright, R, R, R)$

$(q_{copy}, 0, \sqcup, \sqcup) = (q_{copy}, 0, \sqcup, R, R, S)$ $\quad (q_{copy}, 1, \sqcup, \sqcup) = (q_{copy}, 1, \sqcup, R, R, S)$

$(q_{copy}, \sqcup, \sqcup, \sqcup) = (q_{left}, \sqcup, \sqcup, L, L, S)$

$(q_{left}, 0/1, 0/1, \sqcup) = (q_{left}, 0/1, \sqcup, L, S, S)$ $\quad (q_{left}, \triangleright, 0/1, \sqcup) = (q_{comp}, 0/1, \sqcup, R, S, S)$

$(q_{comp}, 0, 0, \sqcup) = (q_{comp}, 0, \sqcup, R, L, S)$ $\quad (q_{comp}, 1, 1, \sqcup) = (q_{comp}, 1, \sqcup, R, L, S)$

$(q_{comp}, 1, 0, \sqcup) = (q_{halt}, 0, 0, S, S, S)$ $\quad (q_{comp}, 0, 1, \sqcup) = (q_{halt}, 1, 0, S, S, S)$

$(q_{comp}, \sqcup, \triangleright, \sqcup) = (q_{halt}, \triangleright, 1, S, S, S)$

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution: • $\delta:$ $(q_{start}, \triangleright, \triangleright, \triangleright) = (q_{copy}, \triangleright, \triangleright, R, R, R)$

$(q_{copy}, 0, \sqcup, \sqcup) = (q_{copy}, 0, \sqcup, R, R, S)$ $\quad (q_{copy}, 1, \sqcup, \sqcup) = (q_{copy}, 1, \sqcup, R, R, S)$

$(q_{copy}, \sqcup, \sqcup, \sqcup) = (q_{left}, \sqcup, \sqcup, L, L, S)$

$(q_{left}, 0/1, 0/1, \sqcup) = (q_{left}, 0/1, \sqcup, L, S, S)$ $\quad (q_{left}, \triangleright, 0/1, \sqcup) = (q_{comp}, 0/1, \sqcup, R, S, S)$

$(q_{comp}, 0, 0, \sqcup) = (q_{comp}, 0, \sqcup, R, L, S)$ $\quad (q_{comp}, 1, 1, \sqcup) = (q_{comp}, 1, \sqcup, R, L, S)$

$(q_{comp}, 1, 0, \sqcup) = (q_{halt}, 0, 0, S, S, S)$ $\quad (q_{comp}, 0, 1, \sqcup) = (q_{halt}, 1, 0, S, S, S)$

$(q_{comp}, \sqcup, \triangleright, \sqcup) = (q_{halt}, \triangleright, 1, S, S, S)$

■

Turing Machine for Palindrome

Example: Construct a TM for *PALIN* = $\{x \mid x$ is a palindromic binary string$\}$

Solution: • $\delta : (q_{start}, \triangleright, \triangleright, \triangleright) = (q_{copy}, \triangleright, \triangleright, R, R, R)$

$(q_{copy}, 0, \sqcup, \sqcup) = (q_{copy}, 0, \sqcup, R, R, S)$ $\quad (q_{copy}, 1, \sqcup, \sqcup) = (q_{copy}, 1, \sqcup, R, R, S)$

$(q_{copy}, \sqcup, \sqcup, \sqcup) = (q_{left}, \sqcup, \sqcup, L, L, S)$

$(q_{left}, 0/1, 0/1, \sqcup) = (q_{left}, 0/1, \sqcup, L, S, S)$ $\quad (q_{left}, \triangleright, 0/1, \sqcup) = (q_{comp}, 0/1, \sqcup, R, S, S)$

$(q_{comp}, 0, 0, \sqcup) = (q_{comp}, 0, \sqcup, R, L, S)$ $\quad (q_{comp}, 1, 1, \sqcup) = (q_{comp}, 1, \sqcup, R, L, S)$

$(q_{comp}, 1, 0, \sqcup) = (q_{halt}, 0, 0, S, S, S)$ $\quad (q_{comp}, 0, 1, \sqcup) = (q_{halt}, 1, 0, S, S, S)$

$(q_{comp}, \sqcup, \triangleright, \sqcup) = (q_{halt}, \triangleright, 1, S, S, S)$ ■

Q: Can we solve PALIN using two tapes?

Computer vs Turing Machine

Computer vs Turing Machine

Simulating a Computer by Turing machine

Computer vs Turing Machine

Simulating a Computer by Turing machine

High-level language programs can be translated into a assembly language program which

Computer vs Turing Machine

Simulating a Computer by Turing machine

High-level language programs can be translated into a assembly language program which is a **finite sequence** of instructions of type:

Computer vs Turing Machine

Simulating a Computer by Turing machine

High-level language programs can be translated into a assembly language program which is a **finite sequence** of instructions of type:

- **Move data** from memory into registers or vice-versa.

Computer vs Turing Machine

Simulating a Computer by Turing machine

High-level language programs can be translated into a assembly language program which is a **finite sequence** of instructions of type:

- **Move data** from memory into registers or vice-versa. E.g., $MOV\ A\ [B]$.

Computer vs Turing Machine

Simulating a Computer by Turing machine

High-level language programs can be translated into a assembly language program which is a **finite sequence** of instructions of type:

- **Move data** from memory into registers or vice-versa. E.g., $MOV\ A\ [B]$.
- **Add or multiply** the content of two registers into some register.

Computer vs Turing Machine

Simulating a Computer by Turing machine

High-level language programs can be translated into a assembly language program which is a **finite sequence** of instructions of type:

- **Move data** from memory into registers or vice-versa. E.g., $MOV\ A\ [B]$.
- **Add or multiply** the content of two registers into some register. E.g., $MUL\ C\ D$.

Computer vs Turing Machine

Simulating a Computer by Turing machine

High-level language programs can be translated into a assembly language program which is a **finite sequence** of instructions of type:

- **Move data** from memory into registers or vice-versa. E.g., $MOV\ A\ [B]$.
- **Add or multiply** the content of two registers into some register. E.g., $MUL\ C\ D$.

Assembly language program can be simulated by a TM by:

Computer vs Turing Machine

Simulating a Computer by Turing machine

High-level language programs can be translated into a assembly language program which is a **finite sequence** of instructions of type:

- **Move data** from memory into registers or vice-versa. E.g., $MOV\ A\ [B]$.
- **Add or multiply** the content of two registers into some register. E.g., $MUL\ C\ D$.

Assembly language program can be simulated by a TM by:

- Allocating portions of tape for registers, memory,

Computer vs Turing Machine

Simulating a Computer by Turing machine

High-level language programs can be translated into a assembly language program which is a **finite sequence** of instructions of type:

- **Move data** from memory into registers or vice-versa. E.g., $MOV\ A\ [B]$.
- **Add or multiply** the content of two registers into some register. E.g., $MUL\ C\ D$.

Assembly language program can be simulated by a TM by:

- Allocating portions of tape for registers, memory, and instructions.

Computer vs Turing Machine

Simulating a Computer by Turing machine

High-level language programs can be translated into a assembly language program which is a **finite sequence** of instructions of type:

- **Move data** from memory into registers or vice-versa. E.g., $MOV\ A\ [B]$.
- **Add or multiply** the content of two registers into some register. E.g., $MUL\ C\ D$.

Assembly language program can be simulated by a TM by:

- Allocating portions of tape for registers, memory, and instructions.

Computer vs Turing Machine

Simulating a Computer by Turing machine

High-level language programs can be translated into a assembly language program which is a **finite sequence** of instructions of type:

- **Move data** from memory into registers or vice-versa. E.g., $MOV\ A\ [B]$.
- **Add or multiply** the content of two registers into some register. E.g., $MUL\ C\ D$.

Assembly language program can be simulated by a TM by:

- Allocating portions of tape for registers, memory, and instructions.

Computer vs Turing Machine

Simulating a Computer by Turing machine

High-level language programs can be translated into a assembly language program which is a **finite sequence** of instructions of type:

- **Move data** from memory into registers or vice-versa. E.g., $MOV\ A\ [B]$.
- **Add or multiply** the content of two registers into some register. E.g., $MUL\ C\ D$.

Assembly language program can be simulated by a TM by:

- Allocating portions of tape for registers, memory, and instructions.

Computer vs Turing Machine

Simulating a Computer by Turing machine

High-level language programs can be translated into a assembly language program which is a **finite sequence** of instructions of type:

- **Move data** from memory into registers or vice-versa. E.g., $MOV\ A\ [B]$.
- **Add or multiply** the content of two registers into some register. E.g., $MUL\ C\ D$.

Assembly language program can be simulated by a TM by:

- Allocating portions of tape for registers, memory, and instructions.

- Executing instructions using δ.

Computer vs Turing Machine

Computer vs Turing Machine

Simulating a Turing machine by Computer

Computer vs Turing Machine

Simulating a Turing machine by Computer

A C program with infinite memory can be written that simulates a Turing machine where:

Computer vs Turing Machine

Simulating a Turing machine by Computer

A C program with infinite memory can be written that simulates a Turing machine where:

- Infinite arrays can act as the tapes of the TM.

Computer vs Turing Machine

Simulating a Turing machine by Computer

A C program with infinite memory can be written that simulates a Turing machine where:

- Infinite arrays can act as the tapes of the TM.
- Transition function's entries can be stored in a finite 2D array.

Computer vs Turing Machine

Simulating a Turing machine by Computer

A C program with infinite memory can be written that simulates a Turing machine where:

- Infinite arrays can act as the tapes of the TM.
- Transition function's entries can be stored in a finite 2D array.

Equal Power but Different Roles

Computer vs Turing Machine

Simulating a Turing machine by Computer

A C program with infinite memory can be written that simulates a Turing machine where:

- Infinite arrays can act as the tapes of the TM.
- Transition function's entries can be stored in a finite 2D array.

Equal Power but Different Roles

- High-level languages are used to **demonstrate an effective procedure** that decides

Computer vs Turing Machine

Simulating a Turing machine by Computer

A C program with infinite memory can be written that simulates a Turing machine where:

- Infinite arrays can act as the tapes of the TM.
- Transition function's entries can be stored in a finite 2D array.

Equal Power but Different Roles

- High-level languages are used to **demonstrate an effective procedure** that decides a given language because they are user-friendly.

Computer vs Turing Machine

Simulating a Turing machine by Computer

A C program with infinite memory can be written that simulates a Turing machine where:

- Infinite arrays can act as the tapes of the TM.
- Transition function's entries can be stored in a finite 2D array.

Equal Power but Different Roles

- High-level languages are used to **demonstrate an effective procedure** that decides a given language because they are user-friendly.
- Turing machines are used to **prove non-existence of an (efficient) effective procedure**

Computer vs Turing Machine

Simulating a Turing machine by Computer

A C program with infinite memory can be written that simulates a Turing machine where:

- Infinite arrays can act as the tapes of the TM.
- Transition function's entries can be stored in a finite 2D array.

Equal Power but Different Roles

- High-level languages are used to **demonstrate an effective procedure** that decides a given language because they are user-friendly.
- Turing machines are used to **prove non-existence of an (efficient) effective procedure** that decides a given language because of their simple mathematical structure.