BU CS 332 – Theory of Computation

https://forms.gle/XDLCbA5mrSRjNvZX6

Lecture 8:

- More on non-regularity
- Turing Machines

Reading:

"Myhill-Nerode" note

Sipser Ch 3.1, 3.3

Mark Bun

February 18, 2025

Last Time: Distinguishing Set Method

Definition: Strings x and y are **distinguishable** by L if there exists a "distinguishing extension" $z \in \Sigma^*$ such that exactly one of xz or yz is in L.

Definition: A set of strings S is **pairwise distinguishable** by L if every pair of distinct strings $x, y \in S$ is distinguishable by L.

Theorem: If S is pairwise distinguishable by L, then every DFA recognizing L needs at least $|S|$ states.

Corollary: If language L has an infinite pairwise distinguishable set, then L is not regular.

Reusing a Proof

Finding a distinguishing set can take some work…

Let's try to reuse that work!

How might we show that

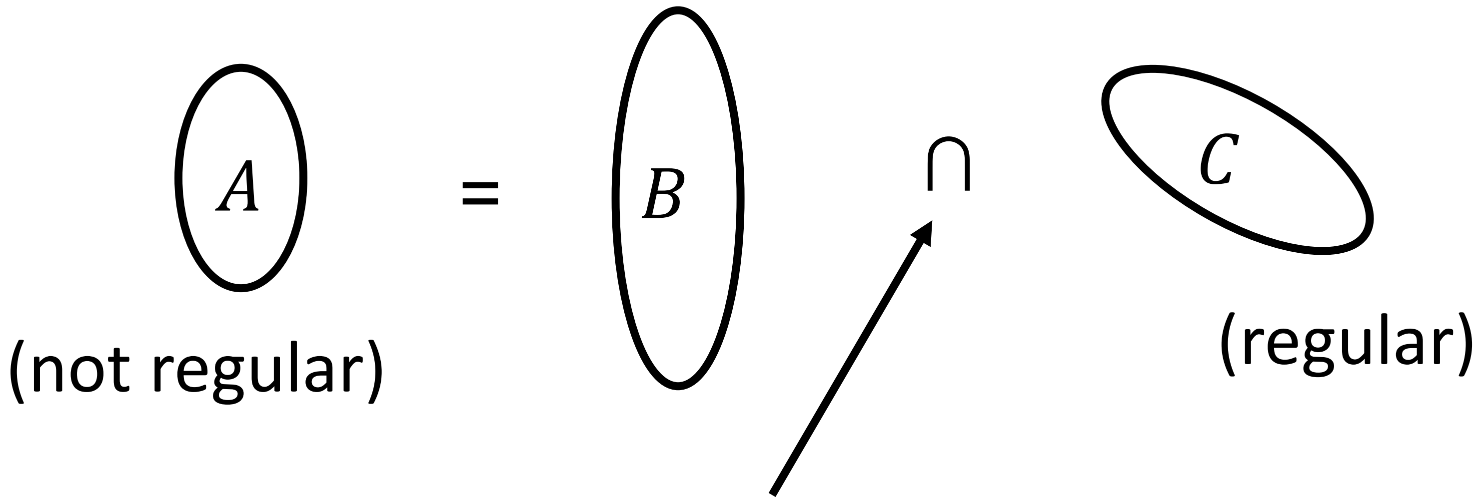
$$BALANCED = \{w \mid w \text{ has an equal \# of 0s and 1s}\}$$

is not regular?

$$\{0^n1^n \mid n \geq 0\} = BALANCED \cap \{w \mid \text{all 0s in } w \text{ appear before all 1s}\}$$

Using Closure Properties

If A is not regular, we can show a related language B is not regular



any of $\{\circ, \cup, \cap\}$ or, for one language, $\{\neg, {}^{R}, {}^{*}\}$

<u>By contradiction:</u> If B is regular, then $B \ \cap \ C \ (= \ A)$ is regular.

But A is not regular so neither is B!

Example

Prove $B = \{0^i 1^j | i \neq j\}$ is not regular using

- Nonregular language

  $A = \{0^n 1^n | n \geq 0\}$ and

- Regular language

  $C = \{w \mid \text{all 0s in } w \text{ appear before all 1s}\}$

Which of the following expresses A in terms of B and C?

a) $A = B \cap C$

b) $A = \bar{B} \cap C$

c) $A = B \cup C$

d) $A = \bar{B} \cup C$

Proof that B is nonregular

Assume for the sake of contradiction that B is regular

We know: $A = \bar{B} \cap C$

!DANGER!

Let $B = \{0^i 1^j | i \neq j\}$ and write $B = A \cup C$ where

- Nonregular language

$$A = \left\{0^i 1^j \middle| i > j \geq 0\right\} \text{ and}$$

- Nonregular language

$$C = \left\{0^i 1^j \middle| j > i \geq 0\right\} \text{ and}$$

Does this let us conclude B is nonregular?

Turing Machines

Turing Machines – Motivation

We've seen finite automata as a restricted model of computation

Finite Automata / Regular Expressions

- Can do simple pattern matching (e.g., substrings), check parity, addition
- Can't perform unbounded counting
- Can't recognize palindromes

Somewhat more powerful (not in this course):

Pushdown Automata / Context-Free Grammars

- Can count and compare, parse math expressions
- Can't recognize $\{a^n b^n c^n \mid n \geq 0\}$

Turing Machines – Motivation

Goal:

Define a model of computation that is

1) General purpose. Captures all algorithms that can be implemented in any programming language.

2) Mathematically simple. We can hope to prove that things are not computable in this model.

A Brief History

1900 – Hilbert’s Tenth Problem

Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: To devise a process according to which it can be determined in a finite number of operations whether the equation is solvable in rational integers.

David Hilbert 1862-1943

1928 – The *Entscheidungsproblem*

The "Decision Problem"

Is there an algorithm which takes as input a formula (in first-order logic) and decides whether it's logically valid?

Wilhelm Ackermann 1896-1962

David Hilbert 1862-1943

1936 – Solution to the *Entscheidungsproblem*

Alonzo Church 1903-1995

"An unsolvable problem of elementary number theory“

Model of computation: λ-calculus (CS 320)

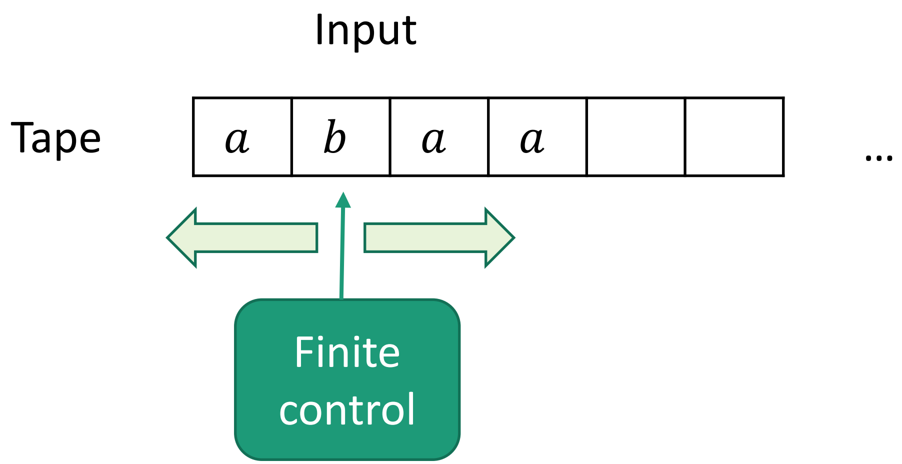
Alan Turing 1912-1954

“On computable numbers, with an application to the *Entscheidungsproblem*”

Model of computation: Turing Machine

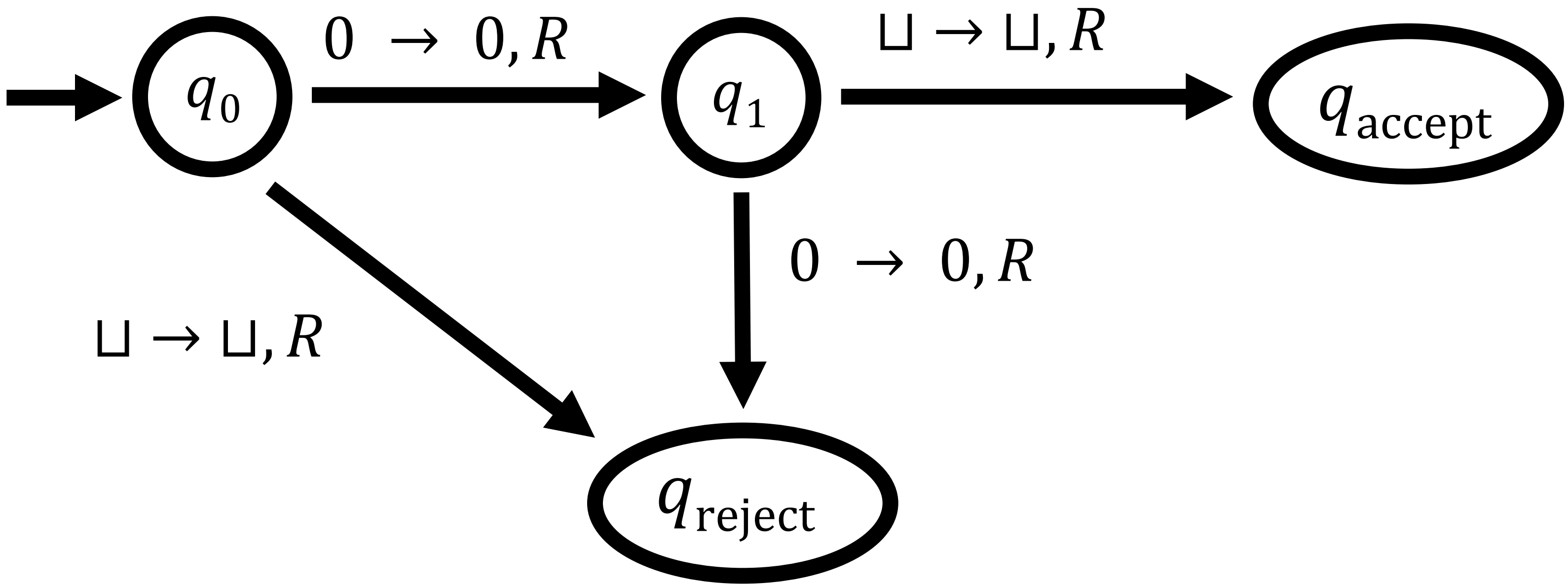
The Turing Machine Model

The Basic Turing Machine (TM)



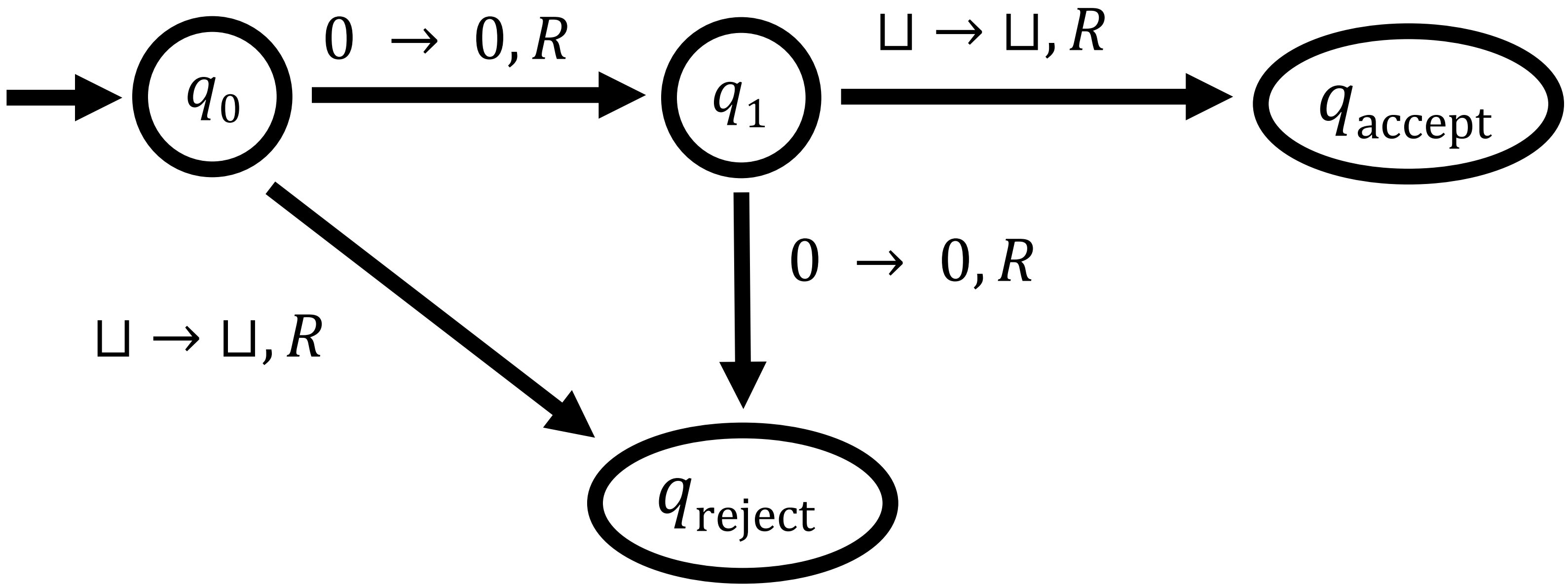
- Input is written on an infinitely long tape
- Head can both read and write, and move in both directions
- Computation halts as soon as control reaches "accept" or "reject" state

Example



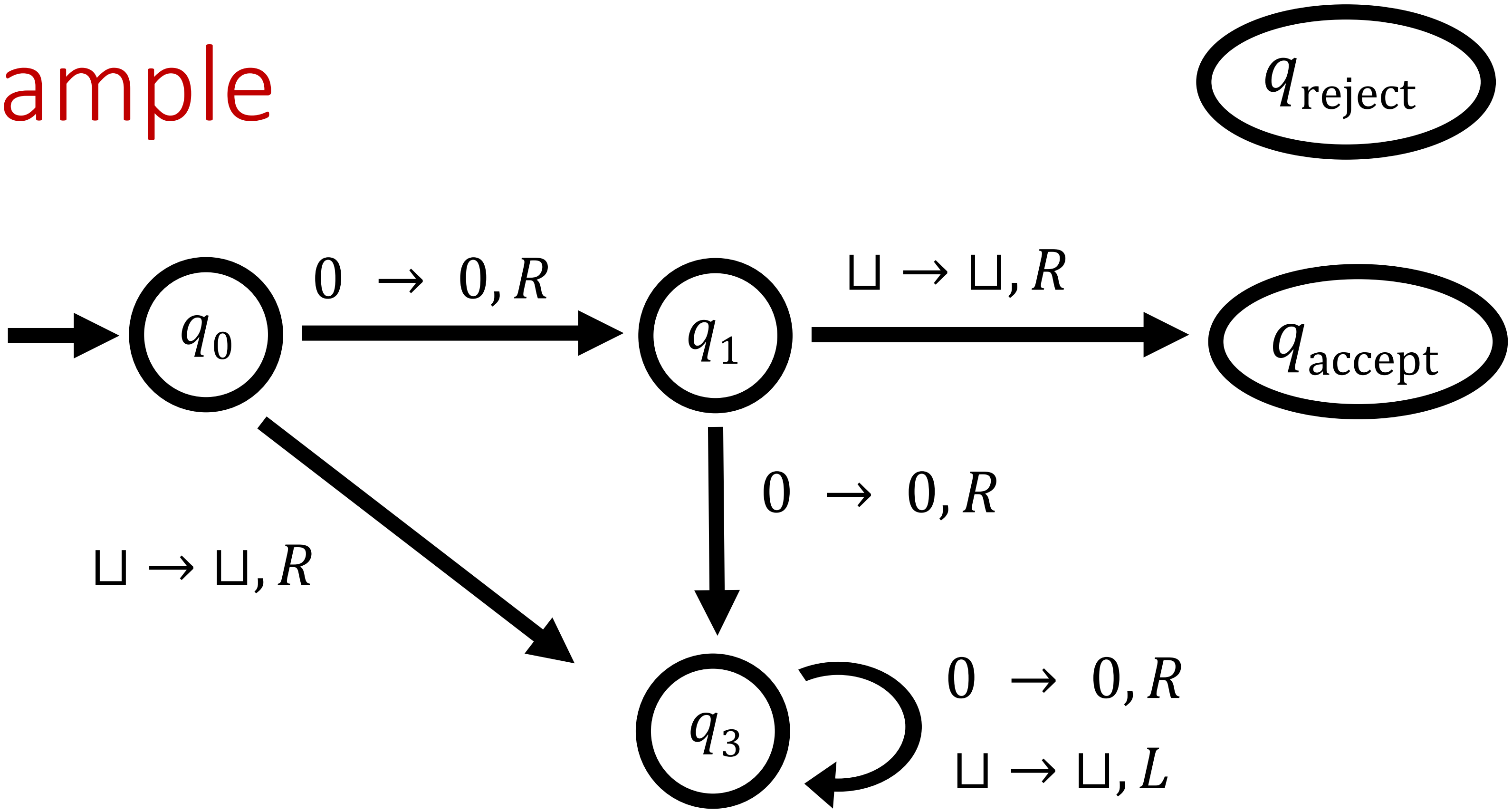
q_0
$0 \rightarrow 0, R$
q_1
$\sqcup \rightarrow \sqcup, R$
q_{accept}
$0 \rightarrow 0, R$
$\sqcup \rightarrow \sqcup, R$
q_{reject}

Example



q_0
$0 \rightarrow 0, R$
q_1
$\sqcup \rightarrow \sqcup, R$
q_{accept}
$0 \rightarrow 0, R$
$\sqcup \rightarrow \sqcup, R$
q_{reject}

Example



What does this TM do on input 000?

a) Halt and accept

b) Halt and reject

c) Halt in state q_3

d) Loop forever without halting

Three Levels of Abstraction

High-Level Description

An algorithm (like CS 330)

Implementation-Level Description

Describe (in English) the instructions for a TM

- How to move the head
- What to write on the tape

Low-Level Description

State diagram or formal specification

Example

Determine if a string $w \in \{0\}^*$ is in the language

$A = \{0^{2^n} \mid n \geq 0\}$

High-Level Description

Repeat the following forever:

- If there is exactly one 0 in w, accept
- If there is an odd (> 1) number of 0s in w, reject
- Delete half of the 0s in w

Example

Determine if a string $w \in \{0\}^*$ is in the language

$$A = \{0^{2^n} \mid n \geq 0\}$$

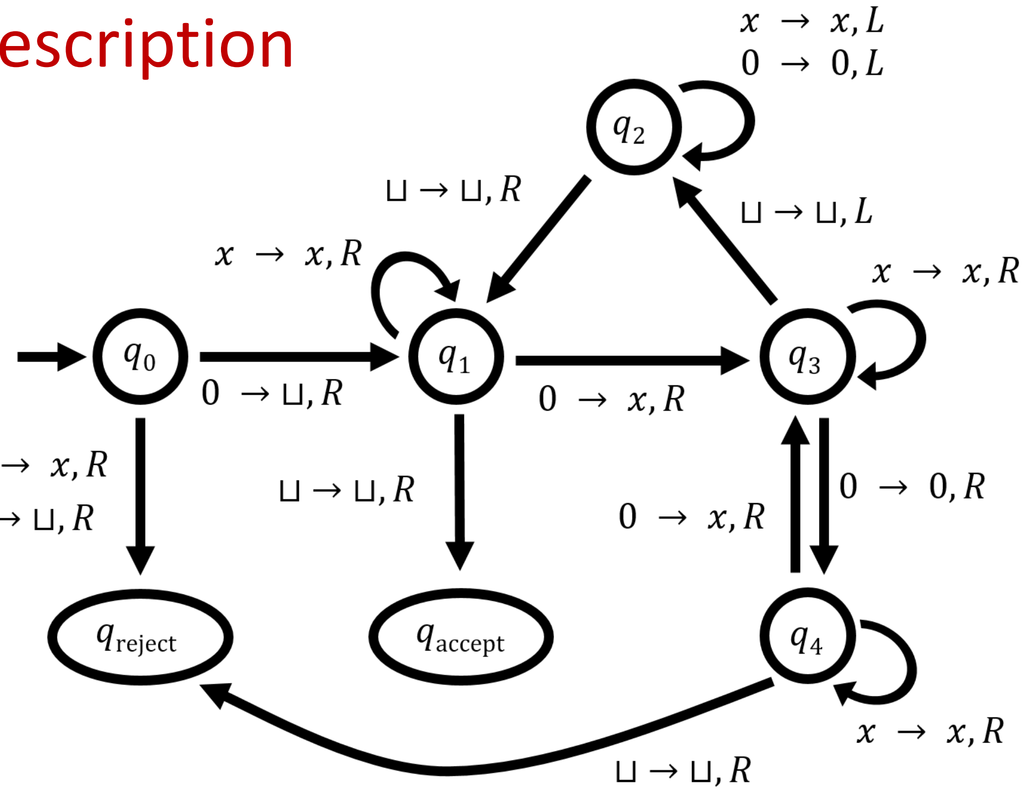
Implementation-Level Description

1. While moving the tape head left-to-right:
   a) Cross off every other 0
   b) If there is exactly one 0 when we reach the first blank symbol, accept
   c) If there is an odd (> 1) number of 0s when we reach first blank symbol, reject
2. Return the head to the left end of the tape
3. Go back to step 1

Example

Determine if a string $w \in A = \{0^{2^n} \mid n \geq 0\}$

Low-Level Description



Differences between TMs and Finite Automata

Formal Definition of a TM

A TM is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$

- Q is a finite set of states
- Σ is the input alphabet (does **not** include ⊔)
- Γ is the tape alphabet (contains ⊔ and Σ)
- δ is the transition function

  …more on this later
- $q_0 \in Q$ is the start state
- $q_{\text{accept}} \in Q$ is the accept state
- $q_{\text{reject}} \in Q$ is the reject state ($q_{\text{reject}} \neq q_{\text{accept}}$)

TM Transition Function

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

L means “move left” and R means “move right”

$\delta(p, a) = (q, b, R)$ means:

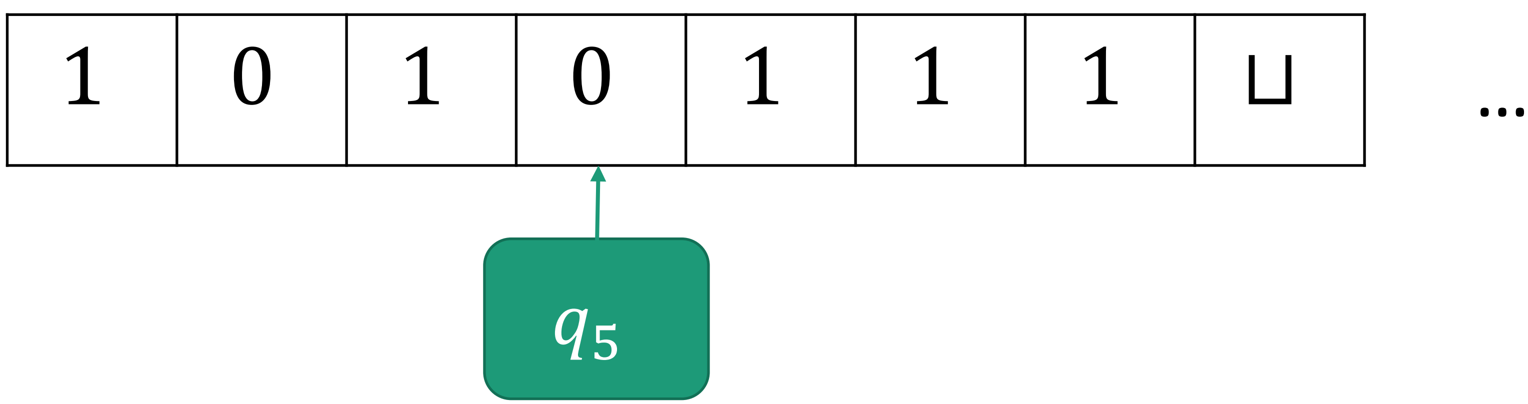
- Replace a with b in current cell
- Transition from state p to state q
- Move tape head right

$\delta(p, a) = (q, b, L)$ means:

- Replace a with b in current cell
- Transition from state p to state q
- Move tape head left UNLESS we are at left end of tape, in which case don’t move

Configuration of a TM

A string that captures the **state** of a TM together with the **contents of the tape**



Configuration of a TM: Formally

A configuration is a string uqv where $q \in Q$ and $u, v \in \Gamma^*$

- Tape contents = uv (followed by infinitely many blanks ⊔)
- Current state = q
- Tape head on first symbol of v

Example: $101q_5 0111$