

CSCI 361 Lecture 2: Countability and Automata

Shikha Singh

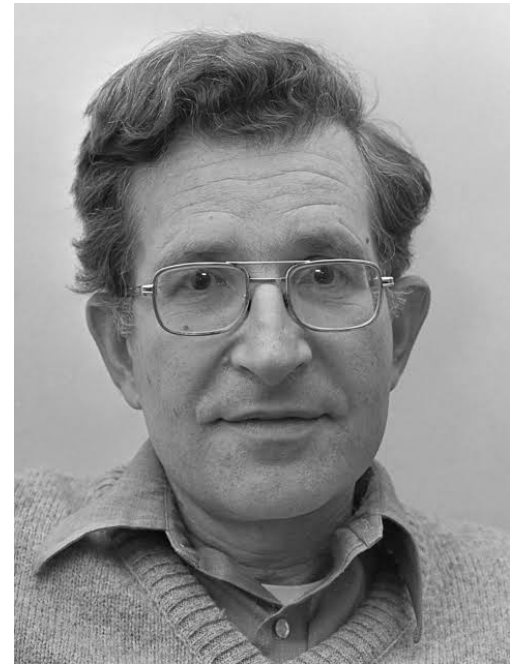
Announcements & Logistics

- **HW 1 released:** due Sept 17 (next Tuesday)
- Office hours and TA hours posted on course calendar
- Hand in reading assignment #1
- Pick up reading assignment #2, due at the start of Thurs lecture
- Looking ahead:
 - I will be out next Thurs (Sept 18, at Tapia conference in San Diego)
 - Lecture will be held as usual, covered by another CS prof
- **Questions?**

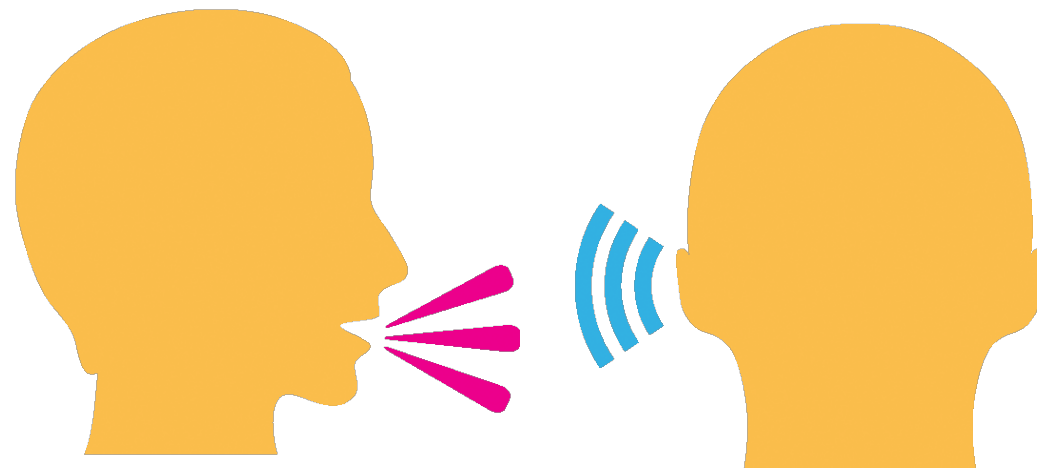
Last Time

- Introduced history and overview of theory of computation
- Discussed course logistics and reviewed syllabus
- Defined fundamentals of input/output representation
 - **Alphabet** Σ and set of all strings Σ^*
 - **Language**: any subset of strings from alphabet, i.e., $L \subseteq \Sigma^*$
 - **Length** of string s (# of symbols)
- All input/output in this course will be **binary strings**, that is, $\Sigma = \{0,1\}$
- Function problem vs decision problem:
 - A function problem is given by $f: \Sigma^* \rightarrow \Sigma^*$
 - A decision problem is given by $f: \Sigma^* \rightarrow \{0,1\}$

Influence of Chomsky



- We will study computation through the lens of languages
- Influence of linguist Chomsky on computation
- A **grammars generates** a language (akin to speaking)
 - Any string in the language can be generated using the rules of the grammar
- A **machine recognizes** a language (akin to listening)
 - If a given input string is in a language, the machine will "accept" (output true), otherwise "reject" (output false)



Background

Review Definitions

- **Sets:**

- Subsets, empty set, equivalence of sets, power set
- Union, intersection, complement, difference, etc
- Cartesian product of sets: $A \times B = \{(a, b) \mid a \in A, b \in B\}$

- **Relations:** subset of Cartesian product of sets

- Definition of a binary relation being reflexive, symmetric, transitive
- Equivalence relation

- **Function:** mapping from an input to an output

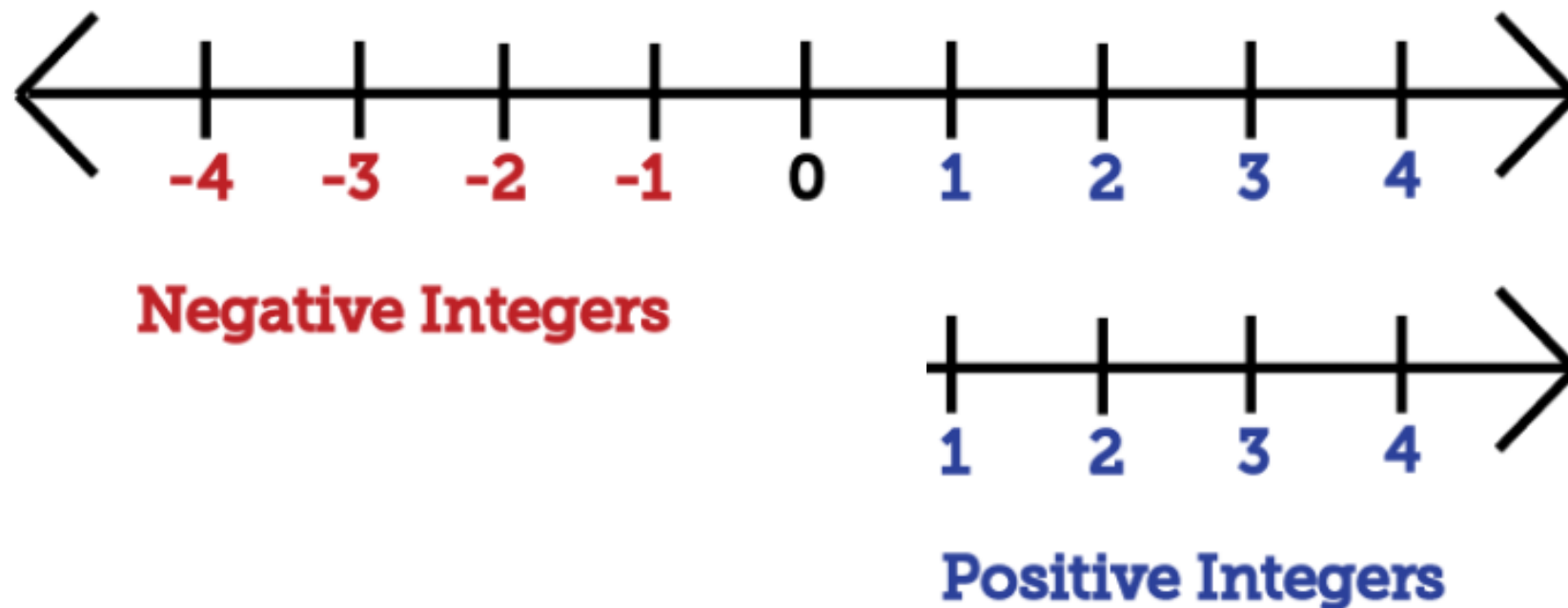
- $f : D \rightarrow R$ is a relation on $D \times R$ such that for all $x \in D$, there is exactly one $y \in R$ such that $(x, y) \in F$
- $(x, y) \in f$ written as $f(x) = y$
- Review definitions of one-to-one, onto and bijective functions

Cardinality

- Sets can be finite or infinite
- S is **finite** if there is a bijection function $f : S \rightarrow \{1, 2, \dots, n\}$ for some $n \in \mathbb{N}$
- Denote the size or cardinality as $|S| = n$
- S is **infinite** if no such bijection exists
- **Question.** Are all *infinite sets* the same size?
- S is a **countably infinite set** if there is a bijection function $f : S \rightarrow \mathbb{N}$
- S is a **countable** if it is finite or countably infinite
- S is a **uncountable** if it is not countable

Proof by Construction Example

- **Theorem:** The set of integers \mathbb{Z} is countable.
- Proof Outline:
 - Need to construct a bijective function f from \mathbb{Z} to \mathbb{N}
 - Show that it is one-one and onto
 - Any ideas?



Proof by Construction Example

- **Theorem:** The set of integers \mathbb{Z} is countable.
- Proof Outline:
 - Need to construct a bijective function f from \mathbb{Z} to \mathbb{N}
 - Show that it is one-one and onto

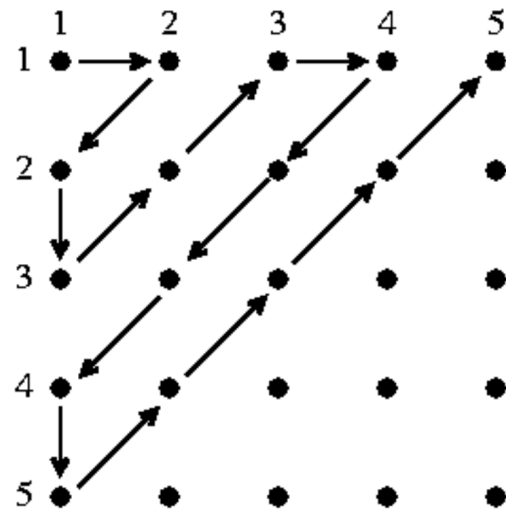
$$f(x) = \begin{cases} -2x & \text{if } x < 0 \\ 2x + 1 & \text{if } x \geq 0 \end{cases}$$

Countability

- Any set that can be "enumerated" using natural numbers is countable
 - Can talk about 1st item, 2nd item and so on
 - All items of the set appear in this on this list
- The set of all strings over a finite alphabet Σ is countable
 - Recall $\Sigma^* = \bigcup_{n \in \mathbb{N} \cup \{0\}} \Sigma^n$
 - Here Σ^n is a set of all strings of length exactly n over Σ
 - Each Σ^n is finite and contains $\ell = |\Sigma|^n$ elements, can list them in lexicographic order at indices $1, \dots, \ell$
- **Theorem:** Σ^* is countable.

Countability

- **Theorem:** $\mathbb{N} \times \mathbb{N}$ is countable.



- **Theorem:** The countable union of countable sets is countable.
 - Proof by picture (on the board)

Uncountability: Proof by Contradiction

- **Theorem:** The power set $\mathcal{P}(\mathbb{N})$ of natural numbers is **not** countable.
- **Proof Outline:**
 - Assume that $\mathcal{P}(\mathbb{N})$ is countable: a bijection to \mathbb{N} exists
 - That is, can enumerate all sets in $\mathcal{P}(\mathbb{N})$ as $S_1, S_2, \dots, S_k, \dots$
 - Reach an absurdity/contradiction $\Rightarrow \Leftarrow$
- Consider set of indices s.t.
 - (i is in corresponding S_i): $D = \{i \in \mathbb{N} \mid i \in S_i\}$
 - (i is not in corresponding S_i): $\bar{D} = \{i \in \mathbb{N} \mid i \notin S_i\}$
 - $\bar{D} = \mathbb{N} - D$, that is, \bar{D} is a valid subset of \mathbb{N} : $\exists k$ such that $S_k = \bar{D}$
 - What is the contradiction?

Cantor's Diagonalization Argument

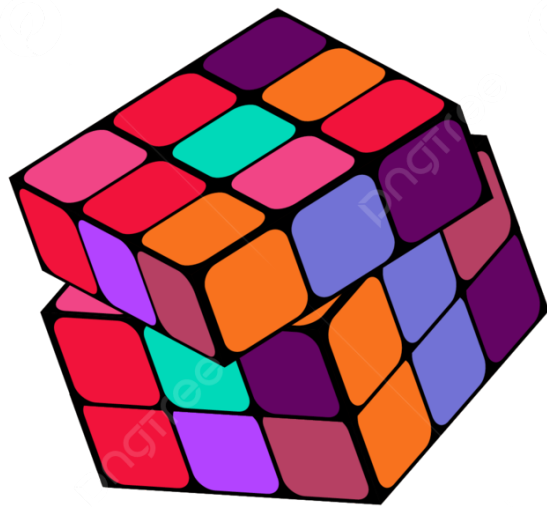
- This is an example of diagonalization argument
- Visualize where S_k appears on the diagonal
- ***Fun soviet version:***
 - <https://algorithmsoup.wordpress.com/2018/09/18/soviet-version-of-cantors-diagonalization-argument/>

Countability and Languages

- Recall, we said that a set A is **encodable** if there is a one-to-one function $\text{Enc} : A \rightarrow \Sigma^*$ for some finite alphabet Σ
 - Thus, the size of any encodable set is *at most* the size of Σ^*
- As we can encode any CS program/Turing machine using a finite alphabet, the set of such programs/Turing machines is countable
- However, the power set $\mathcal{P}(\Sigma^*)$ is uncountable
 - Similar argument as to power set of natural numbers
 - That is, there are uncountably many languages over any alphabet
- **Takeaway:** can only encode countable things, but uncountably many "decidable problems" to solve \implies existence of undecidable problems
 - What do these problems look like?

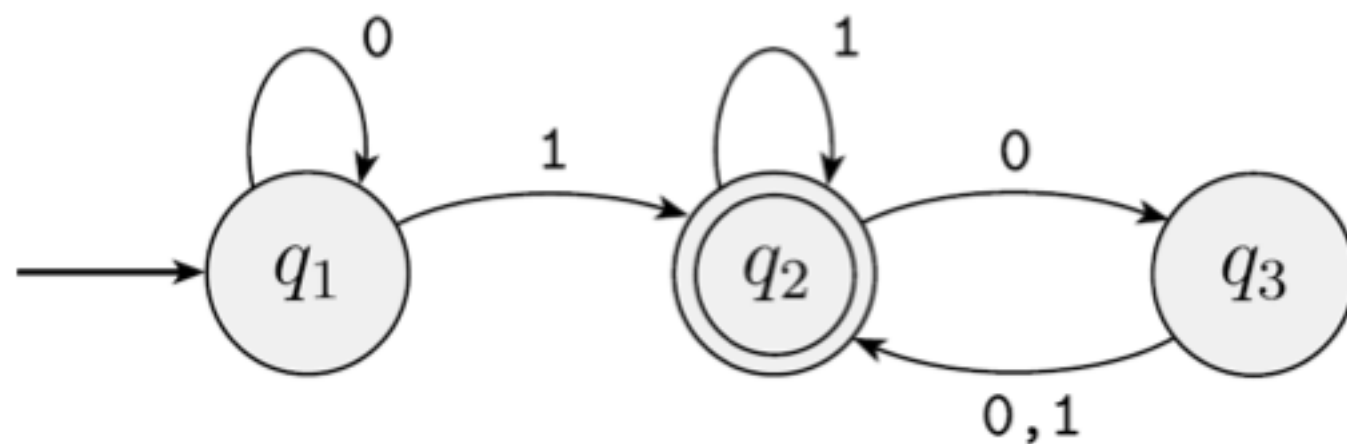
Finite State Automata

Simplest Form of Computation



Example: Deterministic Finite Automata

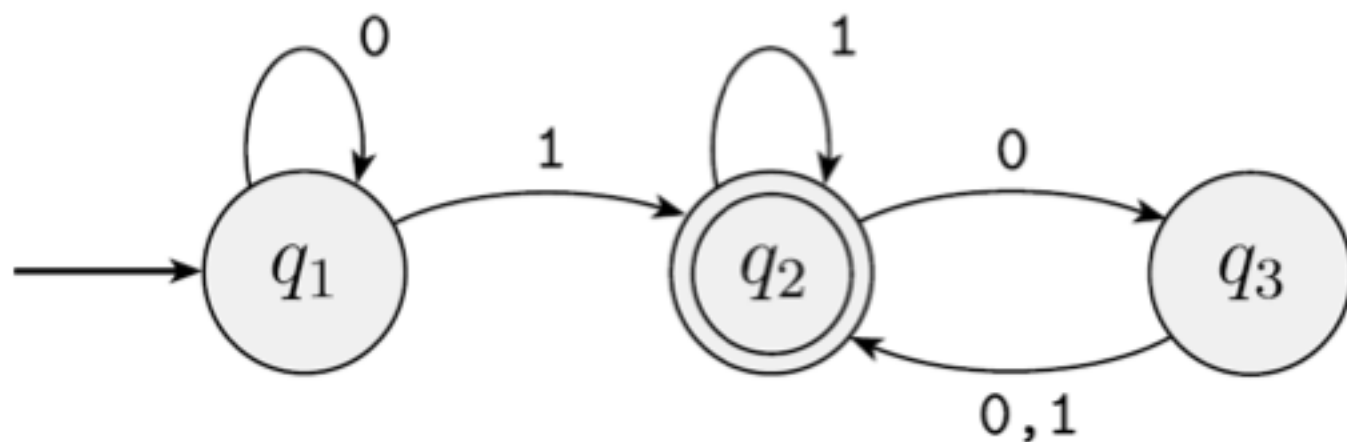
- We will study computation through the lens of languages
- A **machine recognizes** a language (akin to listening)
 - If a given input string is in a language, the machine will "accept" (output true), otherwise "reject" (output false)
- **Question.** What language is recognized by this machine?
 - Try some example strings



Definition of a Finite Automaton

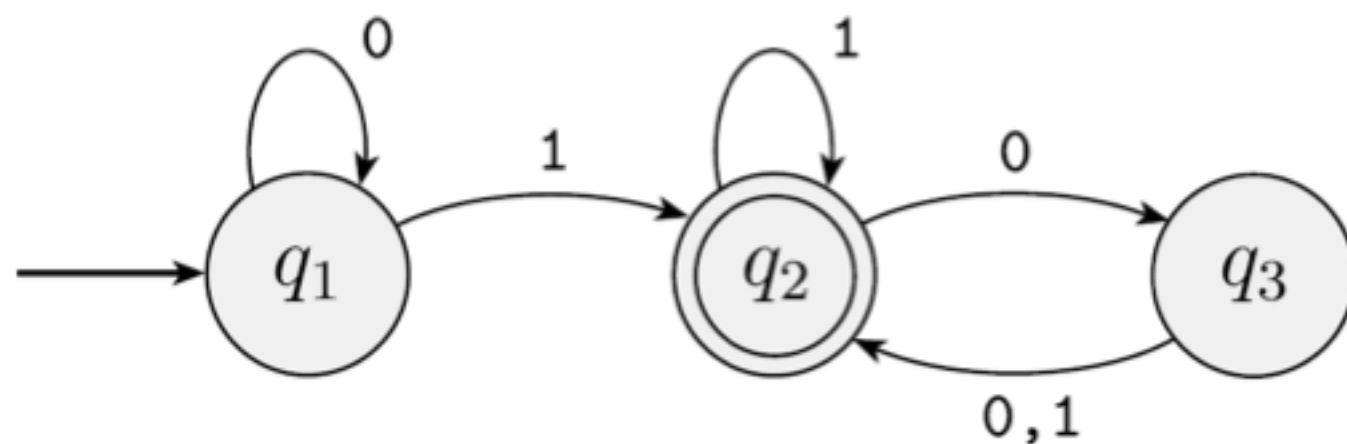
A finite automaton is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

- Q is a finite set called the states,
- Σ is a finite set called the alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function,
- $q_0 \in Q$ is the start state and $F \subseteq Q$ is the set of accept states.

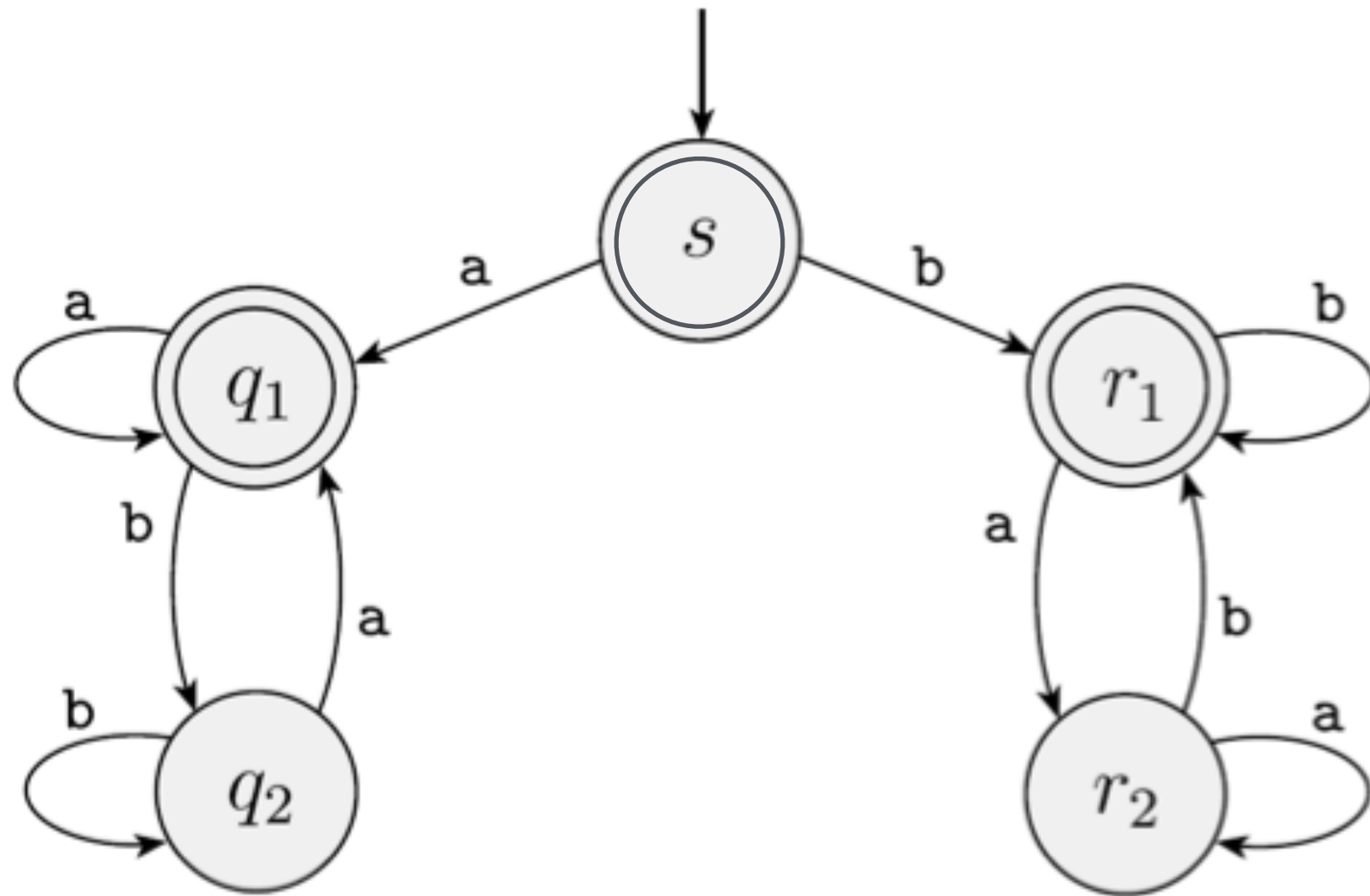


Language of a Machine

- The set of all strings accepted by a finite automaton M is called the language of machine M , and is written $L(M)$.
 - Say M **recognizes** language $L(M)$
- We will define M accepts w more formally
- Intuitive it is the strings on which it reaches an accepting state



What Language?



Automaton Computation

- Let $M = (Q, \Sigma, \delta, q_0, F)$ be a finite automaton and let $w = w_1w_2\cdots w_n$ be a string where each $w_i \in \Sigma$. Then M **accepts** w if there is a sequence of r_0, r_1, \dots, r_n in Q such that
 - $r_0 = q_0$
 - $\delta(r_i, w_{i+1}) = r_{i+1}$ for $i = 0, 1, \dots, n - 1$ and
 - $r_n \in F$



Extended Transition Function

- Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA
- Transition function $\delta : Q \times \Sigma \rightarrow Q$ is often extended to $\delta^* : Q \times \Sigma^* \rightarrow Q$ where $\delta^*(q, w)$ is defined as the state the DFA ends up in if it starts at q and reads the string w
- Alternate definition of M accepts $w \iff \delta^*(q_0, w) \in F$



Regular Languages

- A language is called a regular language if some finite automaton recognizes it.
- Seen examples of the following regular languages today:
- $L(M_1) = \{w \mid w \text{ contains at least one } 1 \text{ and an even number of zeroes follow the last } 1\}$
- $L(M_2) = \{w \mid w \in \{a, b\}^* \text{ that starts and ends with the same symbol}\}$

