Introduction to Finite Automata

Languages

Deterministic Finite Automata

Representations of Automata

Programs as functions

- We consider the task of writing a program which performs a task as computing a function.
- Example: Suppose we want to write a program which calculates gcd of two numbers.
  - We want to compute the function $gcd: \mathbb{N}\times\mathbb{N} \to \mathbb{N}$.

Programs as functions

- Example 2: Suppose we want to write a program which sorts an array of length 10.
  - We are computing the function $sort\colon \mathbb{N}^{10} \rightarrow \mathbb{N}^{10}$

Decision Problems

- A *decision problem* is a function with a one-bit output: “yes” or “no”.
- Examples
  - IsPrime(n): a function which determines whether input number n is a prime number
  - IsConnected(G): a function which determines input graph G is connected
- Question: Can any general function be expressed as a decision problem?

Question

- Can any general function be expressed as a decision problem?
- Yes!
  - Consider $f: A \rightarrow B$
  - Its 'decision problem version' is $f_d: A \times B \rightarrow \{Yes, No\}$
  - $f_d(a, b) = \begin{cases} Yes & \text{if } f(a) = b \\ No & \text{otherwise} \end{cases}$

Decision Problems: Alternative Specification

- A Decision problem f can be specified by
  - the set A of all possible inputs
  - the subset B ⊆ A of "yes" instances
- Examples:
  - For the IsPrime(n) function, $A = \mathbb{N}$, $B = Set\ of\ all\ prime\ numbers = \mathbb{P}$
  - For the IsConnected(G) function, $A = Set\ of\ all\ graphs$, $B = Set\ of\ all\ connected\ graphs$

Decision Problems as Language Inclusion

- Given $x \in A$, determining whether $f(x) = Yes$ is equivalent to determining whether $x \in B$.
- Lan … a sen… sentence is valid according to the rules (grammar) of the language.
  - A=Set of all sentences, B=Set of valid sentences.

Any decision problem can be expressed as a language inclusion problem!

From Computation to Languages

Arbitrary Task → Computing a function → Solving a decision problem

↓

Language Inclusion

Language Inclusion: Abstraction

- We will always consider the input space to be the set of finite-length strings over a fixed, finite alphabet.
- For $IsPrime(n)$, the input space is the set of natural numbers.
  - Each natural number can be seen as a string over alphabet $\{0,1,2,\ldots,9\}$.
- In general, any input can be encoded as a string.

Alphabets

- An *alphabet* is any finite set of symbols.
- Examples: English Alphabet, ASCII, {0,1,2,…,9} (*decimal alphabet*), {0,1} (*binary alphabet)*.
- We will use the Greek letter Σ to denote an alphabet.
  - Elements of an alphabet will be denoted by $a, b, c, \ldots$

Strings

- A *string* over Σ is any finite-length sequence of elements of Σ.
- Examples:
  - For Σ={a,b}, *ab,ba,aba,aa,abaab* are all distinct strings.
- We will use $x, y, z, \ldots$ to denote strings.

String length

- Length of a string is the number of symbols in the string.
  - For $\Sigma=\{0,1\}$, *011011* is string of length 6.
- There is a unique string of length 0, denoted by ϵ
  - Also called the *empty string* or *null string*

String Power Notation

- For $a \in \Sigma$, we write a^n for a string of a's repeated n times.
  - $a^5 = aaaaa, a^1 = a, a^0 = \epsilon$
- The set of all strings over alphabet Σ is denoted by Σ^*.
  - $\{0,1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$
  - $\{0,1,2, \ldots, 9\}^* = ???$
  - $\{0,1,2, \ldots, 9\}^* = \{\epsilon, 0{,}00{,}001, \ldots\} \neq \mathbb{N}$.
- By convention, we define $\emptyset^* = \{\epsilon\}$

String Concatenation

- Concatenation takes two strings x and y and makes a new string xy by putting them together.
  - $x\epsilon = x$
- x^n denotes the string obtained by concatenating n copies of x.
  - $x^0 = \epsilon$
  - $x^{n+1} = x^n x$

String Concatenation

- Examples
  - $(ab)^5 = ababababab$
  - $(ab)^1 = ab$
  - $(ab)^0 = \epsilon$
  - $(12)^2 = ???$
  - $(12)^2 = 1212$

Set Concatenation

- We will denote sets of strings (subsets of Σ^*) by symbols such as $A, B, C, \dots$
  - Also called *languages*.
- Given two sets A, B, the set concatenation AB is defined to be the set $\{xy \mid x \in A \; and \; y \in B\}$
  - Example: $\{a, b\}\{ab, ba\} = \{aab, aba, bab, bba\}$

Announcements

- Office hours on Wednesday 3-4 PM.
  - Online on the class link.
- TA Office hours and student assignment to TAs will be decided by the end of the week.

Questions

- What is $\emptyset\{ab, ba\}$?
  - It is $\emptyset$.
  - In general, for any set of strings A, $\emptyset A = \emptyset$.
- What is $\{\epsilon\}\{ab, ba\}$?
  - It is $\{ab, ba\}$.
  - In general, for any set of strings A, $\{\epsilon\}A = A$.

Powers of Set

- The powers A^n of set A are defined as follows:
  - $A^0 = \{\epsilon\}$
  - $A^{n+1} = A^n A$
- Example:
  - $\{ab, ba\}^0 = \{\epsilon\}$
  - $\{ab, ba\}^1 = \{ab, ba\}^{0+1} = \{\epsilon\}\{ab, ba\} = \{ab, ba\}$
  - $\{ab, ba\}^2 = \{ab, ba\}\{ab, ba\} = \{abab, abba, baab, baba\}$

Asterate of Set

- The asterate A^* of a set A is the union of all finite powers of A
  - $A^* = \bigcup_{n \geq 0} A^n = A^0 \cup A^1 \cup A^2 \cup A^3 \cup \cdots$
- Equivalently, A^* can also be defined as follows:
  - $A^* = \{x_1 x_2 \ldots x_n \mid n \geq 0 \; and \; x_i \in A, 1 \leq i \leq n\}$

Asterate and the set of all strings of an alphabet

- Recall that Σ^* denotes the set of all strings over an alphabet Σ.
  - Considering each symbol as a string, this matches the definition of Asterate of a set of strings.
  - Also explains why $\emptyset^* = \{\epsilon\}$.
- Note that $a \in \Sigma$ can either denote the element a or the string a (depends on the context).

Deterministic Finite Automata: Intuition

- A Model of a finite-memory machine.
  - Consists of *states* and *transitions*.
- State: Contains all the necessary information about the machine at a point of time.
  - Snapshot of the machine frozen in time.

Deterministic Finite Automata: Intuition

- Transition: Changes of state, which happens either spontaneously or in response to external inputs.
- Deterministic: state completely determines how the machine will evolve over time.

Deterministic Finite Automata: Formal Definition

- A Deterministic Finite Automata A is a 5-tuple, $\mathrm{A} = (Q, \Sigma, \delta, s, F)$
  - Q is a finite, non-empty set of *states*
  - Σ is the *input alphabet*
  - $\delta: Q \times \Sigma \rightarrow Q$ is the *transition function*
  - $s \in Q$ is the *start state*
  - $F \subseteq Q$ is the set of *accept or final states*.

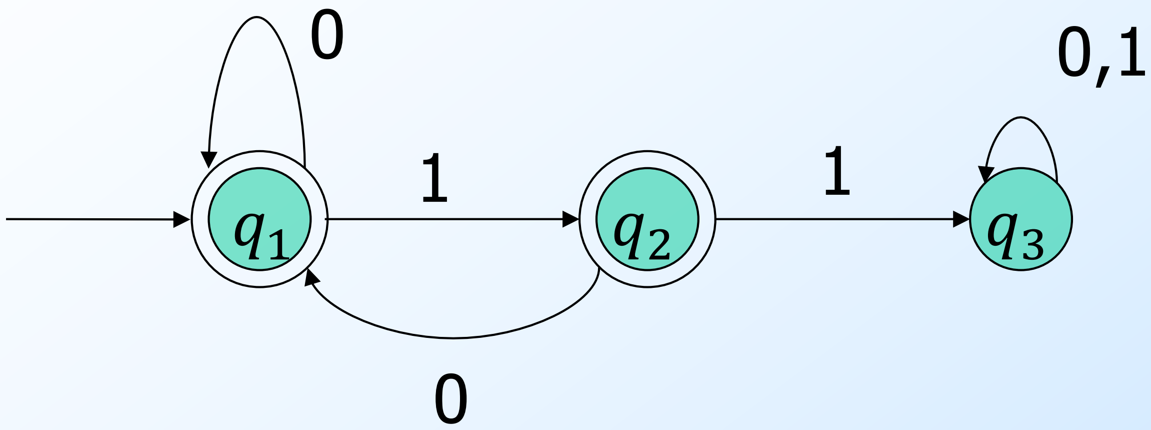
The Transition Function

- Takes two arguments: a state and an input symbol.
- $\delta(q, a)$ = the state that the DFA goes to when it is in state *q* and input *a* is received.

Graph Representation of DFA’s

- Nodes = states.
- Arcs represent transition function.
  - Arc from state p to state q labeled by all those input symbols that have transitions from p to q.
- The start state is indicated by an incoming arrow without any source.
- Final states indicated by double circles.

Example: Graph of a DFA



Formally, the DFA is given by $(Q, \Sigma, \delta, q_1, F)$, where

$$Q = \{q_1, q_2, q_3\}$$

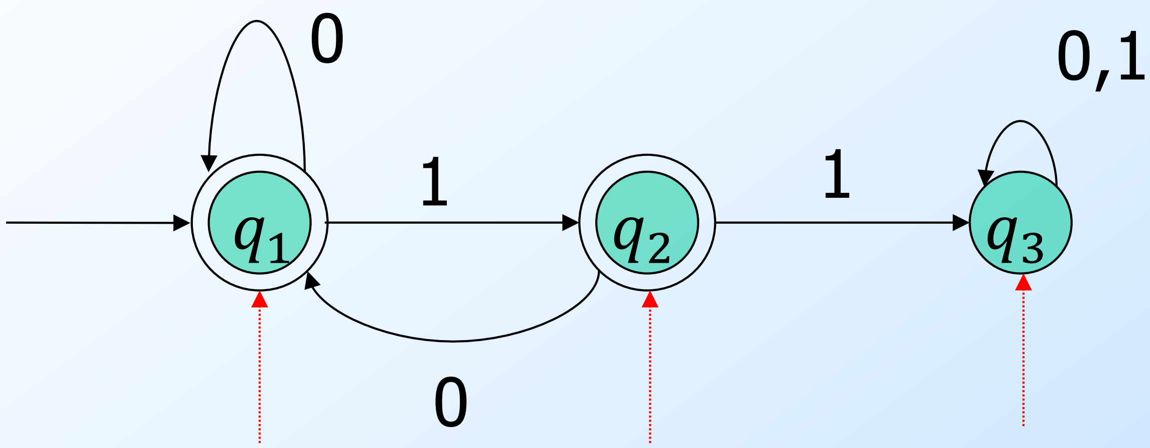
$$\Sigma = \{0,1\}$$

$$F = \{q_1, q_2\}$$

$$\delta(q_1, 0) = q_1, \delta(q_1, 1) = q_2$$
$$\delta(q_2, 0) = q_1, \delta(q_2, 1) = q_3$$
$$\delta(q_3, 0) = q_3, \delta(q_3, 1) = q_3$$

Example: Graph of a DFA

Accepts all strings without two consecutive 1's.



Previous string OK, does not end in 1.

Previous String OK, ends in a single 1.

Consecutive 1's have been seen.

Alternative Representation: Transition Table

Final states starred

Arrow for start state

Columns = input symbols

	0	1
$\rightarrow * \; q_1$	q_1	q_2
$* \; q_2$	q_1	q_3
q_3	q_3	q_3

Rows = states

Extended Transition Function

- We describe the effect of a string on a DFA by extending δ to a state and a string.
- Inductive Definition:
  - $\hat{\delta}(q, \epsilon) = q$
  - $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$
  - Note that w is a string; a is an element of the alphabet.

Extended δ: Intuition

- Extended δ is computed for state q and string $a_1a_2\ldots a_n$ by following a path in the transition graph, starting at q and selecting the arcs with labels $a_1, a_2,\ldots,a_n$ in turn.

Example: Extended Delta

	0	1
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_3	q_3

$$\begin{aligned}
\hat{\delta}(q_1, 011) &= \delta\big(\hat{\delta}(q_1, 01), 1\big) \\
&= \delta\big(\delta\big(\hat{\delta}(q_1, 0), 1\big), 1\big) \\
&= \delta\big(\delta\big(\delta\big(\hat{\delta}(q_1, \epsilon), 0\big), 1\big), 1\big) \\
&= \delta(\delta(\delta(q_1, 0), 1), 1) \\
&= \delta(\delta(q_1, 1), 1) \\
&= \delta(q_2, 1) \\
&= q_3
\end{aligned}$$