CSC236H
Introduction to the Theory of Computation

Bahar Aameri

Fall 2019 – Week 4

Language Recognition

Language Recognition Problem:

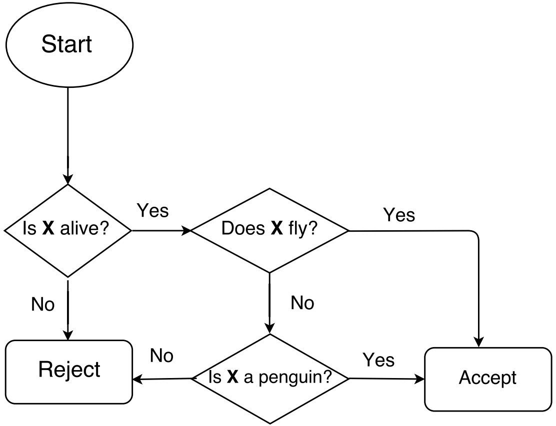
- Given language L and string s, does s belong to L?

Many problems can be reduced to a recognition problem about formal languages.
Examples:

- Syntax Checking for Mathematical Formulas and Parsing.
- Lexical Analysis in Program Compilation.
- Pattern Recognition.

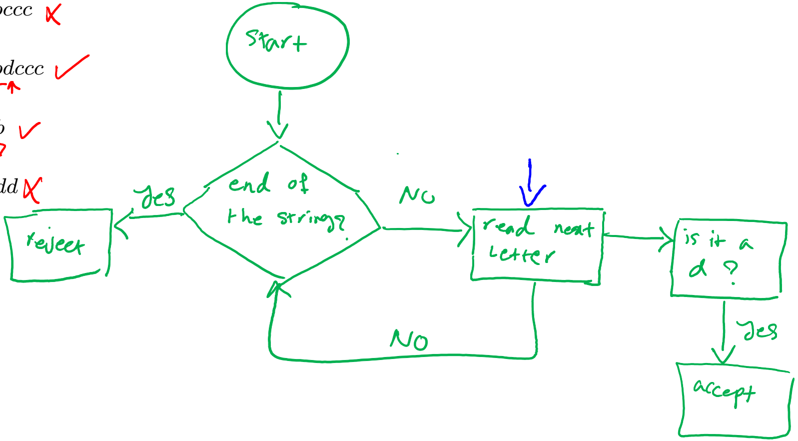
Language Recognition Problem – An analogy

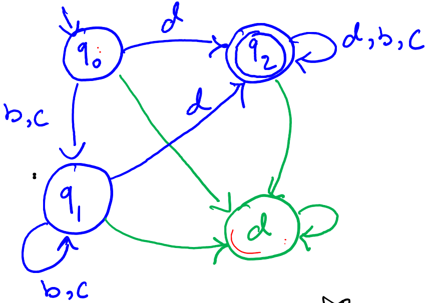
Bird Recognition: Is the given object, **X**, a bird?

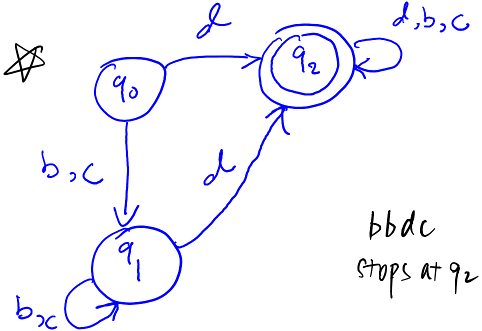


Let $L = \{s \in \{b, c, d\}^* : s$ includes at least one $d\}$.
Among the following strings, determine those that are members of L.

- $bbbccc$
- $bbbdccc$
- dbb
- $b1dd$







bbdc

stops at q_2. accept.

Deterministic Finite State Automata (DFA)

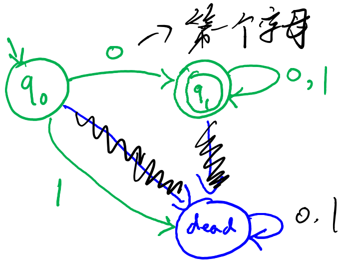
- Very informally, a Deterministic Finite State Automaton (**DFSA** or **DFA**) is a mathematical model of a machine which takes an **input string** x, and accepts or rejects it.
- A DFA consists of
  - a set of states;
  - a set of rules (called transition rules) for transition between states based on the input.
  - A designated initial state.
  - A set of designated accepting states.

A language is **regular** iff a Finite State Automata (**FSA**) can be used to recognize its strings.

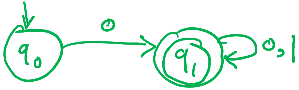
Deterministic Finite State Automata (DFA)

- A DFA is started in its **initial state**;
- Reads a string **one letter** at a time, from **left to right**.
- Depending on the present state of the DFA and the symbol it read, the DFA enters a **new state**.
- The DFA stops when it has processed the **entire input string** in this manner:
  When stops, if it is in an **accepting state**, it accepts the input;
  otherwise, it rejects the input.

Let $L = \{0s : s \in \{0,1\}^*\}$.
Give a DFA which **only accepts** strings in L.



D:



$D = \langle Q, \Sigma, \delta, s, F \rangle$

$\Sigma = \{0, 1\}$ $\quad s = q_0$

$Q = \{q_0, q_1\}$ $\quad F = \{q_1\}$

$\delta(q_0, 0) = q_1$ $\quad$ $\delta(q_1, 1) = q_1$

$\delta(q_1, 0) = q_1$

State \ input	0	1
q_0	q_1	—
q_1	q_1	q_1

$\delta^*(q_0, w) = q_0$ iff $w = \epsilon$

$\delta^*(q_0, w) = q_1$ iff $w = 0S$

where $S \in \{0,1\}^*$

not on lec

A **Deterministic Finite State Automaton (DFA)** $\mathcal{D}$ is a quintuple $\mathcal{D} = \langle Q, \Sigma, \delta, s, F \rangle$ where:

- Q is the **set of states** in $\mathcal{D}$;
- Σ is the **alphabet** of symbols used by $\mathcal{D}$;
- $\delta : Q \times \Sigma \to Q$ is the **transition function**;
- $s \in Q$ is the **initial state** of $\mathcal{D}$;
- $F \subseteq Q$ is the set of **accepting states** of $\mathcal{D}$.

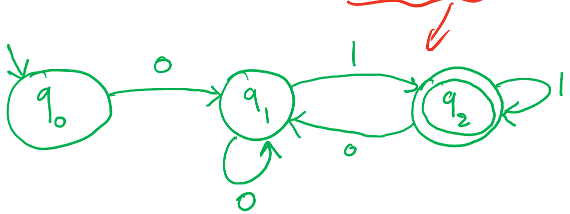
Important Properties of DFA’s

- In a DFA, at a particular state, there is exactly one **transition rule** for each symbol in the alphabet.
- **Dead states** are usually not included in the description of the DFA’s; if there’s **no transition rule** (arrow) for a letter at a state, by convention it’s assumed that it take the DFA to a **dead state**.
- **Inputs** to DFAs can be of any length.
- DFA’s cannot go back and reread previous letters.
- DFA’s have a finite amount of memory, since they have a finite number of states.

Let $L = \{0s1 : s \in \{0,1\}^*\}$.

Give a DFA which **only accepts** strings in L.

Present your DFA both by **a drawing** and by using **formal notation**.



symbolic notation: $D = \langle Q, \Sigma, \delta, q_0, F \rangle$

$\Sigma = \{0, 1\}$ $\quad Q = \{q_0, q_1, q_2\}$ $\quad F = \{q_2\}$

δ:

	0	1
q_0	q_1	—
q_1	q_1	q_2
q_2	q_1	q_2

$\delta^*(q_0, w) = q_0$ iff $w = \epsilon$

$\delta^*(q_0, w) = q_1$ iff $w = 0S0$ where $S \in \{0,1\}^*$ $\boxed{\text{start with 0, end with 0}}$

$w = 0$

$\delta^*(q_0, w) = q_2$ iff $w = 0S1$ where $S \in \{0,1\}^*$

DFA's – Extended Transition Function

$\delta(q, c)$

- **Transition Function**: Takes a state q and a **letter** c.
  Denotes the state that $\mathcal{D}$ is taken to when $\mathcal{D}$ is at q and reads c

- **Extended Transition Function**: Takes a state q and a **string** w. Denotes the state that $\mathcal{D}$ is taken to (when $\mathcal{D}$ is at q and reads w.)

$$\delta^*(q, w) = q_1$$

$$q \xrightarrow{w} q_1$$

DFA's – Extended Transition Function

Let Σ^* be the smallest set such that:

- $\epsilon \in \Sigma^*$. B.R.
- If $w \in \Sigma^*$ and $s \in \Sigma$ then $ws \in \Sigma^*$. RL

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

$\epsilon \in \Sigma^*$

$\epsilon . 0 = 0 \in \Sigma^*$

$\epsilon . 1 = 1 \in \Sigma^*$

Let $\delta : Q \times \Sigma \to Q$ be the transition function of a DFA $\mathcal{D}$.
The **extended transition function** of the DFA is the function

$$\delta^* : Q \times \Sigma^* \to Q$$

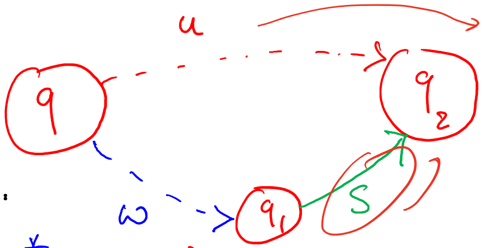
defined recursively:

- $\delta^*(q, \epsilon) = q$. B.R.
- For some $w \in \Sigma^*$ and $s \in \Sigma$,

R.L.

$$\delta^*(q, ws) = \delta(\delta^*(q, w), s).$$

$\boxed{u = ws}$



$\delta^*(q,u) = q_2$ (3)

$= \delta^*(q, ws) =$

$q_1 = \delta^*(q,w)$ (1)

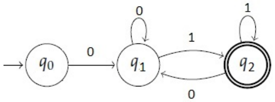
$\delta(q_1, s) = q_2$ (2)

(1), (2) $\Rightarrow \delta(\delta^*(q,w), s) = q_2$

(3) $\Rightarrow \delta^*(q,u) = \delta^*(q,ws) = \delta(\delta^*(q,w), s)$

Let $L = \{0s1 : s \in \{0,1\}^*\}$.
Give a DFA which **only accepts** strings in L.

$w_1 = 001$, $\delta^*(q_0, w_1)$?



$w_1 = \underbrace{00}_{w}\underbrace{1}_{a}$

$$\delta^*(q_0, 001) = \delta(\underbrace{\delta^*(q_0, 00)}, 1) = \delta(q_1, 1) = q_2$$

$$\delta^*(q_0, 00) = \delta(\underbrace{\delta^*(q_0, 0)}, 0) = \delta(q_1, 0) = q_1$$

$$\delta^*(q_0, 0) = \delta(\underbrace{\delta^*(q_0, \epsilon)}, 0) = \delta(q_0, 0) = q_1$$

$0 = \epsilon \cdot 0$

$$\delta^*(q_0, \epsilon) = q_0$$

- If $\delta^*(q, w) = q'$, we say that w takes the automaton $\mathcal{D}$ from q to q'.
- A string $w \in \Sigma^*$ is **accepted** by $\mathcal{D}$, if and only if w takes the automaton from the initial state q_0 to an accepting state.

$$\delta^*(q_0, w) \in F.$$

- The language **accepted** (or recognised) by a DFA $\mathcal{D}$, denoted by $\mathcal{L}(\mathcal{D})$, is the set of all strings accepted by $\mathcal{D}$.

$$\boxed{\mathcal{L}(\mathcal{D})} = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$$

L, D

$$\boxed{\mathcal{L}(D) = L}$$

State Invariants

- An **invariant** for q is a statement that **characterizes** all the strings that take the DFA from the initial state to q.
  Formally, an **invariant** for a state q is a predicate I over the domain Σ^* such that for every string $w \in \Sigma^*$,

$\delta^*(q_0, w) = q_1$

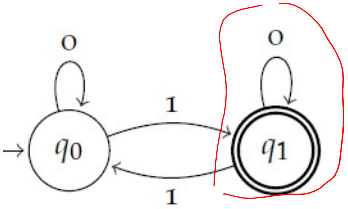
$\delta^*(q_0, w) = q_2$

$$\delta^*(q_0, w) = q \quad \text{iff} \quad I(w) \text{ holds.}$$

- The state invariants for a DFA should be **mutually exclusive**.
  No string should satisfy two different state invariants.

- The state invariants for a DFA should be **exhaustive**. 穷尽
  Every string in Σ^*, including ϵ, should satisfy one of the state invariants.

Describe the language that the following DFA accepts



$\delta^*(q_0, w) = q_0$ iff w

$\delta^*(q_0, w) = q_1$ iff $w =$

$\delta^*(q_0, \omega) = q_0$ iff ω has even number of 1's, $\omega \in \{0,1\}^*$

$\delta^*(q_0, \omega) = q_1$ iff ω has odd number of 1's, $\omega \in \{0,1\}^*$

(accepted)

$\mathcal{L}(D) = \{ w \in \{0,1\}^* \mid w \text{ has odd number of 1's} \}$

$w \in \mathcal{L}(D)$ iff $\delta^*(q_0, w) \in F$

- Identify required states (**State Invariants**).
- For each state, determine how each input symbol changes the state.
- The transition rules must preserve the state invariants

$\mathcal{L}(D)$

- **Example:** $L = \{w \in \{0,1\}^* : w$ has even number of 0's, even number of 1's$\}$.

$\delta^*(q_0, w) = q_0$ iff $w \in \{0,1\}^*$ w has even number of 0's, even number of 1's

$\delta^*(q_0, w) = q_1$ iff $w \in \{0,1\}^*$ has odd 0's and even 1's

$\delta^*(q_0, w) = q_2$ iff $w \in \{0,1\}^*$ has even 0's and odd 1's

$\delta^*(q_0, w) = q_3$ iff $w \in \{0,1\}^*$ has odd 0's and odd 1's