Section: Finite Automata

Deterministic Finite Accepter (or Automata)

A DFA = (Q, Σ, δ, q₀, F)

where
- Q is finite set of states
- Σ is tape (input) alphabet
- q₀ is initial state
- F ⊆ Q is set of final states.
- δ: Q × Σ → Q

Q, Σ, F finite sets

CompSci 334 Spring 2023  1/17/23  1/19/23  1/24/23
Example: DFA that accepts even binary numbers.

Transition Diagram:

\[ M = (Q, \Sigma, \delta, q_0, F) = \{ q_0, q_1, q_2, q_3, q_4 \} \]

Tabular Format

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>q0</td>
<td>q1</td>
<td>q0</td>
</tr>
<tr>
<td>q1</td>
<td>q1</td>
<td>q0</td>
</tr>
</tbody>
</table>

Example of a move: \( \delta(q_0, 1) = \)
Algorithm for DFA:

Start in start state with input on tape
q = current state
s = current symbol on tape
while (s != blank) do
    q = δ(q,s)
    s = next symbol to the right on tape
if q ∈ F then accept

Example of a trace: 11010
Pictorial Example of a trace for 100:

1)  

   1 0 0

   q0 ← q1

2)  

   1 0 0

   q0 ← q1

3)  

   1 0 0

   q0 ← q1

4)  

   1 0 0

   q0 ← q1
Definition:
\[ \delta^*(q, \lambda) = q \]
\[ \delta^*(q, wa) = \delta(\delta^*(q, w), a) \]

Definition The language accepted by a DFA \( M = (Q, \Sigma, \delta, q_0, F) \) is set of all strings on \( \Sigma \) accepted by \( M \). Formally,
\[ L(M) = \{ w \in \Sigma^* \mid \delta^*(q_0, w) \in F \} \]
Trap State

Example: $L(M) = \{b^a a^1 n > 0 \mid n \geq 2 \}$
Example:

\[ L = \{ w \in \Sigma^* \mid \text{w has an even number of a's and an even number of b's} \} \]
Example: DFA that accepts even binary numbers that have an even number of 1’s.

Two solutions:
Definition A language $L$ is regular iff there exists DFA $M$ s.t. $L = L(M)$. 
Chapter 2.2
Nondeterministic Finite Automata (or Accepter)
Definition
An NFA = (Q, Σ, δ, q₀, F)
where
Q is finite set of states
Σ is tape (input) alphabet
q₀ is initial state
F ⊆ Q is set of final states.
δ: Q × (Σ ∪ {λ}) → 2^Q
Example

Note: In this example $\delta(q_0, a) = \{ q_0, q_2, q_3 \}$

$L = \{ aa^2 v \Sigma^* a b^n b^m | n \geq 0 \}$
Example

\[ L = \{(ab)^n \mid n > 0\} \cup \{a^n b \mid n > 0\} \]
Definition $q_j \in \delta^*(q_i, w)$ if and only if there is a walk from $q_i$ to $q_j$ labeled $w$.

Example From previous example:

\[ \delta^*(q_0, ab) = q_1, q_4, q_6 \]
\[ \delta^*(q_0, aba) = q_5 \]

Definition: For an NFA $M$, 
\[ L(M) = \{ w \in \Sigma^* \mid \delta^*(q_0, w) \cap F \neq \emptyset \} \]
2.3 NFA vs. DFA: Which is more powerful? *Neither*

Example:
Theorem Given an NFA $M_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$, then there exists a DFA $M_D = (Q_D, \Sigma, \delta_D, q_0, F_D)$ such that $L(M_N) = L(M_D)$.

Proof:

We need to define $M_D$ based on $M_N$.

$Q_D = 2^{Q_N}$

$F_D = \{ Q \subseteq Q_D \mid \exists g_i \in Q \text{ with } g_i \in F_N \}$

$\delta_D : Q_D \times \Sigma \rightarrow Q_D$
Algorithm to construct $M_D$

1. start state is $\{q_0\} \cup \text{closure}(q_0)$

2. While can add an edge
   
   (a) Choose a state $A = \{q_i, q_j, \ldots q_k\}$ with missing edge for $a \in \Sigma$

   (b) Compute $B = \delta^*(q_i, a) \cup \delta^*(q_j, a) \cup \ldots \cup \delta^*(q_k, a)$

   (c) Add state $B$ if it doesn’t exist

   (d) add edge from $A$ to $B$ with label $a$

3. Identify final states

4. if $\lambda \in L(M_N)$ then make the start state final.
Example:
Properties and Proving - Problem 1

Consider the property
Replace_one_a_with_b or R1awb for short. If L is a regular, prove
R1awb(L) is regular.

The property R1awb applied to a
language L replaces one a in each
string with a b. If a string does not
have an a, then the string is not in
R1awb(L).

Example 1: Consider L={aaab, bbaa}
R1awb(L)={åbaa, abab, aaaa, bbaa, bbaa}

Example 2: Consider Σ = {a, b}, L =
{w ∈ Σ* | w has an even number of a’s
and an even number of b’s}
R1awb(L)={w ∈ Σ* / w has an odd number of
a’s or odd number of b’s}

Proof:
M = \( (Q, \Sigma, \delta, q_0, F) \) for L

M' is a DFA for L

M'' is an NFA for R\text{amb}(L)

For every a

arc

\( s(p; a) = q \)

\( s'' \) add

\( s(p; b) = q' \)

\( Q' = (Q \cup Q') \)

\( Q'' = (Q \cup Q') \)

\( F'' = F' \)

STOPPED HERE
Properties and Proving - Problem 2

Consider the property
Truncate_all_preceeding_b’s or
TruncPreb for short. If L is a regular,
prove TruncPreb(L) is regular.

The property TruncPreb applied to a
language L removes all preceeding b’s
in each string. If a string does not
have an preceeding b, then the string
is the same in TruncPreb(L).

Example 1: Consider L={aaab, bbaa}
TruncPreb(L)={aaab, aa2}

Example 2: Consider L =
{(bb)\textsuperscript{n} | n > 0}
TruncPreb(L)= \{a(bb)\textsuperscript{n} | n \geq 0\}

Proof:

\[ \exists \text{ DFA } M \text{ for } L \]
\[ M = (Q, \ldots) \]
\[ Q = Q \cup Q' \]
\[ F = F' \]

In \( M \)
- replace all \( b \) arcs with \( \epsilon \)
- remove all \( a \) arcs
- for \( S(p_a) = q \)
- add \( S(p_a)' = q' \)
all states in \( M' \) are primed
Minimizing Number of states in DFA

Why?

Algorithm

- Identify states that are indistinguishable
  These states form a new state

Definition Two states $p$ and $q$ are indistinguishable if for all $w \in \Sigma^*$

\[
\delta^*(q, w) \in F \Rightarrow \delta^*(p, w) \in F \\
\delta^*(p, w) \not\in F \Rightarrow \delta^*(q, w) \not\in F
\]

Definition Two states $p$ and $q$ are distinguishable if $\exists$ $w \in \Sigma^*$ s.t.

\[
\delta^*(q, w) \in F \Rightarrow \delta^*(p, w) \not\in F \text{ OR } \\
\delta^*(q, w) \not\in F \Rightarrow \delta^*(p, w) \in F
\]
Example:
Example: