CPS 140 - Mathematical Foundations of CS
Dr. Susan Rodger
Section: Finite Automata (Ch. 2) (handout)

Deterministic Finite Accepter (or Automata)
$\mathrm{A} \mathrm{DFA}=\left(\mathrm{Q}, \Sigma, \delta, q_{0}, \mathrm{~F}\right)$

where
Q is finite set of states
$\Sigma$ is tape (input) alphabet
$q_{0}$ is initial state
$\mathrm{F} \subseteq \mathrm{Q}$ is set of final states.
$\delta: \mathrm{Q} \times \Sigma \rightarrow \mathrm{Q}$
Example: Create a DFA that accepts even binary numbers.

Transition Diagram:
$\mathrm{M}=\left(\mathrm{Q}, \Sigma, \delta, q_{0}, \mathrm{~F}\right)=$
Tabular Format

|  | 0 | 1 |
| :--- | :--- | :--- |
| q0 |  |  |
| q1 |  |  |

Example of a move: $\delta(q 0,1)=$

## Algorithm for DFA:

Start in start state with input on tape
$\mathrm{q}=$ current state
$\mathrm{s}=$ current symbol on tape
while ( $\mathrm{s}!=$ blank) do
$\mathrm{q}=\delta(\mathrm{q}, \mathrm{s})$
$\mathrm{s}=$ next symbol to the right on tape
if $q \in F$ then accept

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


## Definition:

$\delta^{*}(q, \lambda)=q$
$\delta^{*}(q, w a)=\delta\left(\delta^{*}(q, w), a\right)$
Definition The language accepted by a $\mathrm{DFA} \mathrm{M}=\left(\mathrm{Q}, \Sigma, \delta, q_{0}, \mathrm{~F}\right)$ is set of all strings on $\Sigma$ accepted by M. Formally,
$\mathrm{L}(\mathrm{M})=\left\{w \in \Sigma^{*} \mid \delta^{*}\left(q_{0}, w\right) \in F\right\}$

## Trap State

Example: $\mathrm{L}(\mathrm{M})=$


You don't need to show trap states! Any arc not shown will by default go to a trap state.

## Example:

$\mathrm{L}=\left\{w \in \Sigma^{*} \mid \mathrm{w}\right.$ has an even number of a's and an even number of b's $\}$

Example: Create a DFA that accepts even binary numbers that have an even number of 1's.

Definition A language is regular iff there exists DFA M s.t. $L=L(M)$.

## Chapter 2.2

Nondeterministic Finite Automata (or Accepter)

## Definition

$\mathrm{An} \mathrm{NFA}=\left(\mathrm{Q}, \Sigma, \delta, q_{0}, \mathrm{~F}\right)$
where
Q is finite set of states
$\Sigma$ is tape (input) alphabet $q_{0}$ is initial state $\mathrm{F} \subseteq \mathrm{Q}$ is set of final states. $\delta: \mathrm{Q} \times(\Sigma \cup\{\lambda\}) \rightarrow 2^{Q}$

## Example



Note: In this example $\delta\left(q_{0}, a\right)=$
$\mathrm{L}=$

## Example

$\mathrm{L}=\left\{(a b)^{n} \mid n>0\right\} \cup\left\{a^{n} b \mid n>0\right\}$

Definition $q_{j} \in \delta^{*}\left(q_{i}, w\right)$ if and only if there is a walk from $q_{i}$ to $q_{j}$ labeled $w$.
Example From previous example:
$\delta^{*}\left(q_{0}, a b\right)=$
$\delta^{*}\left(q_{0}, a b a\right)=$
Definition: For an NFA M, L(M) $=\left\{w \in \Sigma^{*} \mid \delta^{*}\left(q_{0}, w\right) \cap F \neq \emptyset\right\}$
The language accepted by nfa M is all strings $w$ such that there exists a walk labeled w from the start state to final state.

### 2.3 NFA vs. DFA: Which is more powerful?

## Example:



Theorem Given an NFA $M_{N}=\left(Q_{N}, \Sigma, \delta_{N}, q_{0}, F_{N}\right)$, then there exists a DFA $M_{D}=\left(Q_{D}, \Sigma, \delta_{D}, q_{0}, F_{D}\right)$ such that $L\left(M_{N}\right)=L\left(M_{D}\right)$.

Proof:
We need to define $M_{D}$ based on $M_{N}$.
$Q_{D}=$
$F_{D}=$
$\delta_{D}:$

## Algorithm to construct $M_{D}$

1. start state is $\left\{q_{0}\right\} \cup \operatorname{closure}\left(q_{0}\right)$
2. While can add an edge
(a) Choose a state $\mathrm{A}=\left\{q_{i}, q_{j}, \ldots q_{k}\right\}$ with missing edge for $a \in \Sigma$
(b) Compute $\mathrm{B}=\delta^{*}\left(q_{i}, a\right) \cup \delta^{*}\left(q_{j}, a\right) \cup \ldots \cup \delta^{*}\left(q_{k}, a\right)$
(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 $\lambda \in L\left(M_{N}\right)$ 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 $\mathrm{L}=\{a a a b, b b a a\}$
R1awb $(\mathrm{L})=$
Example 2: Consider $\Sigma=\{a, b\}, \mathrm{L}=\left\{w \in \Sigma^{*} \mid\right.$ whas an even number of a's and an even number of b's $\}$
R1awb $(\mathrm{L})=$
Proof:

## 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 $\mathrm{L}=\{a a a b, b b a a\}$
TruncPreb $(L)=$
Example 2: Consider $L=\left\{(b b a)^{n} \mid n>0\right\}$
TruncPreb $(\mathrm{L})=$
Proof:

## 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 indistinquishable if for all $w \in \Sigma^{*}$

$$
\begin{aligned}
& \delta^{*}(q, w) \in F \Rightarrow \delta^{*}(p, w) \in F \\
& \delta^{*}(p, w) \notin F \Rightarrow \delta^{*}(q, w) \notin F
\end{aligned}
$$

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

$$
\begin{aligned}
& \delta^{*}(q, w) \in F \Rightarrow \delta^{*}(p, w) \notin F \mathrm{OR} \\
& \delta^{*}(q, w) \notin F \Rightarrow \delta^{*}(p, w) \in F
\end{aligned}
$$

Example:


Example:


