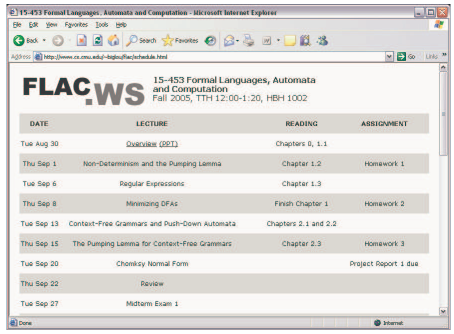
15-453

FORMAL LANGUAGES, AUTOMATA AND COMPUTABILITY

FIRST PROJECT REPORT IS DUE SEPTEMBER 20


15-453 Formal Languages, Automata and Computation - Microsoft Internet Explorer

http://www.cs.cmu.edu/~biglou/flac/schedule.html

FLAC.WS 15-453 Formal Languages, Automata and Computation
Fall 2005, TTH 12:00-1:20, HBH 1002

DATE	LECTURE	READING	ASSIGNMENT
Tue Aug 30	Overview (PPT)	Chapters 0, 1.1	
Thu Sep 1	Non-Determinism and the Pumping Lemma	Chapter 1.2	Homework 1
Tue Sep 6	Regular Expressions	Chapter 1.3	
Thu Sep 8	Minimizing DFAs	Finish Chapter 1	Homework 2
Tue Sep 13	Context-Free Grammars and Push-Down Automata	Chapters 2.1 and 2.2	
Thu Sep 15	The Pumping Lemma for Context-Free Grammars	Chapter 2.3	Homework 3
Tue Sep 20	Chomksy Normal Form		Project Report 1 due
Thu Sep 22	Review		
Tue Sep 27	Midterm Exam 1		

NON-DETERMINISM AND THE PUMPING LEMMA

THURSDAY SEP 1

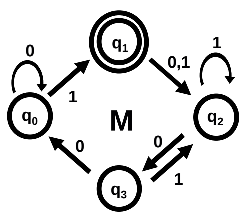
$M = (Q, \Sigma, \delta, q_0, F)$ where $Q = \{q_0, q_1, q_2, q_3\}$

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

$\delta : Q \times \Sigma \rightarrow Q$ transition function*

$q_0 \in Q$ is start state

$F = \{q_1, q_2\} \subseteq Q$ accept states



*

δ	0	1
q_0	q_0	q_1
q_1	q_2	q_2
q_2	q_3	q_2
q_3	q_0	q_2

UNION **THEOREM**

The union of two regular languages is also a regular language

THE REGULAR OPERATIONS

Union: $A \cup B = \{ w \mid w \in A \text{ or } w \in B \}$

Intersection: $A \cap B = \{ w \mid w \in A \text{ and } w \in B \}$

Reverse: $A^R = \{ w_1 \ldots w_k \mid w_k \ldots w_1 \in A \}$

Negation: $\neg A = \{ w \mid w \notin A \}$

Concatenation: $A \cdot B = \{ vw \mid v \in A \text{ and } w \in B \}$

Star: $A^* = \{ w_1 \ldots w_k \mid k \geq 0 \text{ and each } w_i \in A \}$

REVERSE CLOSURE

Regular languages are closed under reverse

Assume L is a regular language and M recognizes L

We build M^R that accepts L^R

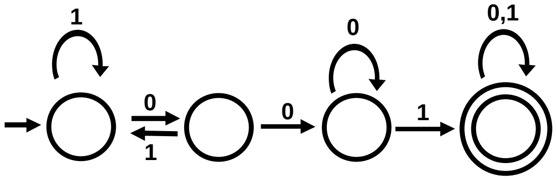
If M accepts w then w describes a directed path in M from start to an accept state

Define M^R as M with the arrows reversed

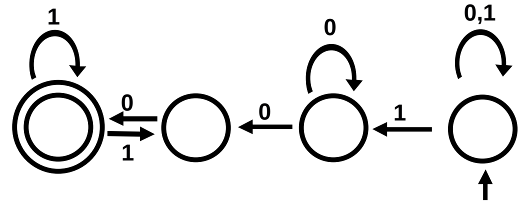
M^R IS NOT ALWAYS A DFA!

It may have many start states

Some states may have too many outgoing edges, or none

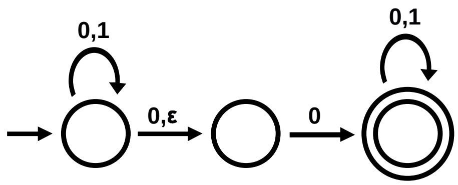


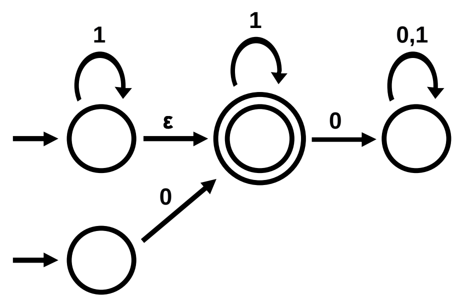
NON-DETERMINISM

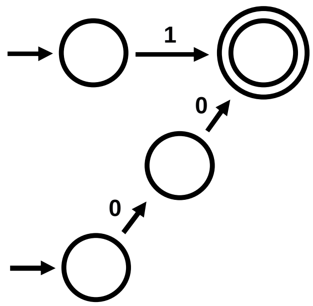


What happens with 100?

We will say that the machine accepts if there is a way to make it reach an accept state







A non-deterministic finite automaton is a 5-tuple N = (Q, Σ, δ, Q_0, F)

Q is the set of states

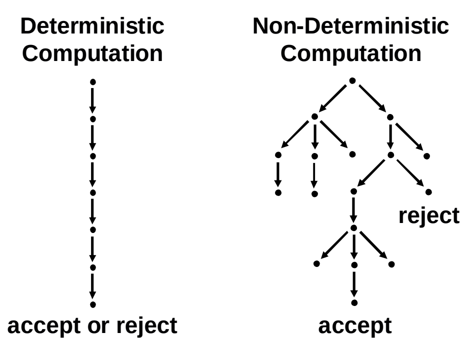
Σ is the alphabet

$\delta : Q \times \Sigma_\varepsilon \rightarrow 2^Q$ is the transition function

$Q_0 \subseteq Q$ is the set of start states

$F \subseteq Q$ is the set of accept states

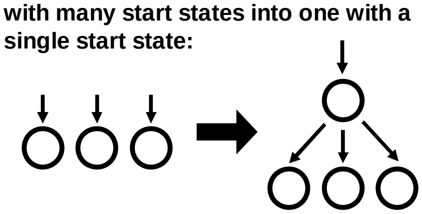
2^Q is the set of subsets of Q and $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$



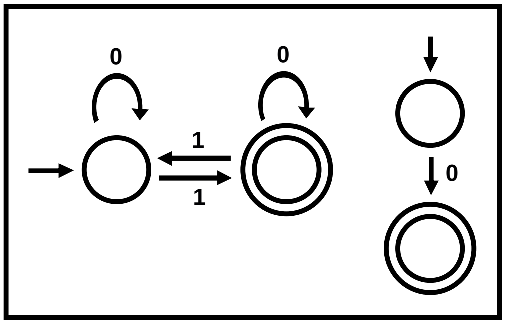
MULTIPLE START STATES

We allow multiple start states for NFAs, and Sipser allows only one

But we can easily convert a machine with many start states into one with a single start state:



UNION THEOREM FOR NFAs



NFAs ARE SIMPLER THAN DFAs

An NFA that recognizes the language {1}:

A DFA that recognizes the language {1}:

Theorem: Every NFA has an equivalent DFA

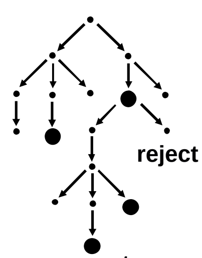
Corollary: A language is regular iff it is recognized by an NFA

Corollary: L is regular iff L^R is regular

FROM NFA TO DFA

Input: N = (Q, Σ, δ, Q_0, F)

Output: M = (Q′, Σ, δ′, q_0′, F′)



To learn if NFA accepts, we could do the computation in parallel, maintaining the set of states where all threads are

Idea:

$Q' = 2^Q$

FROM NFA TO DFA

Input: $N = (Q, \Sigma, \delta, Q_0, F)$

Output: $M = (Q', \Sigma, \delta', q_0', F')$

$Q' = 2^Q$

$\delta' : Q' \times \Sigma \rightarrow Q'$

$\delta'(R,\sigma) = \bigcup_{r \in R} \varepsilon(\ \delta(r,\sigma)\ )$

$q_0' = \varepsilon(Q_0)$

$F' = \{ R \in Q' \mid f \in R \text{ for some } f \in F \}$

REGULAR LANGUAGES CLOSED UNDER **CONCATENATION**

REGULAR LANGUAGES ARE COLSED UNDER **REGULAR OPERATIONS**

Union: $A \cup B = \{ w \mid w \in A \text{ or } w \in B \}$

Intersection: $A \cap B = \{ w \mid w \in A \text{ and } w \in B \}$

Reverse: $A^R = \{ w_1 \ldots w_k \mid w_k \ldots w_1 \in A \}$

Negation: $\neg A = \{ w \mid w \notin A \}$

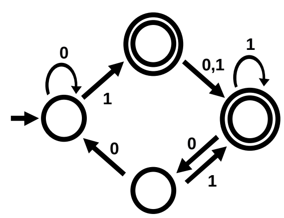
Concatenation: $A \cdot B = \{ vw \mid v \in A \text{ and } w \in B \}$

Star: $A^* = \{ w_1 \ldots w_k \mid k \geq 0 \text{ and each } w_i \in A \}$

REGULAR LANGUAGES CLOSED UNDER **STAR**

Let L be a regular language and M be a DFA for L

We construct an NFA N that recognizes L*



Formally:

Input: $M = (Q, \Sigma, \delta, q_1, F)$

Output: $N = (Q', \Sigma, \delta', \{q_0\}, F')$

$Q' = Q \cup \{q_0\}$

$F' = F \cup \{q_0\}$

$$\delta'(q,a) = \begin{cases} \{\delta(q,a)\} & \text{if } q \in Q \text{ and } a \neq \varepsilon \\ \{q_1\} & \text{if } q \in F \text{ and } a = \varepsilon \\ \{q_1\} & \text{if } q = q_0 \text{ and } a = \varepsilon \\ \varnothing & \text{if } q = q_0 \text{ and } a \neq \varepsilon \\ \varnothing & \text{else} \end{cases}$$

L(N) = L*

Assume $w = w_1 \ldots w_k$ is in L*, where $w_1, \ldots, w_k \in L$

We show N accepts w by induction on k

Base Cases:

- ✓ **k = 0**
- ✓ **k = 1**

Inductive Step:

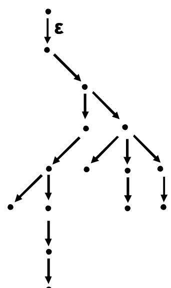
Assume N accepts all strings $v = v_1 \ldots v_k \in L^*$, and let $u = u_1 \ldots u_k u_{k+1} \in L^*$

Since N accepts $u_1 \ldots u_k$ and M accepts u_{k+1}, N must accept u

Assume w is accepted by N, we show $w \in L^*$

If $w = \varepsilon$, then $w \in L^*$

If $w \neq \varepsilon$



accept

REGULAR LANGUAGES ARE COLSED UNDER REGULAR OPERATIONS

Union: $A \cup B = \{ w \mid w \in A \text{ or } w \in B \}$

Intersection: $A \cap B = \{ w \mid w \in A \text{ and } w \in B \}$

Reverse: $A^R = \{ w_1 \ldots w_k \mid w_k \ldots w_1 \in A \}$

Negation: $\neg A = \{ w \mid w \notin A \}$

Concatenation: $A \cdot B = \{ vw \mid v \in A \text{ and } w \in B \}$

Star: $A^* = \{ w_1 \ldots w_k \mid k \geq 0 \text{ and each } w_i \in A \}$

SOME LANGUAGES **ARE NOT** REGULAR

B = $\{0^n1^n \mid n \geq 0\}$ is NOT regular!

WHICH OF THESE ARE REGULAR

C = { w | w has equal number of 1s and 0s}

D = { w | w has equal number of occurrences of 01 and 10}

THE PUMPING LEMMA

Let L be a regular language with $|L| = \infty$

Then there exists a positive integer P such that

if $w \in L$ and $|w| \geq P$

then w = xyz, where:

1. $|y| > 0$
2. $|xy| \leq P$
3. $xy^iz \in L$ for any $i \geq 0$

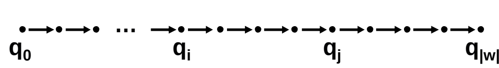
Let M be a DFA that recognizes L

Let P be the number of states in M

Assume $w \in L$ is such that $|w| \geq P$

We show w = xyz

1. $|y| > 0$
2. $|xy| \leq P$
3. $xy^iz \in L$ for any $i \geq 0$



q_0 q_i q_j $q_{|w|}$

There must be j > i such that $q_i = q_j$

USING THE **PUMPING LEMMA**

Use the pumping lemma to prove that
$B = \{0^n1^n \mid n \geq 0\}$ is not regular

Hint: Assume B is regular, and try pumping $s = 0^P1^P$

WWW.FLAC.WS

Read Chapter 1.2 of the book for next time