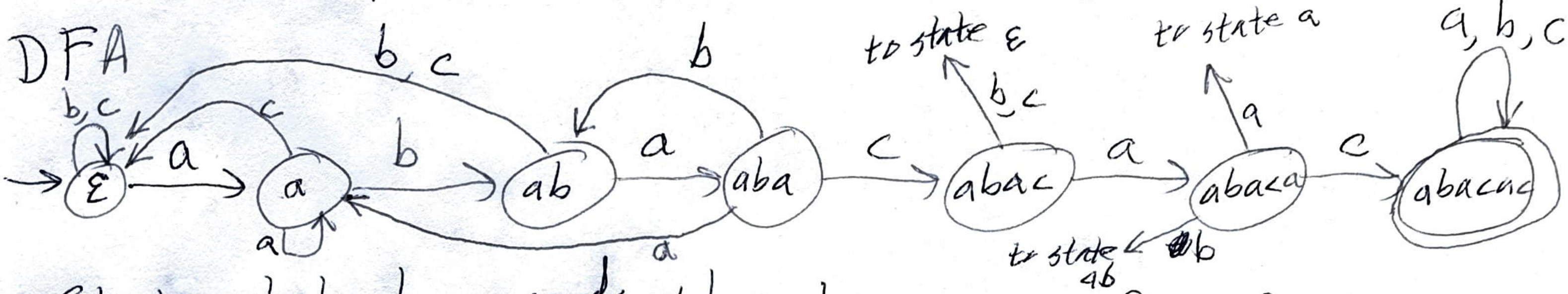
CSCE 355
1/26/2026

DFA for text search

$\Sigma = \{a, b, c\}$ $w = abacac$

Accept an input x iff w is a substring of x

DFA



State label records the longest prefix of ~~the~~ w that is a suffix of what has been read so far.

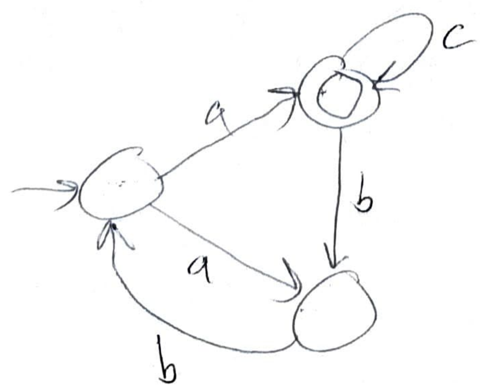
Building the DFA from w uses the Knuth-Morris-Pratt algorithm (adapted).

Nondeterminism

For a DFA, in the transition diagram, the rule is exactly one edge ~~leave~~ leaving each state labeled with each alphabet symbol.

A nondeterministic finite automaton (NFA) has no such rules, e.g.

N:



ab rejected
aa rejected
abbac accepted

An NFA accepts an input string w if there is some sequence of choices to step thru that end in an accepting state after reading the entire string.

<u>Def</u>: An NFA is a tuple $\langle Q, \Sigma, \delta, q_0, F \rangle$ where

Q is as with a DFA

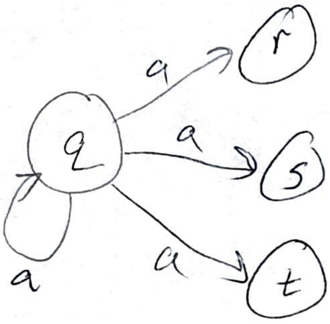
Σ " " " "

q_0 " " " "

F " " " "

$\delta : Q \times \Sigma \to 2^Q$

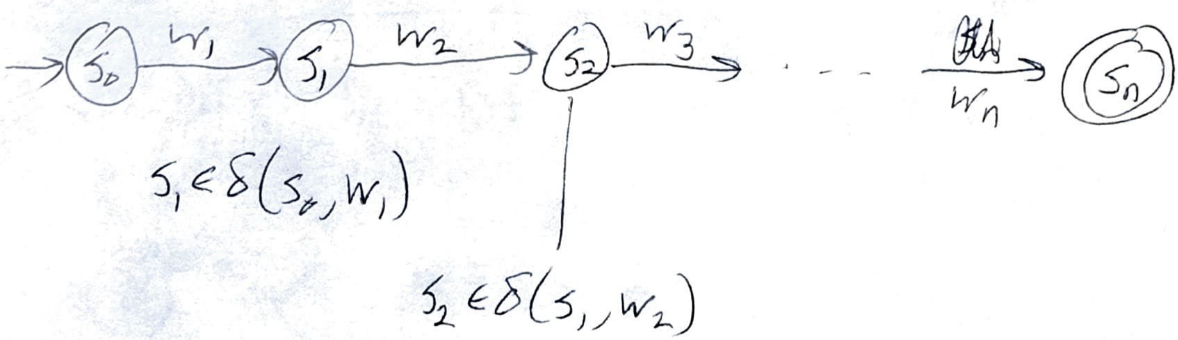
2^Q is the set of all subsets of Q ("powerset" of Q)



means $\delta(q, a) = \{q, r, s, t\}$

<u>Def</u>: Let $N = \langle Q, \Sigma, \delta, q_0, F \rangle$ be an NFA, and let $w \in \Sigma^*$ be a string. <u>N accepts w</u> if there exist $s_0, s_1, \ldots, s_n \in Q$ and symbols $w_1, w_2, \ldots, w_n \in \Sigma$ such that

1) $w = w_1 w_2 \cdots w_n$ (concatenated)
2) $s_0 = q_0$
3) $s_n \in F$
4) For all $1 \leq j \leq n$, $s_j \in \delta(s_{j-1}, w_j)$

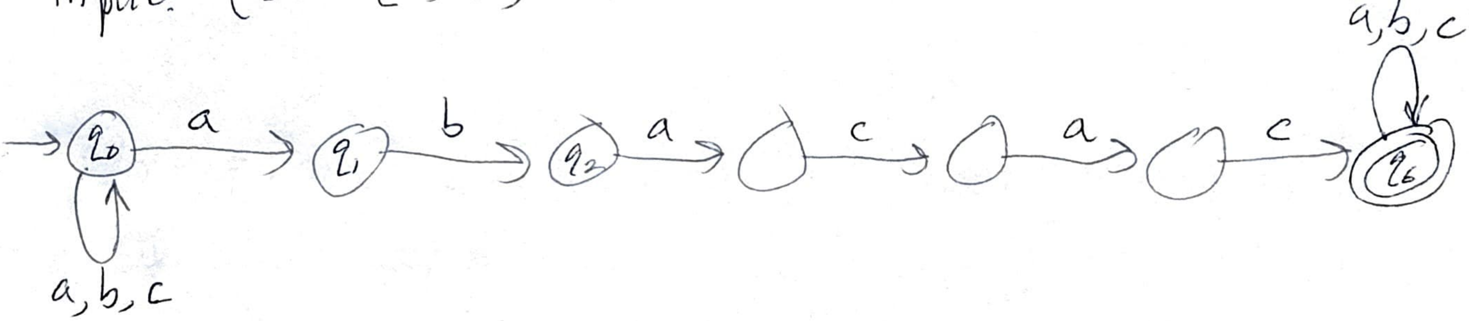


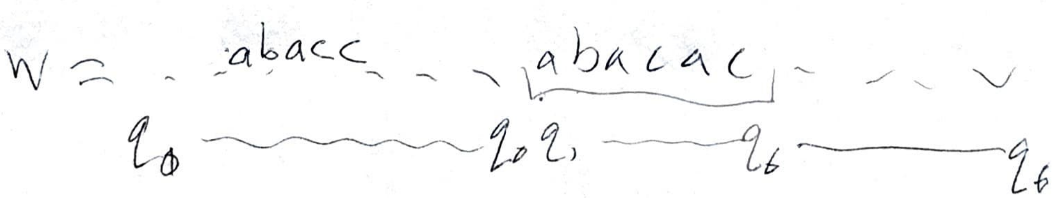
We call $s_0, \ldots, s_n$ an accepting path of N on input w.
(other paths are rejecting paths)

$L(N) := \{w : N \text{ accepts } w\}$ (just as with a DFA)

N recognizes $L(N)$

Ex: NFA to accept if abacac is a substring of the input. ($\Sigma = \{a,b,c\}$)



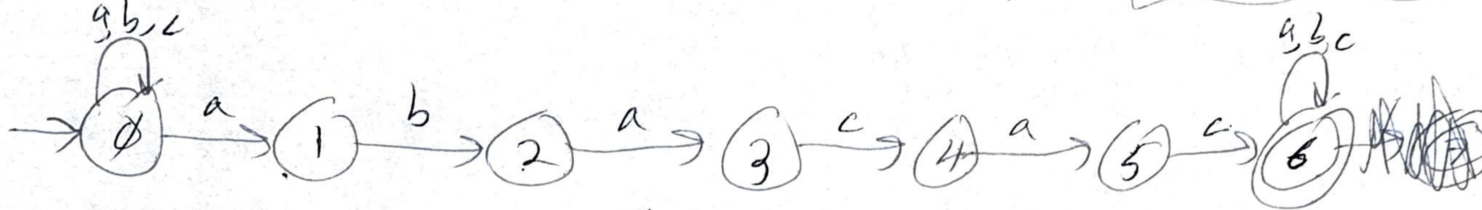


N is equivalent (recog. same lang.) as the DFA we did earlier.

Theorem: For every NFA there is an equivalent DFA.
[Proof later]

Simulating an NFA on input string.

Example: The NFA below with input aabacacb



symbol	possible states after reading
initial	0
a	0, 1
a	0, 1
b	0, 2
a	0, 1, 3
c	0, 4
a	0, 1, 5
c	0, 6
b	0, 6

Each state is uniquely determined by the previous state set and the symbol being read.

Key idea: Make the states of the DFA be sets of states of the NFA.

Sets-of-states construction to turn an NFA into a equiv. DFA.

Given NFA $N = \langle Q, \Sigma, \delta, q_0, F \rangle$

Define the DFA $A = \langle 2^Q, \Sigma, \Delta, Q_0, \mathcal{F} \rangle$

where $2^Q := \{S : S \subseteq Q\}$

$Q_0 := \{q_0\}$

$\mathcal{F} := \{S \subseteq Q : S \cap F \neq \emptyset\}$

$\forall S \subseteq Q, a \in \Sigma, \quad \Delta(S, a) = \bigcup_{q \in S} \delta(q, a)$