4/21/2010

☐ HW3

OK to write a DFA for #1. (The problem given is not a good one for an NFA.)

But make sure the formal definition is in NFA form

DFA is 8 states

3A - B can be DFA or NFA

(The combined form will be an NFA)

#2 If #1 is a DFA, all your subsets will be singletons. Make sure you write down the formal construction. Use null set for all the wasted transitions.

#5 : Your output will be either accept or reject. (e.g. accept if first is larger, reject otherwise)

The alphabet is not specified. You can specify yours.

Suggestion: read pairs of symbols.

Assume the smaller string is padded with zeroes.

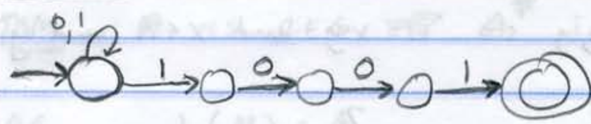
NFA'S : EXAMPLES

"ends in" - good for NFA's

L = {w | w **ends in** 1001}

For DFA 16 possible ending combos requires 16 states.

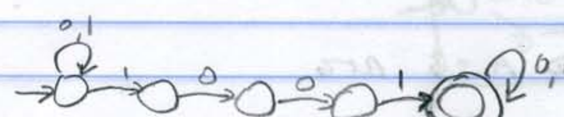
But NFA:



110010 → dies when it reads 0.
No need for trap state.

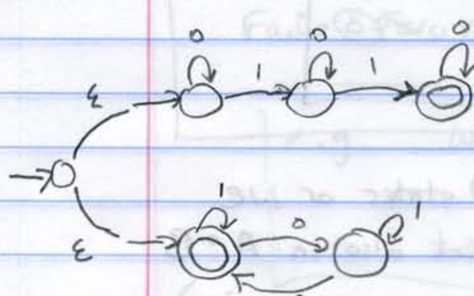
"contains a substring" - good for NFA's

L = {w | w **contains** 1001}



"OR" - good for NFA's

L = {w | w contains exactly 2 1's
OR EVEN # of 0's}

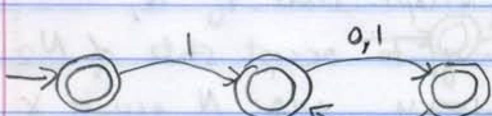


[exactly 2 1's]

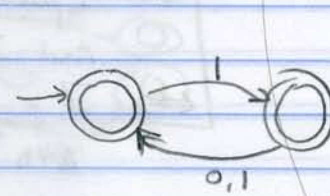
[even 0's]

L = {w | every odd position of w is a 1}

examples: ε, 1, 01, 101, 111 [ε in set b/c it has no odd positions]



or



Even though all states are accept, it will still reject all the right strings (by dying)

CLOSURE PROPERTIES OF REG. LANG'S

LAST TIME: A, B are reg $\Rightarrow A \cup B$ REG

CONCAT $A \circ B = \{ w \mid \exists\, x, y \text{ s.t. } w = xy \text{ and } x \in A \text{ and } y \in B \}$

$A = \{00, 11\}$
$A \circ A = \{0000, 0011, 1100, 1111\}$ } EXAMPLE

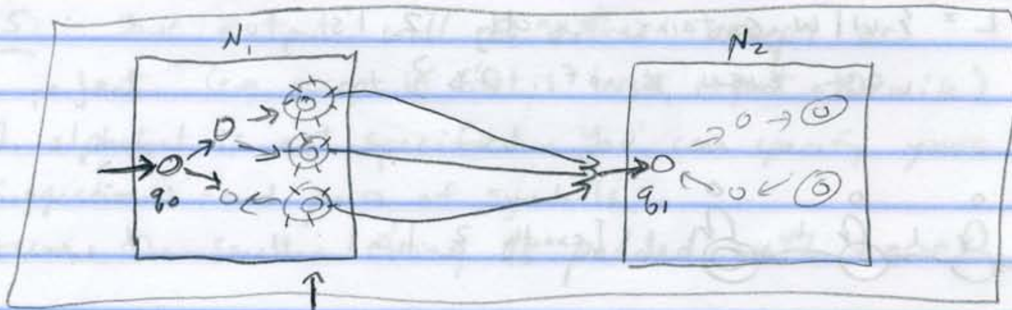
THM A, B REG $\Rightarrow A \circ B$ REG

Pf

$L(N_1) = A$ $\qquad\qquad$ $L(N_2) = B$

N

$L(N) = A \circ B$



These cannot remain accept states or we would accept strings in A, not also in $A \circ B$ (like 00 or 11 from our example)

FORMAL PROOF THAT THE MACHINE YOU CREATED IS CORRECT

$w \in L(N) \Leftrightarrow N$ accepts w
$\Leftrightarrow \exists$ path from q_0 to accept state of N_2
$\Leftrightarrow \exists$ path from q_0 to accept state of N_1 And $\exists$ path from q_1 to accept state of N_2
$\Leftrightarrow \exists\, x, y$ s.t. $w = xy$ AND N_1 accepts x AND N_2 accepts y

STAR OPERATION

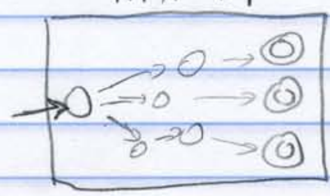
A is some lang.

$$A^* = \{x_1, x_2, x_3, \dots x_k \mid x_i \in A \text{ and } k \geq 0\}$$

THM A is REG $\Rightarrow A^*$ is REG.

Pf. $L(N_1) = A$

NFA N_1



FIRST IDEA: JUST DO ε TRANSITIONS FROM ALL THE ACCEPT STATES BACK TO THE START STATE.

BUT WHAT ABOUT ε, which is always $\in A^*$?

Can we just always make the start state an accept state? Fails for some NFA's.

e.g. w contains a 1

So: Add a new accept state at beginning

NFA N