CSE 396 Lecture Tue Feb 12 Spring 2019

Top Hat #7348 A nondeterministic finite automaton (NFA) is a 5-tuple $N = (Q, \Sigma, \delta, s, F)$ where:

- Q is a finite set of states
- Σ is a finite alphabet
- $s \in Q$ is the start state
- $F \subseteq Q$ is the set of final states

} all as in a DFA. What's different?

$\delta \subseteq (Q \times \Sigma) \times Q$ is the instruction set — without ε-arcs for now.

A DFA is an NFA in which no two instructions (p, c, r) and (q, d, r') have both $p = q$ and $c = d$, and for which every $q \in Q$, $c \in \Sigma$, has an instruction (q, c, r).

The text uses as default the <u>NFA with ε-transitions</u>, where

$$\delta \subseteq (Q \times \{\Sigma \cup \{\varepsilon\}\}) \times Q \quad \text{Typical instructions:}$$

(p, c, q) $c \in \Sigma$

or (p, ε, q) also allowed.

$p \xrightarrow{c} q$ $p \xrightarrow{\varepsilon} q$

Self-loops $q = p$ also allowed: (p, c, p).

Text defines $\delta : (Q \times (\Sigma \cup \{\varepsilon\})) \rightarrow \underbrace{\mathcal{P}(Q)}_{\text{power set}}$

en $\delta(p, c) = \{$all q s.t. (p, c, q) is an instruction$\}$.

$\delta(p, \varepsilon) = \{q : (p, \varepsilon, q)$ is allowed$\}$

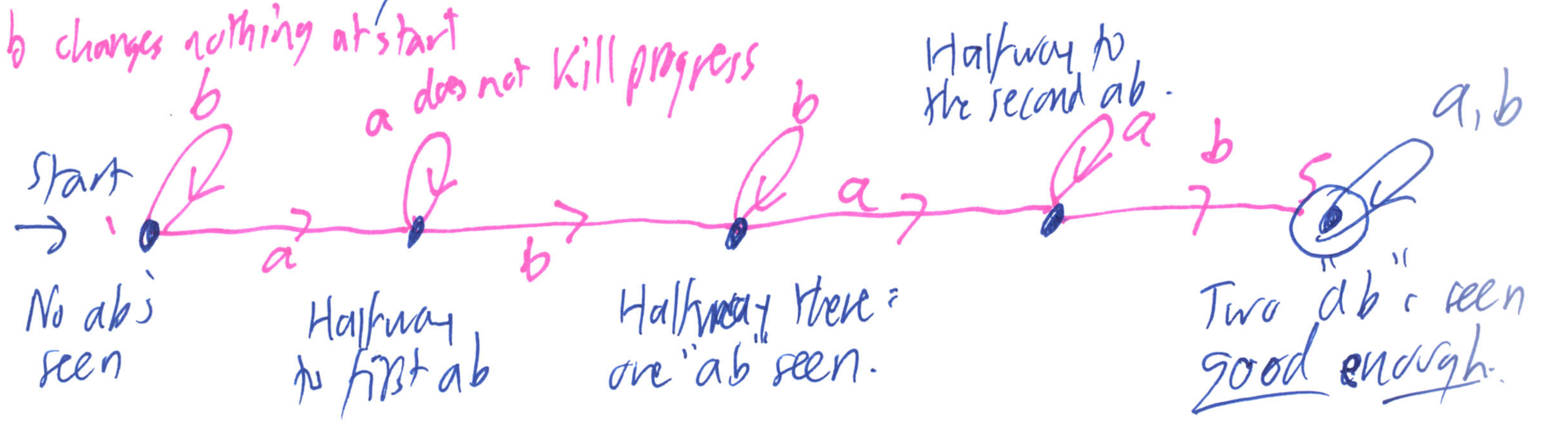
Examples: Let $L = \{x \in \{a,b\}^* :$ the substring $\underline{ab}$ occurs in x at most once $\}$

$\Sigma = \{a, b\}$

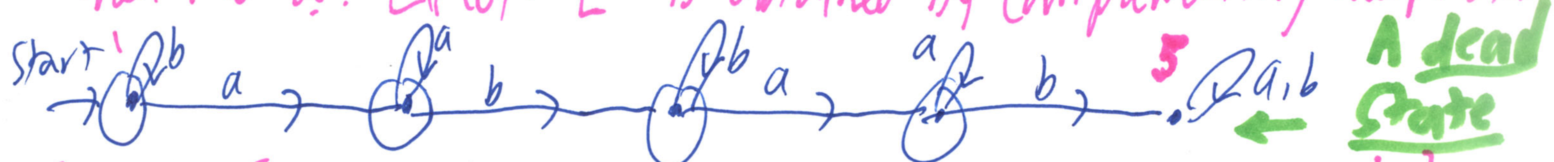
$\tilde{L} = \Sigma^* \setminus L$

Note $\tilde{L} = \{x \in \{a,b\}^* :$ ab occurs in x $\underline{\text{at least twice}}\}$

Design a DFA M_1 s.t. $L(M_1) = \tilde{L}$ first.



Then M_0 st. $L(M_0) = L$ is obtained by complementing accept states



$Q_0 = Q_1 = \{1, 2, 3, 4, 5\}$ $F_1 = \{5\}$ $F_0 = Q \setminus F_1 = \{1, 2, 3, 4\}$.

NFA:

Claim: This N_1 is correct because $\tilde{L} = \underline{(a \cup b)^* ab (a \cup b)^* ab (a \cup b)^*}$

To execute N_1 on a string such as $x = \underline{abbabaa}$, we keep track of all the possible states N_1 could be in after $\underline{\text{processing}}$ the ith char

$\{1\}$ a $\{1,2\}$ b $\{1,3\}$ b $\{1,3\}$ a $\{1,2,3,4\}$ b $\{1,3,5\}$ accept.

a $\{1,2,3,4,5\}$ a $\{1,2,3,4,5\}$

N_1 has nondeterminism $\underline{\text{at}}$ 1 and at 3 on 'a'.

<u>Process</u> for converting the NFA N_1 (which has no {ε-arcs}) into an equivalent DFA. Use the functional view of δ. DFA M is

$$(Q, \Sigma, \Delta, S, \mathcal{F}) \text{ where}$$

$$\Delta : Q \times \Sigma \to Q.$$

δ

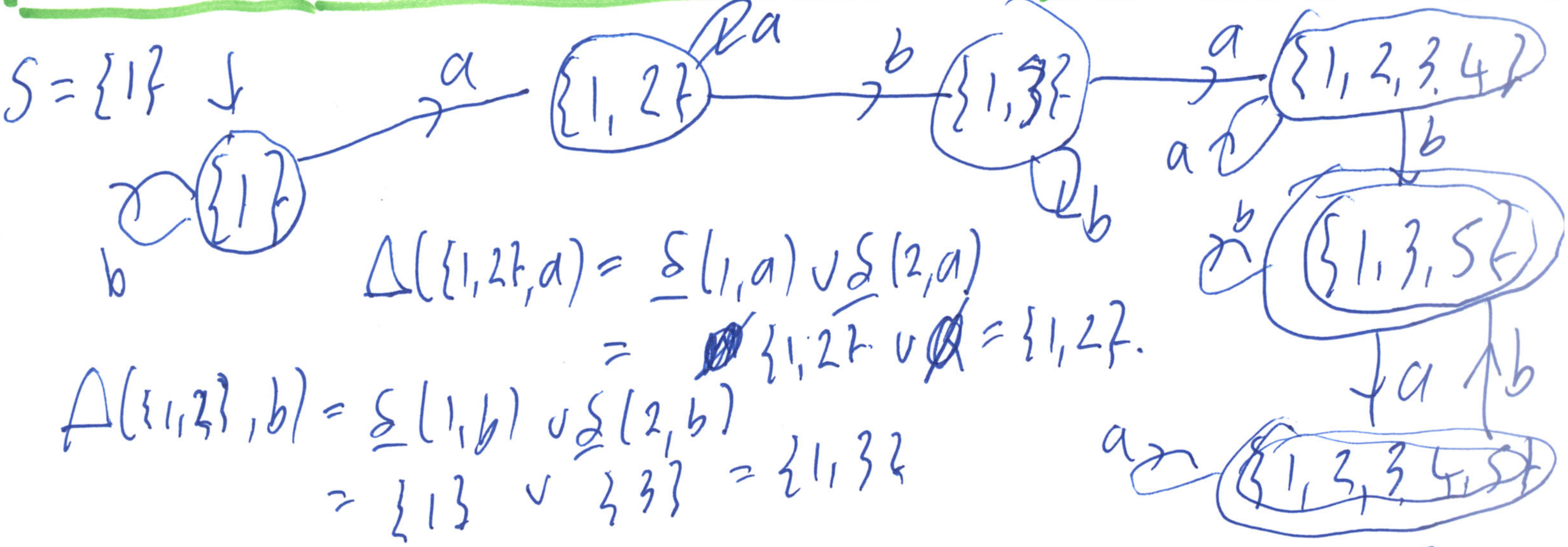
State	a	b
1	1, 2	1
2	∅	3
3	3, 4	3
4	∅	5
5	5	5

$\Delta(P, c) \mapsto \{q : \text{for some } p \in P, (p, c, q) \in \delta\}$ i.e. $q \in \underline{\delta}(p, c)$.

$P \subseteq Q$ Q is <u>the</u> Q of N_1

$\mathcal{F} = \{R \subseteq Q : R$ includes an accepting state, i.e. $R \cap F \neq \emptyset\}$.

$S = \{1\}$



$\Delta(\{1,2\}, a) = \underline{\delta}(1, a) \cup \underline{\delta}(2, a)$
$= \{1, 2\} \cup \emptyset = \{1, 2\}.$

$\Delta(\{1,2\}, b) = \underline{\delta}(1, b) \cup \underline{\delta}(2, b)$
$= \{1\} \cup \{3\} = \{1, 3\}$

This is a correct DFA M' and we can complement it as before.

<u>Extra/Footnote</u>: Note that this last DFA has two accepting states, not just the one of our earlier M, designed by logical reasoning. You can of course just merge {1,3,5} and {1,2,3,4,5} together. What this <u>means</u> is that the "NFA to DFA" algorithm does not always produce a minimum DFA.