Algorithms for DFAs

Algorithms for regular languages

NFA $\longrightarrow$ DFA algorithmic

RegExp $\longrightarrow$ NFAs closure prop.

↳ algorithm

NFA $\xrightarrow{\text{GNFAs}}$ RegExp algorithmic.

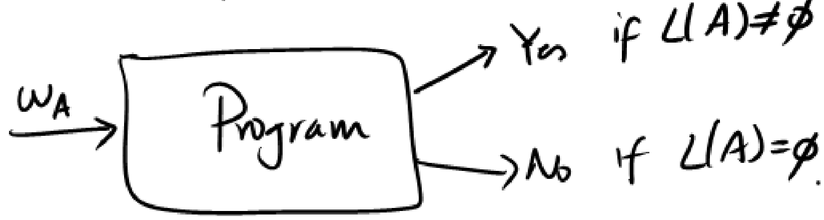
Emptiness problem.

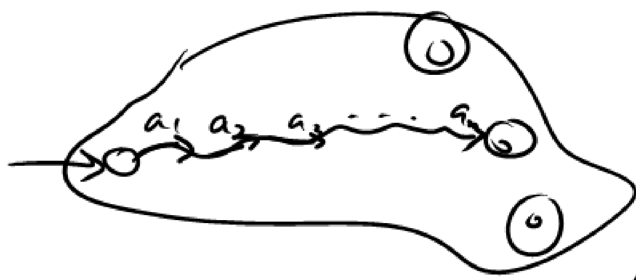
"Given a DFA A, is $L(A) \neq \emptyset$?"

Is this problem *decidable*?

DFA $A = (Q, \Sigma, \delta, q_0, F)$

encoded as a string $\to w_A$.





$a_1 \dots a_n \in L$

~~states~~ nodes.

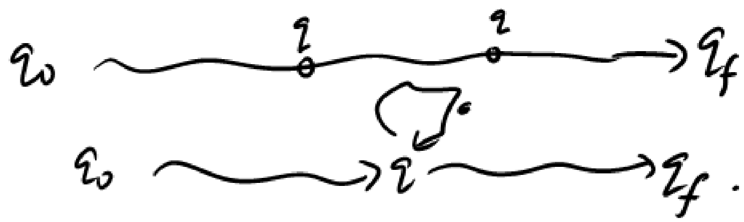
edges

DFS works in time $O(n+m)$

Given a DFA A, is $L(A)$ infinite?

$uv^iw \in L$ $\forall i$

If L is infinite, take a long word $w \in L$ $|w| > n$



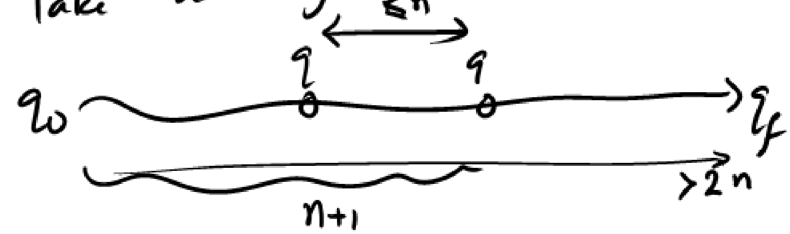
$L(A)$ is infinite iff $\exists w \in L.\ n \leq |w| \leq 2n$

Suppose $\exists w \in L.\ n \leq |w| \leq 2n$.

Then PL, $L(A)$ is infinite.

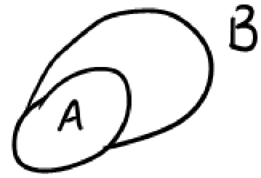
Suppose $L(A)$ is infinite.

Take a long word $x \in L$, $|x| > n$.



DFA minimization

$$\underline{L(A_1) \subseteq L(A_2)}$$



$$\iff \underline{L(A_1) \cap \overline{L(A_2)}} = \phi$$

$A \subseteq B$

$\iff A \cap \overline{B} = \phi .$

Given $A_2 \xrightarrow{\text{complement construction}} B$ $\quad L(B) = \overline{L(A_2)}$

$$\iff L(\underline{A_1}) \cap L(\underline{B}) = \phi$$

$\to C$ $\quad L(C) = L(A_1) \cap L(B)$

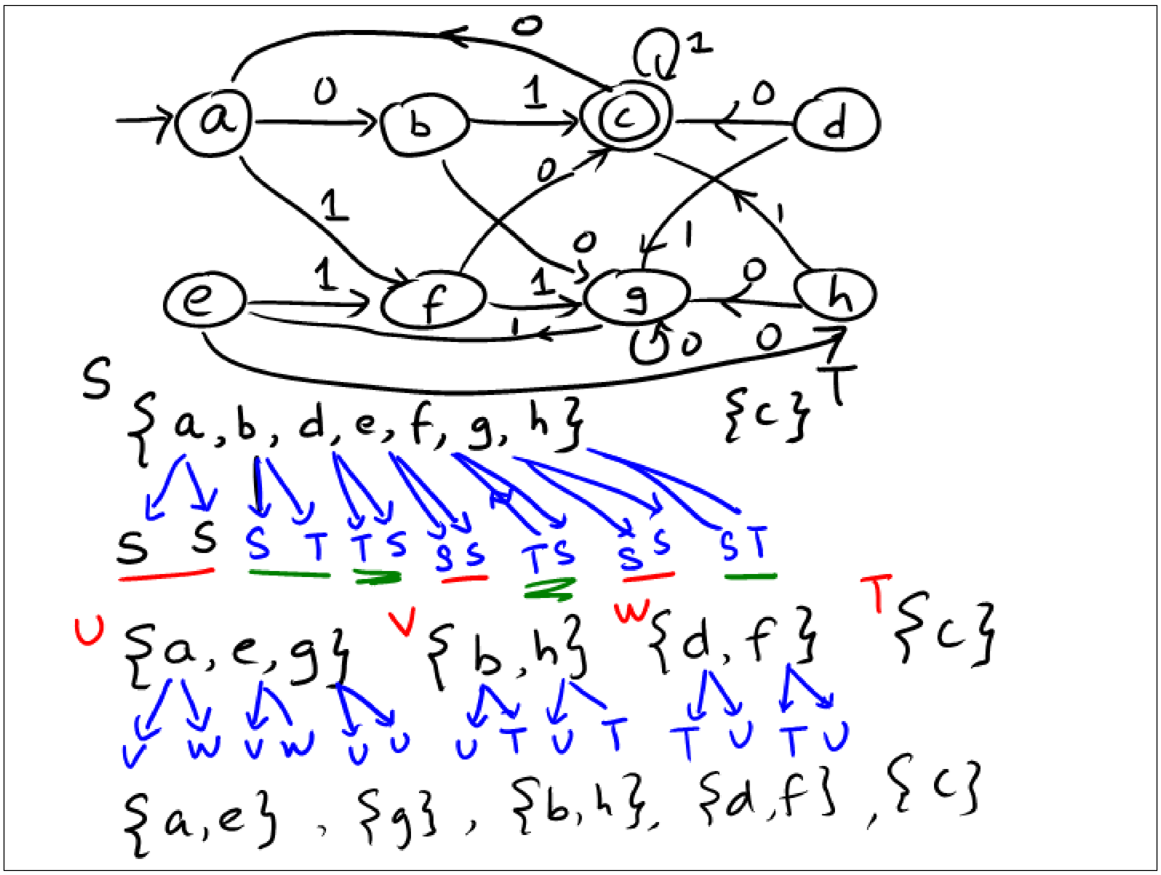
$$\iff L(C) = \phi$$

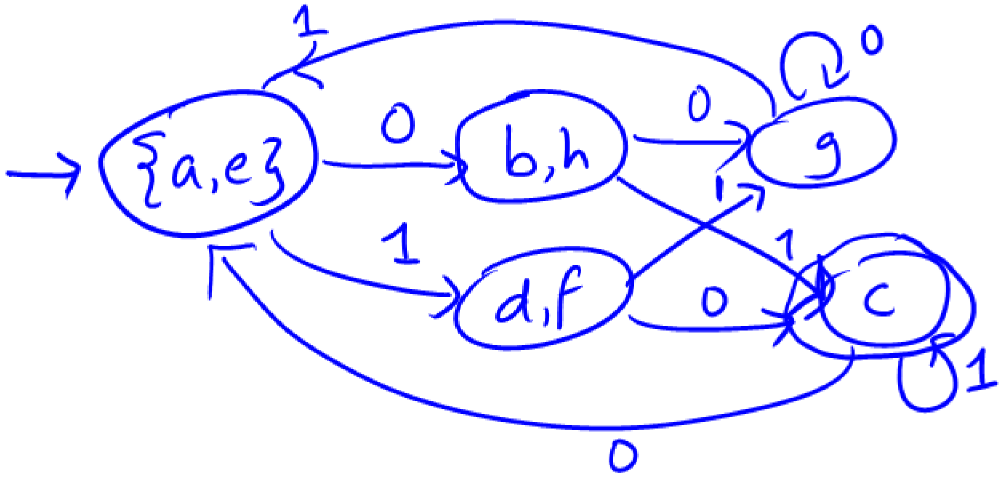
$$L(A_1) = L(A_2)$$

$$\iff L(A_1) \subseteq L(A_2) \land L(A_2) \subseteq L(A_1)$$

Minimization.

Given a DFA A, compute the minimal DFA B, $L(A) = L(B)$.


S {a, b, d, e, f, g, h}
{c} T
S S S T T S S S T S S S S T
U {a, e, g}
V {b, h}
W {d, f}
T {c}
V W V W U U U T U T T U T U
{a, e}, {g}, {b, h}, {d, f}, {c}



Algm works in time $O(kn^2)$
$k = |\Sigma|$ n: # states.

Algorithm: Input $A = (Q, \Sigma, \delta, q_0, F)$

If $F = \emptyset$ return 

If $F = Q$ return 

$P_{new} = (Q \setminus F, F)$; $i := 0$

do { $P_{old} := P_{new}$

P_{new} : If q_1, q_2 are in diff sets in P_{old}, then they must be in diff sets in P_{new}

: If q_1, q_2 are in the same set in P_{old}, then q_1, q_2 are in the same set in P_{new} iff $\forall a \in \Sigma.\ q_1 \xrightarrow{a} q_1'$ & $q_2 \xrightarrow{a} q_2'$ then q_1' & q_2' are in the same set in P_{old}.

} while ($P_{old} \neq P_{new}$)

Let $P_{new} = (S_1, \ldots S_k)$

Create DFA with states $\{S_1, \ldots S_k\}$

Initial state : S_i where $q_0 \in S_i$

Final states : $\{S_i \mid S_i \subseteq F\}$

$$\hat{\delta}(S_i, a) = S_j \quad \text{if}$$

$$\exists q \in S_i, \ \delta(q, a) \in S_j$$

This is the minimal DFA for $L(A)$.