1/13/11 CS 360 Lecture 3

Review:

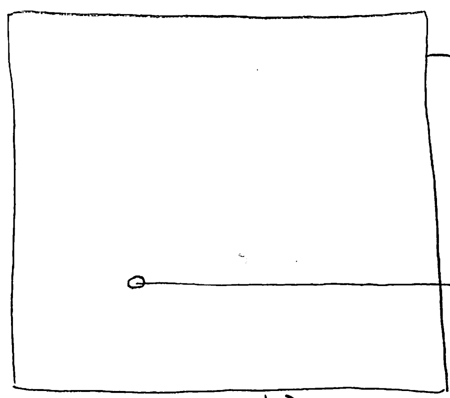
$\{0,1\}^*$ = set of all finite-length strings over the alphabet $\{0,1\}$

e.g. $\varepsilon \in \{0,1\}^*$ (empty string), $000\ldots \notin \{0,1\}^*$ (infinite length)

$L \subseteq \{0,1\}^* \iff$ language / decision problem

e.g. $L = \{\underbrace{\langle G \rangle}_{\text{binary encoding of } G} : G \text{ is a connected graph}\}$ Graph connectivity

$L = \{\langle n \rangle : n \in \mathbb{Z} \text{ is prime}\}$ Primality



(not to scale)

$2^{\{0,1\}^*}$ = set of all languages
- uncountable

set of all languages accepted by some DFA
- countable, since {DFAs} is countable

= set of all languages accepted by some NFA

= "regular languages"

Theorem: Language $L \subseteq \Sigma^*$ is accepted by a DFA if and only if it is accepted by an NFA.

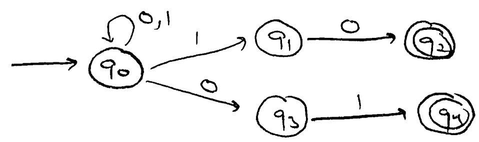
DFA: $\delta : Q \times \Sigma \to Q$

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

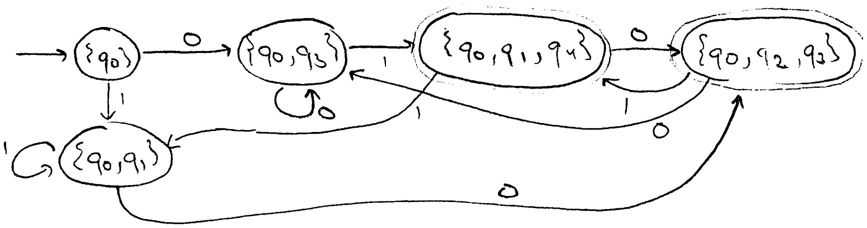
Idea: One direction is trivial, since DFAs are also NFAs. For the other direction, simulate the NFA deterministically by adding a state in M for every subset of states of N.

Example: $L = \{w \mid w \text{ ends in } 01 \text{ or } 10\}$

N



M



Claim: There exist regular languages that require exponentially many more states in an accepting DFA than in an NFA.

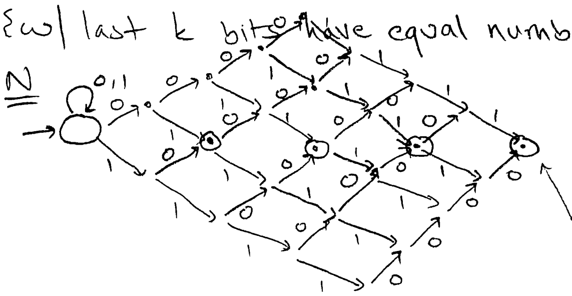
$\Rightarrow$ Simple computational model (finite automata) where nondeterminism (power of guessing) provably gives exponential efficiency improvements.

Idea: Force the DFA to remember substrings of length k.

Example?

$L = \{w \mid \text{last } k \text{ bits have equal numbers of 0s and 1s}\}$

N



$\left(\frac{k}{2}\right)^2$ states

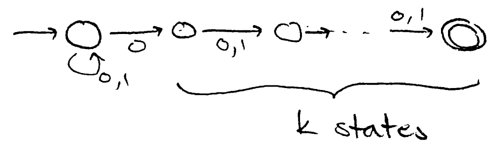
from here, what should DFA do?

M If a DFA only remembers the # of 0s and 1s in the last k characters, then how can it increment to the $k+1$'st character?

- it needs to know what character fell from the window — which requires remembering all the input in the window...

Example: $L_k = \{ w \mid k\text{th character before the end is a } 0 \}$

NFA N:



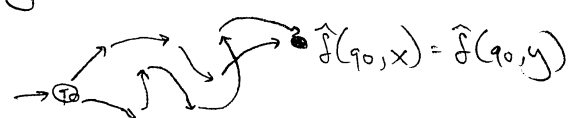
$k+1$ states total

<u>Claim</u>: Any DFA M with $L(M) = L$ must have at least 2^k states

Proof:

Assume otherwise. Let $L(M) = L$, where M has $< 2^k$ states.

$\Rightarrow \exists\ x, y \in \{0,1\}^k,\ x \neq y$, with
$\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$
(pigeon-hole principle)



Assume $x_j = 0 \neq y_j = 1$ and consider inputs

xz

yz

where $z \in \{0,1\}^{j-1}$.

Either M accepts <u>both</u> xz and yz or it accepts <u>neither</u> (since $\hat{\delta}(q_0, xz) = \hat{\delta}(\hat{\delta}(q_0,x),z) = \hat{\delta}(\hat{\delta}(q_0,y),z) = \hat{\delta}(q_0, yz)$).
But $x \in L_k$ while $y \notin L_k$, a contradiction. $\square$

NFA $\Rightarrow$ DFA

Theorem: Let $N = (\overset{\text{states}}{Q}, \overset{\text{alphabet}}{\Sigma}, \overset{\text{trans fn}}{\delta}, \overset{\text{start}}{q_0}, \overset{\text{final}}{F})$ be an NFA, and $L = L(N) \subseteq \Sigma^*$.
Then there exists a DFA M s.t. $L = L(M)$.

Proof:

Let $M = (2^Q, \Sigma, \rho, \{q_0\}, \{S \subseteq Q : S \cap F \neq \emptyset\})$, where

$$\rho(S,a) = \bigcup_{q \in S} \delta(q,a).$$

Recall the definitions of extended transition functions:

- For an NFA

$$\hat{\delta} : Q \times \Sigma^* \to 2^Q$$

- $\hat{\delta}(q, \varepsilon) = \{q\}$
- $\hat{\delta}(q, xa) = \bigcup_{p \in \hat{\delta}(q,x)} \delta(p,a)$

For a DFA:

$$\hat{\rho} : 2^Q \times \Sigma^* \to 2^Q$$

- $\hat{\rho}(S, \varepsilon) = S$
- $\hat{\rho}(S, xa) = \rho(\hat{\rho}(S,x), a)$

Let $D : 2^Q \times \Sigma^* \to 2^Q$ be defined by $D(S,x) = \bigcup_{q \in S} \hat{\delta}(q,x)$.

Claim: $D = \hat{\rho}$.

Proof: By induction in $|x|$:

Base case: $|x| = 0$, ie. $x = \varepsilon$

$$D(S, \varepsilon) = \bigcup_{q \in S} \hat{\delta}(q, \varepsilon) = \bigcup_{q \in S} \{q\} = S = \hat{\rho}(S, \varepsilon) \quad \checkmark$$

Induction step:

Assume $D(S,x) = \hat{\rho}(S,x)$ $\forall S \subseteq Q$ and $x \in \Sigma^n$:

$$D(S, xa) = \bigcup_{q \in S} \hat{\delta}(q, xa) = \bigcup_{q \in S} \bigcup_{p \in \hat{\delta}(q,x)} \delta(p,a) = \bigcup_{p \in D(S,x)} \delta(p,a)$$

$$\hat{\rho}(S, xa) = \rho(\hat{\rho}(S,x), a) = \bigcup_{p \in \hat{\rho}(S,x)} \delta(p,a) \quad \checkmark \quad \square$$

Now

$$D(\{q_0\}, x) = \hat{\rho}(\{q_0\}, x)$$

$$L(N) = \{x \in \Sigma^* : \hat{\delta}(q_0, x) \cap F \neq \emptyset\}$$

$$L(M) = \{x \in \Sigma^* : \hat{\rho}(\{q_0\}, x) \in \{T \subseteq Q : T \cap F \neq \emptyset\}\}$$

$$= \{x \in \Sigma^* : \hat{\rho}(\{q_0\}, x) \cap F \neq \emptyset\}$$

$\therefore L(M) = L(N)$. $\checkmark$ $\square$