Review:

\( \Sigma^* \) = set of all finite-length strings over the alphabet \( \Sigma \)

- \( \varepsilon \in \Sigma^* \) (empty string), 000... \( \notin \Sigma^* \) (infinite length)

\( L \subseteq \Sigma^* \iff \) language / decision problem

- \( L = \{ \langle G \rangle : G \text{ is a connected graph} \} \) Graph connectivity
  - binary encoding of \( G \)

\( L = \{ \langle n \rangle : n \in \mathbb{N} \text{ is prime} \} \) Primality

\( \mathcal{L}^{\Sigma^*} = \) set of all languages

- uncountable

(set of all languages accepted by some DFA)

- countable, since \( \{\text{DFAs}\} \) is countable

- set of all languages accepted by some NFA

\( \text{"regular languages"} \)

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

\[ \delta : Q \times \Sigma \rightarrow 2^Q \]

Idea: One direction is trivial, since DFAs are also NFAs.

For the other direction, simulate the NFA deterministically by adding a state \( m \) for every subset of states of \( N \).
Example: \( L = \{ \omega \mid \omega \text{ ends in } 01 \text{ or } 10 \} \)

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

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:

\( L = \{ \omega \mid \text{last } k \text{ bits have equal numbers of 0s and 1s} \} \)

\( M \) if a DFA only remembers the \# of 0s and 1s in the last \( k \) characters, then how can it remember 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: \( L_k = \varepsilon \omega \mid \text{kth character before the end is a } \alpha \).

NFA \( N \): \[
\begin{array}{c}
\text{\( q \)} \\
\downarrow \\
\text{\( \alpha \)} \\
\downarrow \\
\text{\( \varepsilon \)} \\
\downarrow \\
\text{\( \omega \)} \\
\downarrow \\
\text{\( \alpha \)} \\
\downarrow \\
\text{\( \varepsilon \)}
\end{array}
\]
k states

\( k+1 \) states, total

Claim: 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.

\( \exists x, y \in \{ 0, 1 \}^k, x \neq y, \) with

\( \delta(q_0, x) = \delta(q_0, y) \)

(pigeon-hole principle)

\( \Rightarrow \exists z \in \{ 0, 1 \}^{k-1} \) such that

\( \delta(q_0, xz) = \delta(q_0, yz) \)

Assume \( xz = \alpha \neq yz = 1 \) and consider inputs

\( x^2 \)

\( y^2 \)

where \( z \in \{ 0, 1 \}^{k-1} \).

Either \( M \) accepts both \( x^2 \) and \( y^2 \) or it accepts

neither (since \( \delta(q_0, xz) = \delta(q_0, yz) = \delta(q_0, yz) = \delta(q_0, yz) \)),

but \( x \in L_k \) while \( y \notin L_k \), a contradiction. \( \square \)
NFA $\Rightarrow$ DFA

Theorem: Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA, and $L = L(N) \subseteq \Sigma^*$. Then there exists a DFA $M$ st. $L = L(M)$.

Proof:

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

$\rho(S, a) = \bigcup \{ \delta(q, a) : q \in S \} \setminus \emptyset$.

Recall the definitions of extended transition functions:

- For an NFA

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

$\hat{\delta}(q, \epsilon) = \{ q \}$

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

Let $D : 2^Q \times \Sigma^* \rightarrow 2^Q$ 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: $|x| = 0$, i.e. $x = \epsilon$

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

Induction step:

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

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

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

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

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

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

$= L(N)$.

$$