DFA Minimization Theorem:

For every regular language $L'$, there is a unique (up to re-labeling of states) minimal-state DFA $M^*$ such that $L(M^*) = L'$.

Furthermore, there is an efficient algorithm which, given any DFA $M$, will output this unique $M^*$. 
Extending transition function $\delta$ to strings

Given $M = (Q, \Sigma, \delta, q_0, F)$, we can extend $\delta$ to a function $\Delta : Q \times \Sigma^* \to Q$ that works on strings:

$\Delta(q, \varepsilon) = q$

$\Delta(q, \sigma) = \delta(q, \sigma)$

$\Delta(q, \sigma_1 \ldots \sigma_{k+1}) = \delta(\Delta(q, \sigma_1 \ldots \sigma_k), \sigma_{k+1})$

$\Delta(q, w) =$ the state of $M$ reached after reading in $w$, starting from state $q$

Note: $\Delta(q_0, w) \in F \iff M$ accepts $w$

Def. $w \in \Sigma^*$ distinguishes states $q_1$ and $q_2$ iff

$\Delta(q_1, w) \in F \iff \Delta(q_2, w) \notin F$
Extending transition function $\delta$ to strings

Given $M = (Q, \Sigma, \delta, q_0, F)$, we can extend $\delta$ to a function \( \Delta : Q \times \Sigma^* \rightarrow Q \) that works on strings:

\[
\Delta(q, \varepsilon) = q \\
\Delta(q, \sigma) = \delta(q, \sigma) \\
\Delta(q, \sigma_1 \ldots \sigma_{k+1}) = \delta(\Delta(q, \sigma_1 \ldots \sigma_k), \sigma_{k+1})
\]

\( \Delta(q, w) \) = the state of $M$ reached after reading in $w$, starting from state $q$

Note: \( \Delta(q_0, w) \in F \iff M \) accepts $w$

Def. $w \in \Sigma^*$ distinguishes states $q_1$ and $q_2$ iff exactly one of $\Delta(q_1, w)$, $\Delta(q_2, w)$ is a final state
Fix $M = (Q, \Sigma, \delta, q_0, F)$ and let $p, q \in Q$

**Definition:**

State $p$ is *distinguishable* from state $q$

iff there is $w \in \Sigma^*$ that distinguishes $p$ and $q$

iff there is $w \in \Sigma^*$ so that

exactly one of $\Delta(p, w)$, $\Delta(q, w)$ is a final state

State $p$ is *indistinguishable* from state $q$

iff $p$ is not distinguishable from $q$

iff for all $w \in \Sigma^*$, $\Delta(p, w) \in F \iff \Delta(q, w) \in F$

*Pairs of indistinguishable states are redundant...*
Fix $M = (Q, \Sigma, \delta, q_0, F)$ and let $p, q, r \in Q$

Define a binary relation $\sim$ on the states of $M$:

- $p \sim q$ iff $p$ is indistinguishable from $q$
- $p \not\sim q$ iff $p$ is distinguishable from $q$

**Proposition:** $\sim$ is an equivalence relation

- $p \sim p$ (reflexive)
- $p \sim q \Rightarrow q \sim p$ (symmetric)
- $p \sim q$ and $q \sim r \Rightarrow p \sim r$ (transitive)
Fix $M = (Q, \Sigma, \delta, q_0, F)$ and let $p, q, r \in Q$

**Proposition:** $\sim$ is an equivalence relation

As a consequence, the relation $\sim$ partitions $Q$ into disjoint equivalence classes

$$[q] := \{ p \mid p \sim q \}$$
Algorithm: MINIMIZE-DFA

Input: DFA $M$

Output: DFA $M_{\text{MIN}}$ such that:

$L(M) = L(M_{\text{MIN}})$

$M_{\text{MIN}}$ has no inaccessible states

$M_{\text{MIN}}$ is irreducible

||

For all states $p \neq q$ of $M_{\text{MIN}}$, $p$ and $q$ are distinguishable

Theorem: $M_{\text{MIN}}$ is the unique minimal DFA that is equivalent to $M$
The Table-Filling Algorithm

Input: DFA $M = (Q, \Sigma, \delta, q_0, F)$

Output: 

1. $D_M = \{ (p, q) \mid p, q \in Q \text{ and } p \neq q \}$
2. $\text{EQUIV}_M = \{ [q] \mid q \in Q \}$

Base Case: For all $(p, q)$ such that $p$ accepts and $q$ rejects $\Rightarrow p \not\sim q$

Iterate: If there are states $p, q$ and symbol $\sigma \in \Sigma$ satisfying:

\[
\delta(p, \sigma) = p' \\
\delta(q, \sigma) = q' \\
\not\sim \Rightarrow p \not\sim q
\]

Repeat until no more $D$'s can be added.
Algorithm MINIMIZE

Input: DFA M

Output: Equivalent minimal-state DFA $M_{\text{MIN}}$

1. Remove all inaccessible states from M

2. Run Table-Filling algorithm on M to get:
   $\text{EQUIV}_M = \{ \left[ q \right] \mid q \text{ is an accessible state of } M \}$

3. Define: $M_{\text{MIN}} = (Q_{\text{MIN}}, \Sigma, \delta_{\text{MIN}}, q_{0 \text{MIN}}, F_{\text{MIN}})$
   
   $Q_{\text{MIN}} = \text{EQUIV}_M$, $q_{0 \text{MIN}} = \left[ q_0 \right]$, $F_{\text{MIN}} = \{ \left[ q \right] \mid q \in F \}$

   $\delta_{\text{MIN}}( \left[ q \right], \sigma ) = \left[ \delta( q, \sigma ) \right]$

Claim: $L(M_{\text{MIN}}) = L(M)$
**Thm:** $M_{\text{MIN}}$ is the **unique** minimal DFA equivalent to $M$

**Claim:** Suppose $L(M') = L(M_{\text{MIN}})$ and $M'$ has no inaccessible states and $M'$ is irreducible. Then **there is an isomorphism** between $M'$ and $M_{\text{MIN}}$

Suppose for now the **Claim** is true. If $M'$ is a minimal DFA, then $M'$ has no inaccessible states and is irreducible (*why?*)

So the **Claim** implies:

*Let $M'$ be a minimal DFA for $M$. Then, there is an isomorphism between $M'$ and the DFA $M_{\text{MIN}}$ that is output by MINIMIZE($M$). Therefore the Thm holds!*
Thm: $M_{\text{MIN}}$ is the unique minimal DFA equivalent to $M$

Claim: Suppose $L(M')=L(M_{\text{MIN}})$ and $M'$ has no inaccessible states and $M'$ is irreducible. Then there is an isomorphism between $M'$ and $M_{\text{MIN}}$

Proof: We recursively construct a map from the states of $M_{\text{MIN}}$ to the states of $M'$

Base Case: $q_{0\text{MIN}} \mapsto q_0'$

Recursive Step: If $p \mapsto p'$

Then $q \mapsto q'$
Base Case: $q_{0 \text{MIN}} \mapsto q_0'$

Recursive Step: If $p \mapsto p'$

Then $q \mapsto q'$
Base Case: $q_{0, \text{MIN}} \mapsto q_0'$

Recursive Step: If $p \mapsto p'$
\[ \sigma \quad \sigma \]
\[ q \quad q' \]

Then $q \mapsto q'$

Goal: Show this is an isomorphism. Need to prove:

The map is **defined** everywhere

The map is **well defined**

The map is a **bijection**

The map **preserves all transitions**: If $p \mapsto p'$ then $\delta_{\text{MIN}}(p, \sigma) \mapsto \delta'(p', \sigma)$

*(this follows from the definition of the map!)*
The map is defined everywhere

That is, for all states \( q \) of \( M_{\text{MIN}} \) there is some state \( q' \) of \( M' \) such that \( q \mapsto q' \)

If \( q \in M_{\text{MIN}} \), there is a string \( w \) such that \( \Delta_{\text{MIN}}(q_{0_{\text{MIN}}},\sigma) = q \) (Why?)

Let \( q' = \Delta'(q_{0'_{\text{MIN}}}w) \). Then \( q \mapsto q' \)

(proof by induction on \( |w| \))
Base Case: \( q_{0, \text{MIN}} \mapsto q_0' \)

Recursive Step: If \( p \mapsto p' \)
\[ \sigma \quad \sigma \]
\[ q \quad q' \]

Then \( q \mapsto q' \)

The map is well defined

*Proof by contradiction.*

Suppose there are states \( q' \) and \( q'' \) such that \( q \mapsto q' \) and \( q \mapsto q'' \)

We show that \( q' \) and \( q'' \) are *indistinguishable*, so it must be that \( q' = q'' \)
Suppose there are states $q'$ and $q''$ such that $q \leftrightarrow q'$ and $q \leftrightarrow q''$

Now suppose $q'$ and $q''$ are distinguishable...

Contradiction!
Base Case: $q_{0_{\text{MIN}}} \mapsto q_0'$

Recursive Step: If $p \mapsto p'$

Then $q \mapsto q'$

The map is onto

Want to show: For all states $q'$ of $M'$ there is a state $q$ of $M_{\text{MIN}}$ such that $q \mapsto q'$

For every $q'$ there is a string $w$ such that $M'$ reaches state $q'$ after reading in $w$

Let $q$ be the state of $M_{\text{MIN}}$ after reading in $w$

Claim: $q \mapsto q'$ (proof by induction on $|w|$)
The map is **one-to-one**

*Proof by contradiction.* Suppose there are states \( p \neq q \) such that \( p \mapsto q' \) and \( q \mapsto q' \)

If \( p \neq q \), then \( p \) and \( q \) are **distinguishable**.
How can we prove that two regular expressions are equivalent?
The Myhill-Nerode Theorem
In DFA Minimization, we defined an equivalence relation between states.

We can also define a similar equivalence relation over strings and languages:

Let $L \subseteq \Sigma^*$ and $x, y \in \Sigma^*$

$x \equiv_L y \iff \text{for all } z \in \Sigma^*, [xz \in L \iff yz \in L]$ 

Define: $x$ and $y$ are indistinguishable to $L$ iff $x \equiv_L y$

Claim: $\equiv_L$ is an equivalence relation

Proof?
Let $L \subseteq \Sigma^*$ and $x, y \in \Sigma^*$

$x \equiv_L y$ iff for all $z \in \Sigma^*$, $[xz \in L \iff yz \in L]$

**The Myhill-Nerode Theorem:**

A language $L$ is regular *if and only if* the number of equivalence classes of $\equiv_L$ is **finite**.

**Proof** ($\Rightarrow$) Let $M = (Q, \Sigma, \delta, q_0, F)$ be a min DFA for $L$.

Define the relation: $x \sim_M y \iff \Delta(q_0, x) = \Delta(q_0, y)$

**Claim:** $\sim_M$ is an equivalence relation with $|Q|$ classes

**Claim:** If $x \sim_M y$ then $x \equiv_L y$

**Proof:** $x \sim_M y$ implies for all $z \in \Sigma^*$, $xz$ and $yz$ reach the *same state* of $M$. So $xz \in L \iff yz \in L$, and $x \equiv_L y$

**Corollary:** Number of equiv. classes of $\equiv_L$ is *at most* the number of equiv. classes of $\sim_M$ (which is $|Q|$)
Let $L \subseteq \Sigma^*$ and $x, y \in \Sigma^*$

$x \equiv_L y \text{ iff for all } z \in \Sigma^*, [xz \in L \iff yz \in L]$

$(\iff)$ If the number of equivalence classes of $\equiv_L$ is $k$
then there is a DFA for $L$ with $k$ states

**Idea:** Build a DFA using equivalence classes of $\equiv_L$!

Define a DFA $M$ where

- $Q$ is the set of equivalence classes of $\equiv_L$
- $q_0 = [\epsilon] = \{y \mid y \equiv_L \epsilon\}$
- $\delta([x], \sigma) = [x \sigma]$
- $F = \{[x] \mid x \in L\}$

**Claim:** $M$ accepts $x$ if and only if $x \in L$
The Myhill-Nerode Theorem gives us a new way to prove that a given language is not regular:

L is not regular if and only if there are infinitely many equiv. classes of \( \equiv_L \)

L is not regular if and only if

There are infinitely many strings \( w_1, w_2, \ldots \) so that for all \( w_i \neq w_j \), \( w_i \) and \( w_j \) are distinguishable to \( L \):

there is a \( z \in \Sigma^* \) such that

\( \text{exactly one of } w_i \text{ and } w_j \) is in \( L \)

Distinguishing set for \( L \)
The Myhill-Nerode Theorem gives us a new way to prove that a given language is not regular:

**Theorem:** \( L = \{0^n 1^n \mid n \geq 0\} \) is not regular.

**Proof:** Consider the infinite set of strings 
\[ S = \{0, 00, 000, \ldots, 0^n, \ldots\} \]

Take any pair \((0^m, 0^n)\) of distinct strings in \(S\)

Let \( z = 1^m \)

Then \(0^m 1^m\) is in \(L\), but \(0^n 1^m\) is not in \(L\)

That is, all pairs of strings in \(S\) are distinguishable

Hence there are infinitely many equivalence classes of \(\equiv_L\), and \(L\) is not regular.