CS154
Finishing up DFA Minimization, The Myhill-Nerode Theorem, and Streaming Algorithms
Theorem

For every regular language L, there is a unique (up to re-labeling of the states) minimal-state DFA $M^*$ such that $L = L(M^*)$. Furthermore, there is an efficient algorithm which, given any DFA $M$, will output this unique $M^*$. 
Extending the transition function $\delta$

Given DFA $M = (Q, \Sigma, \delta, q_0, F)$, we extend $\delta$ to a function $\Delta : Q \times \Sigma^* \rightarrow Q$ as follows:

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

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 the transition function $\delta$

Given DFA $M = (Q, \Sigma, \delta, q_0, F)$, we extend $\delta$ to a function $\Delta : Q \times \Sigma^* \rightarrow Q$ as follows:

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

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) $D_M = \{ (p, q) \mid p, q \in Q \text{ and } p \not\sim q \}$

(2) $\text{EQUIV}_M = \{ [q] \mid q \in Q \}$

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

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

Output:  
(1) $D_M = \{ (p, q) \mid p, q \in Q \text{ and } p \not\sim q \}$

(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:

$\delta(p, \sigma) = p'$

$\not\sim \Rightarrow p \not\sim q$

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

Repeat until no more D’s can be added.
Claim: If \((p, q)\) is marked \(D\) by the Table-Filling algorithm, then \(p \not\sim q\)
Claim: If \((p, q)\) is not marked D by the Table-Filling algorithm, then \(p \sim q\)

Proof (by contradiction):
Suppose the pair \((p, q)\) is not marked D by the algorithm, yet \(p \not\sim q\) (call this a “bad pair”)
Of all such bad pairs, let \(p, q\) be a pair with the shortest distinguishing string \(w\)
\[ \Delta(p, w) \in F \text{ and } \Delta(q, w) \not\in F \] (Why is \(|w| > 0\)?)
We have \(w = \sigma w'\), for some string \(w'\) and some \(\sigma \in \Sigma\)
Let \(p' = \delta(p, \sigma)\) and \(q' = \delta(q, \sigma)\)

Then \((p', q')\) is also a bad pair, but with a SHORTER \(w'\)!
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:
   $\text{EQUIV}_M = \{ [q] \mid q \text{ is an accessible state of } M \}$
3. Define: $M_{\text{MIN}} = (Q_{\text{MIN}}, \Sigma, \delta_{\text{MIN}}, q_{0_{\text{MIN}}}, F_{\text{MIN}})$
   $Q_{\text{MIN}} = \text{EQUIV}_M$, $q_{0_{\text{MIN}}} = [q_0]$, $F_{\text{MIN}} = \{ [q] \mid q \in F \}$
   $\delta_{\text{MIN}}([q], \sigma) = [\delta(q, \sigma)]$

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

*If $M'$ is a minimal DFA for $M$, then there is an isomorphism between $M'$ and $M_{\text{MIN}}$.*

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: $q_{0\text{MIN}} \mapsto q_0^\prime$

Recursive Step: If $p \mapsto p'$
\[\sigma \quad \sigma\]
Then $q \mapsto q'$
Base Case: \( q_{0\text{MIN}} \mapsto q_0' \)

Recursive Step: If \( p \mapsto p' \)

\[ \begin{array}{c}
\downarrow \sigma \\
q \\
\end{array} \quad \begin{array}{c}
\downarrow \sigma \\
q' \\
\end{array} \quad \text{Then } q \mapsto q' \]

We need to prove:

- The map is defined everywhere
- The map is well defined
- The map is a bijection
- The map preserves all transitions:
  \( \text{If } p \mapsto p' \text{ then } \delta_{\text{MIN}}(p, \sigma) \mapsto \delta'(p', \sigma) \)  
  \( (this \text{ follows from the definition of the map!}) \)
Base Case: \( q_{0, \text{MIN}} \mapsto q_0' \)

Recursive Step: \( \text{If } p \mapsto p' \)

\[ \begin{array}{c}
\sigma \\
\downarrow \\
q \\
\sigma \\
\downarrow \\
q' \\
\end{array} \]

Then \( q \mapsto q' \)

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

\[ \Delta_{\text{MIN}}(q_{0, \text{MIN}}, w) = q \] (Why?)

Let \( q' = \Delta'(q_0', w) \). Then \( q \mapsto q' \)
The map is well defined
Suppose there are states $q'$ and $q''$ such that $q \rightarrow q'$ and $q \rightarrow q''$

We show that $q'$ and $q''$ are indistinguishable, so it must be that $q' = q''$

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

Recursive Step: If $p \leftrightarrow p'$

\[
\begin{array}{c}
\sigma \\
q
\end{array}
\begin{array}{c}
\sigma \\
q'
\end{array}
\quad \text{Then} \quad q \rightarrow q'
\]

$\begin{array}{c}
\sigma \\
q
\end{array}$

$\begin{array}{c}
\sigma \\
q'
\end{array}$
Suppose there are states $q'$ and $q''$ such that $q \xrightarrow{} q'$ and $q \xrightarrow{} q''$

Now suppose $q'$ and $q''$ are distinguishable...
Base Case: $q_{0_{MIN}} \rightarrow q_0'$

Recursive Step: If $p \rightarrow p'$

\[
\begin{array}{c}
\downarrow \sigma \\
q \\
\end{array} \quad \begin{array}{c}
\downarrow \sigma \\
q' \\
\end{array} \quad \text{Then } q \rightarrow q'
\]

The map is onto

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

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

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

Claim: $q \rightarrow q'$
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.

\[
\begin{array}{c|c|c}
\text{M}_{\text{MIN}} & \text{M}' & \\
\hline
q_0 \quad \text{MIN} & p & u \quad \text{Accept} \\
& v \quad \text{Reject} & \\
q_0 \quad \text{MIN} & q & u \quad \text{Accept} \\
& v \quad \text{Reject} & \\
& w \quad \text{Reject} & u \quad \text{Accept} \\
& w \quad \text{Reject} & \\
q_0' & q' & \\
& v & w \quad \text{Reject} \\
& w \quad \text{Reject} &
\end{array}
\]
How can we prove that two regular expressions are equivalent?
The Myhill-Nerode Theorem
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 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?
The Myhill-Nerode Theorem:
A language $L$ is regular if and only if the number of equivalence classes of $\equiv_L$ is finite.

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

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

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 \( x, y \in \Sigma^* \)

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

(\( \Leftarrow \)) 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 with these equivalence classes!
Define a DFA \( M \) where

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

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

*exactly one* of \( w_i z \) and \( w_j z \) is in \( L \)
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
\[
S = \{ 0, 0^1, 0^2, \ldots, 0^n, \ldots \}
\]
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.
Streaming Algorithms
Streaming Algorithms
L = \{x \mid x \text{ has more 1’s than 0’s}\}

Initialize C := 0 and B := 0
Read the next bit x from the stream
If (C = 0) then B := x, C := 1
If (C ≠ 0) and (B = x) then C := C + 1
If (C ≠ 0) and (B ≠ x) then C := C – 1
When the stream stops, accept
    if and only if B=1 and C > 0

B = the majority bit
C = how many more times that B appears

On all strings of length n, the algorithm uses \((1+\log_2 n)\) bits of space (to store B and C)