How can we prove that two regular expressions are equivalent?

How can we prove that two DFAs (or two NFAs) are equivalent?

How can we prove that two regular languages are equivalent? (Does this question make sense?)
How can we prove that two DFAs (or two NFAs) are equivalent?
MINIMIZING DFAs
IS THIS MINIMAL? NO
IS THIS MINIMAL?
THEOREM

For every regular language $L$, there exists a **UNIQUE** (up to re-labeling of the states) minimal DFA $M$ such that $L = L(M)$

Minimal means wrt number of states

Given a specification for $L$, via DFA, NFA or regex, this theorem is constructive.
NOT TRUE FOR NFAs
NOT TRUE FOR RegExp
EXTENDING $\delta$

Given DFA $M = (Q, \Sigma, \delta, q_0, F)$ extend $\delta$ to $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$ as follows:

\[
\begin{align*}
\hat{\delta}(q, \varepsilon) &= q \\
\hat{\delta}(q, \sigma) &= \delta(q, \sigma) \\
\hat{\delta}(q, \sigma_1 \ldots \sigma_{k+1}) &= \delta(\hat{\delta}(q, \sigma_1 \ldots \sigma_k), \sigma_{k+1})
\end{align*}
\]

Note: $\hat{\delta}(q_0, w) \in F \iff M$ accepts $w$

String $w \in \Sigma^*$ distinguishes states $p$ and $q$ iff

$\hat{\delta}(p, w) \in F \iff \hat{\delta}(q, w) \notin F$
EXTENDING $\delta$

Given DFA $M = (Q, \Sigma, \delta, q_0, F)$ extend $\delta$ to $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$ as follows:

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

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

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

Note: $\hat{\delta}(q_0, w) \in F \iff M$ accepts $w$

String $w \in \Sigma^*$ distinguishes states $p$ and $q$ iff exactly ONE of $\hat{\delta}(p, w), \hat{\delta}(q, w)$ is a final state
Fix $M = (Q, \Sigma, \delta, q_0, F)$ and let $p, q \in Q$

**DEFINITION:**

$p$ is *distinguishable* from $q$ iff

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

$p$ is *indistinguishable* from $q$ iff

for all $w \in \Sigma^*$, $\delta(p, w) \in F \iff \delta(q, w) \in F$
Fix $M = (Q, \Sigma, \delta, q_0, F)$ and let $p, q \in Q$

**DEFINITION:**

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

$p$ is *indistinguishable* from $q$
iff
$p$ is not distinguishable from $q$
iff
for all $w \in \Sigma^*$, $\delta(p, w) \in F \iff \delta(q, w) \in F$
\( \varepsilon \) distinguishes accept from non-accept states
Fix $M = (Q, \Sigma, \delta, q_0, F)$ and let $p, q, r \in Q$

Define relation $\sim$:

$p \sim q$ 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)

Proof (of transitivity): for all $w$, we have:

$\delta(p, w) \in F \iff \delta(q, w) \in F \iff \delta(r, w) \in F$
Fix $M = (Q, \Sigma, \delta, q_0, F)$ and let $p, q, r \in Q$

so $\sim$ partitions the set of states of $M$ into disjoint equivalence classes

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)

Proof (of transitivity): for all $w$, we have:

$\delta(p, w) \in F \iff \delta(q, w) \in F \iff \delta(r, w) \in F$
Fix $M = (Q, \Sigma, \delta, q_0, F)$ and let $p, q, r \in Q$

so $\sim$ partitions the set of states of $M$ into disjoint equivalence classes.

Proposition: $\sim$ is an equivalence relation

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

Input: DFA \( M \)

Output: DFA \( M_{\text{MIN}} \) such that:

\[ M \equiv M_{\text{MIN}} \text{ (that is, } L(M) = L(M_{\text{MIN}})) \]

\( M_{\text{MIN}} \) has no inaccessible states

\( M_{\text{MIN}} \) is *irreducible*

\[ \text{all states of } M_{\text{MIN}} \text{ are pairwise distinguishable} \]

Theorem: \( M_{\text{MIN}} \) is the *unique* minimum DFA equivalent to \( M \)

*up to relabelling
NOTE: Theorem not true for NFAs

What does this say about Regexs?
Intuition for Algorithm:
States of $M_{\text{MIN}}$ will be blocks of equivalent states of $M$

We’ll find these equivalent states with a “Table-Filling” Algorithm
**TABLE-FILLING ALGORITHM**

**Input:** \( \text{DFA } M = (Q, \Sigma, \delta, q_0, F) \)

**Output:**
1. \( D_M = \{ (p,q) | p,q \in Q \text{ and } p \neq q \} \)
2. \( E_M = \{ [q] | q \in Q \} \)

**IDEA:**
- We know how to find those pairs of states that \( \varepsilon \) distinguishes…
- Use this and recursion to find those pairs distinguishable with *longer* strings
- Pairs left over will be indistinguishable
**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. $E_M = \{ [q] \mid q \in Q \}$

**Base Case:** $p$ accepts and $q$ “rejects” $\Rightarrow p \nRightarrow q$
**TABLE-FILLING ALGORITHM**

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

**Output:**
1. $D_M = \{ (p,q) | p, q \in Q \text{ and } p \not\rightarrow q \}$
2. $E_M = \{ [q] | q \in Q \}$

**Base Case:** $p$ accepts and $q$ "rejects" $\Rightarrow p \not\sim q$

**Recursion:**

$p \xrightarrow{\sigma} p' \not\rightarrow q' \Rightarrow p \not\rightarrow q$

Repeat until no more new D's
Claim: If \( p, q \) are distinguished by Table-Filling algorithm (ie pair labelled by \( D \)), then \( p \sim q \)

Proof: By induction on the stage of the algorithm

Claim: If \( p, q \) are not distinguished by Table-Filling algorithm, then \( p \sim q \)

Proof (by contradiction):
**Claim:** If \( p, q \) are distinguished by Table-Filling algorithm (ie pair labelled by \( D \)), then \( p \not\sim q \)

**Proof:** By induction on the stage of the algorithm

If \( (p, q) \) is marked \( D \) at the start, then one’s in \( F \) and one isn’t, so \( \epsilon \) distinguishes \( p \) and \( q \)

Suppose \( (p, q) \) is marked \( D \) at stage \( n+1 \)

Then there are states \( p', q' \), string \( w \in \Sigma^* \) and \( \sigma \in \Sigma \) such that:

1. \( (p', q') \) are marked \( D \) \( \Rightarrow \) \( p' \not\sim q' \) (by induction)
   \( \Rightarrow \) \( \delta(p', w) \in F \) and \( \delta(q', w) \notin F \)

2. \( p' = \delta(p, \sigma) \) and \( q' = \delta(q, \sigma) \)

The string \( \sigma w \) distinguishes \( p \) and \( q \)!
Claim: If $p, q$ are not distinguished by Table-Filling algorithm, then $p \sim q$

Proof (by contradiction):

Suppose the pair $(p, q)$ is not marked $D$ by the algorithm, yet $p \not\sim q$ (a “bad pair”)

Suppose $(p, q)$ is a bad pair with the shortest $w$.

$$\delta(p, w) \in F \quad \text{and} \quad \delta(q, w) \notin F \quad \text{(Why is } |w| > 0 \text{ ?)}$$

So, $w = \sigma w'$, where $\sigma \in \Sigma$

Let $p' = \delta(p, \sigma)$ and $q' = \delta(q, \sigma)$

Then $(p', q')$ cannot be marked $D$ (Why?)

But $(p', q')$ is distinguished by $w'$!

So $(p', q')$ is also a bad pair, but with a SHORTER $w'$!

Contradiction!
Algorithm MINIMIZE

Input: DFA $M$

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

(1) Remove all inaccessible states from $M$

(2) Apply Table-Filling algorithm to get:
$E_M = \{ [q] \mid q \text{ is an accessible state of } M \}$

Define: $M_{\text{MIN}} = (Q_{\text{MIN}}, \Sigma, \delta_{\text{MIN}}, q_{0 \text{ MIN}}, F_{\text{MIN}})$

$Q_{\text{MIN}} = E_M$, $q_{0 \text{ MIN}} = [q_0]$, $F_{\text{MIN}} = \{ [q] \mid q \in F \}$

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

Must show $\delta_{\text{MIN}}$ is well defined!
Algorithm MINIMIZE

Input: DFA $M$

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

(1) Remove all inaccessible states from $M$

(2) Apply Table-Filling algorithm to get:
$E_M = \{ [q] \mid q \text{ is an accessible state of } M \}$

Define: $M_{\text{MIN}} = (Q_{\text{MIN}}, \Sigma, \delta_{\text{MIN}}, q_{0 \text{MIN}}, F_{\text{MIN}})$

$Q_{\text{MIN}} = E_M$, $q_{0 \text{MIN}} = [q_0]$, $F_{\text{MIN}} = \{ [q] \mid q \in F \}$

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

Claim: $\delta_{\text{MIN}}([q], w) = [\hat{\delta}(q, w)]$, $w \in \Sigma^*$
Algorithm MINIMIZE

Input: DFA $M$
Output: DFA $M_{\text{MIN}}$

(1) Remove all inaccessible states from $M$

(2) Apply Table-Filling algorithm to get: $E_M = \{ [q] | q \text{ is an accessible state of } M \}$

Define: $M_{\text{MIN}} = (Q_{\text{MIN}}, \Sigma, \delta_{\text{MIN}}, q_{0 \text{MIN}}, F_{\text{MIN}})$

$Q_{\text{MIN}} = E_M$, $q_{0 \text{MIN}} = [q_0]$, $F_{\text{MIN}} = \{ [q] | q \in F \}$

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

So: $\delta_{\text{MIN}}([q_0], \sigma) = [\delta(q_0, \sigma)]$
Algorithm MINIMIZE

Input: DFA $M$

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

(1) Remove all inaccessible states from $M$

(2) Apply Table-Filling algorithm to get:
$E_M = \{ [q] \mid q \text{ is an accessible state of } M \}$

Define: $M_{\text{MIN}} = (Q_{\text{MIN}}, \Sigma, \delta_{\text{MIN}}, q_{0_{\text{MIN}}}, F_{\text{MIN}})$

$Q_{\text{MIN}} = E_M$, $q_{0_{\text{MIN}}} = [q_0]$, $F_{\text{MIN}} = \{ [q] \mid q \in F \}$

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

Follows: $M_{\text{MIN}} \equiv M$
MINIMIZE

Graph with states q₀, q₁, q₂ and transitions:
- q₀ → q₁ on input 0
- q₁ → q₁ on input 0
- q₁ → q₀ on input 1
- q₀ → q₂ on input 0
- q₂ → q₁ on input 1
- q₂ → q₂ on input 1

PROPOSITION. Suppose $M' \equiv M$ and $M'$ has no inaccessible states and is irreducible.

Then, there exists a 1-1 onto correspondence between $M_{\text{MIN}}$ and $M'$ (preserving transitions).

**COR:** $M_{\text{MIN}}$ is unique minimal DFA $\equiv M$.
PROPOSITION. Suppose $M' \equiv M$ and $M'$ has no inaccessible states and is irreducible.

Then, there exists a 1-1 onto correspondence between $M_{\text{MIN}}$ and $M'$ (preserving transitions)

i.e., $M_{\text{MIN}}$ and $M'$ are “Isomorphic”

COR: $M_{\text{MIN}}$ is unique minimal DFA $\equiv M$

Proof of Prop: We will construct a map recursively

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

Recursive Step: If $p \rightarrow p'$

Then $q \rightarrow q'$
PROPOSITION. Suppose $M' \equiv M$ and $M'$ has no inaccessible states and is irreducible.

Then, there exists a 1-1 onto correspondence between $M_{\text{MIN}}$ and $M'$ (preserving transitions)

\[ \text{i.e., } M_{\text{MIN}} \text{ and } M' \text{ are "isomorphic"} \]

COR: $M_{\text{MIN}}$ is unique minimal DFA $\equiv M$

Proof of Prop: We will construct a map recursively

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

Recursive Step: If $p \rightarrow p'$

and $\delta(p, \sigma) = q$ and $\delta(p', \sigma) = q'$ Then $q \rightarrow q'$
We need to show:

- The map is **everywhere defined**
- The map is **well defined**
- The map is a **bijection (1-1 and onto)**
- The map **preserves transitions**
Base Case: $q_{0MIN} \rightarrow q_0'$

Recursive Step: If $p \rightarrow p'$

Then $q \rightarrow q'$

The map is everywhere defined:
That is, for all $q \in M_{MIN}$
there is a $q' \in M'$ such that $q \rightarrow q'$

If $q \in M_{MIN}$, there is a string $w$ such that

$\delta_{MIN}(q_{0MIN}, w) = q$ \text{ (WHY?)}

Let $q' = \delta(q_0', w)$. $q$ will map to $q'$ (by induction)
Base Case: $q_{0 \text{MIN}} \rightarrow q_0'$

Recursive Step: If $p \rightarrow p'$

\[ \sigma \downarrow \sigma \downarrow \]

Then $q \rightarrow q'$

$q$ $q'$

The map is well defined
That is, for all $q \in M_{\text{MIN}}$
there is at most one $q' \in M'$ such that $q \rightarrow q'$

Suppose there exist $q'$ and $q''$ such that $q \rightarrow q'$ and $q \rightarrow q''$

We show that $q'$ and $q''$ are indistinguishable, so it must be that $q' = q''$ (Why?)
Suppose there exist $q'$ and $q''$ such that $q \rightarrow q'$ and $q \rightarrow q''$

Suppose $q'$ and $q''$ are distinguishable

Contradiction!
The map is 1-1

Suppose there are distinct p and q such that p → q′ and q → q′

p and q are distinguishable (why?)

The map is 1-1

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

Recursive Step: If $p \rightarrow p'$

\[ \begin{align*}
\sigma & \downarrow \\
q & \rightarrow \\
q' & \downarrow \\
\sigma & \downarrow \sigma
\end{align*} \]

Then $q \rightarrow q'$

The map is onto
That is, for all $q' \in M'$ there is a $q \in M_{\text{MIN}}$ such that $q \rightarrow q'$

If $q' \in M'$, there is $w$ such that $\delta'(q_0', w) = q'$

Let $q = \delta_{\text{MIN}}^\wedge(q_{0 \text{ MIN}}, w)$. $q$ will map to $q'$ (why?)
Base Case: $q_0_{\text{MIN}} \rightarrow q_0'$

Recursive Step: If $p \rightarrow p'$

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

Then $q \rightarrow q'$

The map preserves transitions

That is, if $p \rightarrow p'$ and $q \rightarrow q'$ and $\delta(p, \sigma) = q$

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

(Why?)
How can we prove that two regular expressions are equivalent?
WWW.FLAC.WS

Read Chapters 2.1 & 2.2 for next time