Welcome to
Cpts 317

Wed Jan 29
(aka HW1 due date)
EQUIVALENCE OF NFAs and DFAs

We say two machines are equivalent if they recognize the same language.

THEOREM

Every nondeterministic finite automaton has an equivalent deterministic finite automaton.
Prop. Idea

1. Convert the NFA that recognizes the language into an equivalent DFA that simulates the NFA.

2. What do you need to keep track of as the input string is processed?

3. If \( k \) is the number of states of the NFA, it has subsets of states.
Proof

Let $N = (Q, \Sigma, \delta, q_0, F)$ be the NFA recognizing some language $A$.

We construct a DFA

$M = (Q', \Sigma, \delta', q_0', F)$

recognizing $A$.

We will first consider the case where $N$ has no $E$-transitions.
1. $Q' = \mathcal{P}(Q)$

2. For $R \in Q'$ and $a \in \Sigma$, let

$$S'(R, a) = \{ q \in Q | \exists s \in S(r, a) \text{ for some } r \in R \}.$$ 

Another way to write this is

$$S'(R, a) = \bigcup_{r \in R} S(r, a)$$

3. $q'_0 = \{ q_0 \}$

4. $F' = \{ R \in Q' | R \text{ contains an accept state of } N \}$. 
Now we need to consider the \( E \) arrows. We set up an extra list of notation.

For any state \( q \) of \( M \), we define \( E(q) \) to be the collection of states that can be reached from members of \( R \) by going along \( E \) arrows, including the member of \( R \) themselves.

Formally, for \( R \subseteq Q \) let

\[
E(q) = \{ s \in Q \mid s \text{ can be reached from } q \text{ by traveling along } 0 \text{ or more } E \text{ arrows} \}
\]

Then we modify the transition function of \( M \) to use \( E(q) \).
\[ \delta'(q_0, a) = \{ q \in Q | q \in E(\delta(r, a)) \} \text{ for some } r \in R \]

Additionally, we need to modify the start state of \( M \).

Changing \( q_0 \) to be \( E(\{q_0\}) \) achieves this.

We have now completed the construction of the DFA \( M \) that simulates the NFA \( N \).
Example

Illustrating converting an NFA to a DFA.

NFA $N_4$

![Diagram of NFA]

$N_4 = (Q, \{a, b\}, \delta, 1, \{4\})$

$Q = \{2, 3\}$

Goal: to construct a DFA $D$ equivalent to $N_4$. 
First, let us determine D's states.

Since \( N_4 \) has three states, we construct \( D \) with 8 states, one for each subset of \( N_4 \)'s states.

Thus D's state set is:

\[
\{ \emptyset, \{1,2\}, \{1,3\}, \{1,3\}, \{1,2,3\} \}
\]

Next, we determine the start and accept states of D.
* Start state is
  \[ E(\{1,2,3\}) \cup \{1\} \]

- An \( \varepsilon \) arrow goes from 2 to 3,

- so \( E(\{1\}) = \varepsilon \{1,2,3\} \).

* New Accept States are those
  Accepting \( N \)'s Accept States;
  thus \( \{\{1\}, \{1,2\}, \{2,3\}, \{1,2,3\}\} \).

\( D \)'s transition function

- Each of \( D \)'s states goes to one
  Place on input an \( B \) one place on
  input \( B \).
A few examples of transitions:

- In D:
  - State \{2\} goes to \{2, 3\} on input a because in N4, State 2 goes to both 2 and 3 on input a and we can't go further from 2 or 3 along e arrows.
  - State \{2\} goes to \{2, 3\} on input b because in N4, State 2 goes only to State 3 on input b.
state \{1\} goes to \(\emptyset\) on a because no a arrows exit it.

state \{1\} goes to \{2\} on b.

state \{3\} goes to \{1,3\} on a because in \(N_4\), state 3 goes to 1 on a and 1 in turn goes to 3 with an E-arrows.

state \{3\} on b goes to \(\emptyset\).

state \{1,3\} on a goes to \{2,3\} because 1 points at no states with a arrows, 2 points at both 2 & 3 with a arrows.
State \( \{1, 2\} \) on \( b \) goes to \( \{2, 3\} \).

Continuing in this manner, we obtain the following diagram.

Simplifying
Closure under regular operators

- Union
- Concatenation
- Star

Union (we saw a proof earlier with DFA & we will see an alternative using NFA).
Proof Idea

- Have regular languages $A_1$ & $A_2$ with $A_1 \cup A_2$ is regular.

- Take two NFAs $N_1$ & $N_2$ for $A_1$ & $A_2$ and combine into one new NFA, $N$.

- $N$ must accept its input if either $N_1$ or $N_2$ accepts this input.

- $N$ has a new start state that branches into the start states of the old machines with $E$ arrows.
In this way, the new machine non-deterministically guesses which of the two machines accepts the input. If one of them accepts the input, \( N \) will accept it, too.

Pictorially:

- \( N_1 \)
- \( N_2 \)
- \( N \)
Proof

Let $N_1 = (Q_1, \Sigma, \delta_1, q_0, F_1)$ recognize $A_1$

$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognize $A_2$

Constant $N = (Q, \Sigma, \delta, q_0, F)$

This to recognize $A_1 \cup A_2$

1. $Q = \{q_0\} \cup Q_1 \cup Q_2$
2. The state $q_0$ is the start state of $N$
3. The set of accept states, $F = F_1 \cup F_2$
4. Define $\delta$ so that for any $q \in Q$ and any $a \in \Sigma$,
\[ S(g, a) = \begin{cases} 
S_1(g, a) & g \in Q_1 \\
S_2(g, a) & g \in Q_2 \\
\{g_1, g_2\} & \exists \; g = g_0 \land a = \varepsilon \\
\emptyset & g = \varepsilon \text{ and } a \neq \varepsilon 
\end{cases} \]
We now prove the following theorem, which started the conversation about nondeterminism.

**Theorem**

The class of regular languages is closed under the concatenation operation.
Proof Idea

* Have regular languages $A_1$ & $A_2$ that $A_1 \circ A_2$ is regular.

* Take two NFAs $N_1$ & $N_2$ for $A_1$ & $A_2$, and combine them into a new NFA $N$. 

---

$N_1$ 

$N_2$ 

$N$
Proof

Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$

Recognize $A_1$ &

$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$

Recognize $A_2$

Construct

$N = (Q, \Sigma, \delta, q_1, F_2)$

to recognize $A_1 \circ A_2$

1. $Q = Q_1 \cup Q_2$

2. The state $q_1$ is the same as
   the start state of $N_1$

3. The accept states $F_2$ are the same as
   the accept states of $N_2$

4. Define $\delta$ so that
for any $q \in Q$ and any $a \in \Sigma$, 

$$
\delta(q, a) = \begin{cases} 
S_n(q, a) & q \in Q_1 \text{ and } q \in F_1 \\
S_1(q, a) & q \in F_1 \text{ and } a \neq \epsilon \\
S_1(q, a) \cup \{q_2\} & q \in F_1 \text{ and } a = \epsilon \\
S_2(q, a) & q \in Q_2 
\end{cases} 
$$
Our last theorem for this discussion

**Theorem**

The class of regular languages is closed under the star operation.
Proof Idea

* Have a regular language $A_1$ and want to prove that $A_1^*$ also is regular.

* Take NFA $N_1$ for $A_1$ and modify it to recognize $A_1^*$. 

![Diagram of NFAs](image-url)
Proof

Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize $A_1$.

Construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize $A_1^*$

1. $Q = \{q_0\} \cup Q_1$
2. The state $q_0$ is the new start state.
3. $F = \{q_0\} \cup F_1$
4. Define $\delta$ so that for any $q \in Q$ and any $a \in \Sigma$, 

\[ \delta(q, a) \]
\[ S(\tau, a) = \begin{cases} 
\delta_1(\tau, a) & 9 \in \mathbb{Q}, \text{ and } 9 \neq 9 \\
\delta_1(\tau, a) & 9 \in F_1, \text{ and } a \neq 3 \\
\delta_1(\tau, a) \cup \{9, \} & 9 \in F_1, \text{ and } a = 3 \\
\{9, \} & 9 = \tau_0 \text{ and } a = 3 \\
\emptyset & 9 = \tau_0 \text{ and } a \neq 3 
\end{cases} \]