Nondeterminism

(Deterministic) FA required for every state $q$ and every symbol $\sigma$ of the alphabet to have exactly one arrow out of $q$ labeled $\sigma$. What happens when we drop this requirement?

$$L = \{ w \in \{a,b\}^* \mid w \text{ contains abba as a substring} \}$$

Transition diagram (of this new type):
A **nondeterministic finite automaton** (NFA) is a 5-tuple \((Q,\Sigma,q_0,A,\delta)\) where

- \(Q\) is a finite set of states
- \(\Sigma\) is a finite alphabet (input symbols)
- \(q_0 \in Q\) is the initial state
- \(A \subseteq Q\) is a set of accepting states
- \(\delta : Q \times \Sigma \rightarrow 2^Q\) is the transition function

\(\delta\) maps each state and input symbol to a subset of states, representing the possible next states. This subset is given by the powerset of \(Q\) denoted as \(2^Q\) or \(\mathcal{P}(Q)\).
Defining the computation of an NFA $M = (Q, \Sigma, q_0, A, \delta)$.

**Extended transition function** $\delta^* : Q \times \Sigma^* \rightarrow 2^Q$:

1) For every $q \in Q$, let $\delta^*(q, \Lambda) = \{ q \}$

2) For every $q \in Q$, $y \in \Sigma^*$, and $\sigma \in \Sigma$, let $\delta^*(q, y\sigma) = \bigcup_{r \in \delta^*(q, y)} \delta(r, \sigma)$

We say that a string $x \in \Sigma^*$ is **accepted by** $M$ if $\delta^*(q_0, x) \cap A \neq \emptyset$

A string which is not accepted by $M$ is **rejected by** $M$.

The **language accepted by** $M$, denoted by $L(M)$, is the set of all strings accepted by $M$. 

[Section 4.1]
NFA’s vs. FA’s

Is the following statement true? **YES**

For every FA $M = (Q, \Sigma, q_0, A, \delta)$ there exists an NFA $M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$ such that $L(M) = L(M_1)$.

**Construction**

$Q_1 = Q$
$q_1 = q_0$
$A_1 = A$

$\delta_1(q_1, \sigma) = \{ \delta(q, \sigma) \}$ for $\forall q \in Q$ $\forall \sigma \in \Sigma$

must return a set of states

returns a single state
NFA’s vs. FA’s

Is the following statement true? Yes.

For every NFA $M = (Q, \Sigma, q_0, A, \delta)$ there exists an FA $M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$ such that $L(M) = L(M_1)$.

Example:

DFA:

NFA:

[Section 4.1]
Thm: For every NFA $M = (Q, \Sigma, q_0, A, \delta)$ there exists an FA $M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$ such that $L(M) = L(M_1)$.

Construction of $M_1$: 

- $Q_1 = 2^Q$
- $q_1 = \{q_0\}$
- $A_1 = \{S \subseteq Q | S \cap A \neq \emptyset\}$
- $\delta_1(s, \sigma) = \bigcup_{r \in S} \delta_1(r, \sigma)$ $\forall s \subseteq Q \forall \sigma \in \Sigma$ ($\ast$)

Proof that $L(M) = L(M_1)$:

CLAIM: $\delta_1^*(q_1, x) = \delta^*(q_0, x) \quad \forall x \in \Sigma^*$

PF by induction on $|x|$

BASE: $\delta_1^*(q_1, \lambda) = q_1 \quad \delta^*(q_0, \lambda) = \{q_0\}$

IND: IH: $\delta_1^*(q_1, x) = \delta^*(q_0, x)$

$\delta_1^*(q_1, x \sigma) \overset{\text{DEF}}{=} \delta_1(\delta_1^*(q_1, x), \sigma) \overset{\text{IH}}{=} \delta_1(\delta^*(q_0, x), \sigma) \overset{\text{DEF}}{=} \bigcup_{r \in q_0} \delta(r, \sigma) \leftarrow \text{LHS}$

$\delta^*(q_0, x \sigma) \overset{\text{DEF}}{=} \bigcup_{r \in \delta^*(q_0, x)} \delta(r, \sigma) \leftarrow \text{RHS}$

\[D\]
Example: NFA accepting $a^*b^*c^*$. 

NFA's with $\Lambda$
A **nondeterministic finite automaton with \( \Lambda \)-transitions** (NFA-\( \Lambda \)) is a 5-tuple \((Q, \Sigma, q_0, A, \delta)\) where

- \( Q \) is a finite set of states
- \( \Sigma \) is a finite alphabet (input symbols)
- \( q_0 \in Q \) is the initial state
- \( A \subseteq Q \) is a set of accepting states
- \( \delta : Q \times (\Sigma \cup \{\Lambda\}) \rightarrow 2^Q \) is the transition function

Give a formal definition of the NFA-\( \Lambda \) from the previous slide.
Formal definition of an NFA-$\Lambda$

Defining the computation of an NFA-$\Lambda$  $M=(Q,\Sigma,q_0,A,\delta)$.

Extended transition function $\delta^*$ : $Q \times \Sigma^* \rightarrow 2^Q$:

1) For every $q \in Q$, let $\delta^*(q,\Lambda) = \text{"all states reachable on } \Lambda \text{"}$

2) For every $q \in Q$, $y \in \Sigma^*$, and $\sigma \in \Sigma$, let $\delta^*(q,y\sigma) =$
Defining the computation of an NFA-$\Lambda$ $M=(Q,\Sigma,q_0,A,\delta)$.

**Extended transition function** $\delta^* : Q \times \Sigma^* \rightarrow 2^Q$:

1) For every $q \in Q$, let $\delta^*(q,\Lambda) = \Lambda(\{q\})$

2) For every $q \in Q$, $y \in \Sigma^*$, and $\sigma \in \Sigma$, let $\delta^*(q,y\sigma) = \Lambda(\bigcup\limits_{r \in \delta^*(q,y)} \delta(r,\sigma))$

$\Lambda$-**closure** of a set $S \subseteq Q$, denoted $\Lambda(S)$, is the set of all states reachable from $S$ by a sequence of $\Lambda$-transitions.

Define $\Lambda(S)$:

1. let $S \subseteq \Lambda(S)$
2. if $q \in \Lambda(S)$ then $\delta(q,\Lambda) \subseteq \Lambda(S)$
3. n.e. in $\Lambda(S)$

[Section 4.2]
Defining the computation of an NFA-Λ $M=(Q,\Sigma,q_0,A,\delta)$.

Extended transition function $\delta^*: Q \times \Sigma^* \rightarrow 2^Q$:

1) For every $q \in Q$, let $\delta^*(q,\Lambda) \neq \{q\}$
2) For every $q \in Q$, $y \in \Sigma^*$, and $\sigma \in \Sigma$, let $\delta^*(q,y\sigma) = \bigcup_{p \in \delta^*(q,y)} \Lambda(\delta(p,\sigma))$

We say that a string $x \in \Sigma^*$ is **accepted by $M$** if $\delta^*(q_0,x) \cap A \neq \emptyset$

A string which is not accepted by $M$ is **rejected by $M$**.

The **language accepted by $M$**, denoted by $L(M)$ is the set of all strings accepted by $M$. 

[Section 4.2]
Is the following statement true? **Yes.**

For every NFA $M = (Q, \Sigma, q_0, A, \delta)$ there exists an NFA-$\Lambda$ $M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$ such that $L(M) = L(M_1)$. 

**NFA-$\Lambda$'s vs. NFA's**
Is the following statement true?

For every NFA-Λ $M = (Q, \Sigma, q_0, A, \delta)$ there exists an NFA $M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$ such that $L(M) = L(M_1)$.

$$Q_1 = Q$$

$$q_1 = q_0$$

$$A_1 = \begin{cases} A_1 \cup \{q_0\} & \text{if } \Delta(q_0) \cap A \neq \emptyset \\ A_1 & \text{otherwise} \end{cases}$$

$$\delta_1(q, \sigma) = \delta^*(q, \sigma) \quad \forall q \in Q, \forall \sigma \in \Sigma$$

(Note: This diagram is not yet finished! (Finish it))
Kleene's Thm: A language $L$ over $\Sigma$ is regular iff there exists a finite automaton that accepts $L$.

Part 1: For every regular language, there exists an NFA-\(\Lambda\) accepting it.

1. $\emptyset$, \{\(\Lambda\)\}, \{\(\varepsilon\)\} for $\forall \varepsilon \in \Sigma$ are regular.
2. If $L_1$, $L_2$ are regular, so are:
   ① $L_1 \cup L_2$
   ② $L_1 \cdot L_2$
   ③ $L_1^*$
Proof of Kleene’s Thm

Part 2: Any language accepted by an FA is regular.

Let $M=\langle \{1,2,3,\ldots,k\}, \Sigma, q_0, A, \delta \rangle$ be an FA. We will show that for every $p,q,r \in Q$, the following language is regular:

\[ L(p,q,r) = \{ x \in \Sigma^* \mid \delta^*(p,x)=q \text{ and for every prefix } y \text{ of } x \text{ other than } \Lambda \text{ and } x \text{ we have } \delta^*(p,y) \leq r \} \]

Can we then conclude that $L(M)$ is regular? 

\[ \text{YES, because:} \]

\[ \text{if } |A|=1 \text{ then } L(M) = L(q_0, s, k) \quad A = \{s\} \]

\[ \text{if } |A| > 1 \text{ then } L(M) = \bigcup_{s \in A} L(q_0, s, k) \]

\[ \text{finite union of reg. lang.} \]

\[ \Rightarrow L(M) \text{ is regular} \]
**Proof of Kleene's Thm**

**Claim**: Let $M = ({1, 2, 3, \ldots, k}, \Sigma, q_0, A, \delta)$ be an FA. For every $p, q, r \in Q$, the following language is regular:

$$L(p, q, r) = \{ x \in \Sigma^* | \delta^*(p, x) = q \text{ and for every prefix } y \text{ of } x \text{ other than } \Lambda \text{ and } x \text{ we have } \delta^*(p, y) \leq r \}$$

**Pf**: by induction on $r$:

**BASE CASE**: $r = 0, \forall p, q \in Q$: $L(p, q, 0) = \{ \varepsilon \in \Sigma^* | \delta(p, \varepsilon) = q \}$ finite, thus regular.

**IH**: $\forall p, q \in Q, L(p, q, r)$ is regular.

**IND. CASE**: $\forall p, q \in Q, L(p, q, r+1)$ is regular. ∈ WANT TO PROVE

$$L(p, q, r+1) = L(p, q, r) \cup \left( L(p, r+1, r) \cup L(r+1, r+1, r) \right)^* \cup L(r+1, q, r)$$

- **not going through** $r+1$
- **going through** $r+1$