Lecture 6: DFA = NFA = regular expressions

Let $L \subseteq \Sigma^*$. Then the following three statements are equivalent.

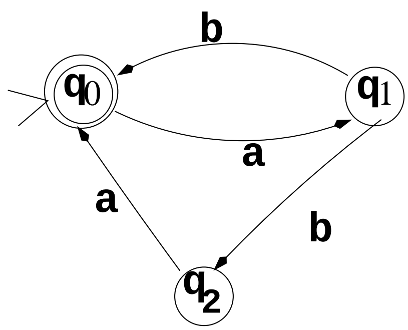
1. *L is accepted by some DFA.*
2. *L is accepted by some NFA.*
3. *L can be represented by a regular expression*

Proof: We will prove the following: i) $1 \Leftrightarrow 2$, ii) $2 \Leftrightarrow 3$.

<u>i) $1 \Leftrightarrow 2$</u>

$1 \Rightarrow 2$: A DFA is an NFA, by definition.

To prove that $2 \Rightarrow 1$: This is a constructive proof.
First, let's work on a simple NFA that does not contain e-edges.



It is possible to go from one state to several states after consuming one input symbol. Hence, the states now are subsets of K:

δ	a	b
$\{q_0\}$	$\{q_1\}$	$\emptyset$
$\{q_1\}$	$\emptyset$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\emptyset$
$\{q_0, q_1\}$	$\{q_1\}$	$\{q_0, q_2\}$
$\emptyset$	$\emptyset$	$\emptyset$

How do we decide the final states?

What if the given NFA has some e edges?

Idea: On reading $a \in \Sigma$, the DFA M' imitates a move of the NFA M on a, possibly follows any number of e-moves.

```
s --> q1 --> q2 --> q3 --> q4 --> q5 -> q6 -> q7 -> q8
  e       e      a      e      e     b     e     e
```

Initial state is $\{s, q1, q2\}$.

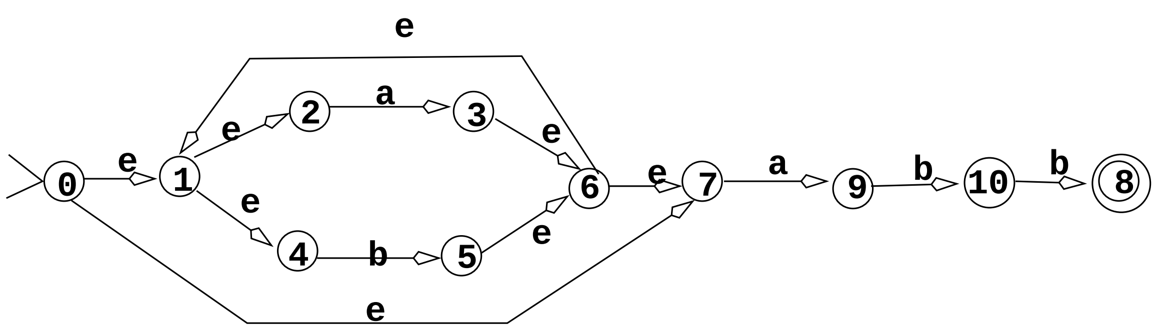
$\delta(\{s, q1, q2\}, a) = \{q3, q4, q5\}$

$\delta(\{q3, q4, q5\}, b) = \{q6, q7, q8\}$

Define

$$E(q) = \text{the set of states } \textit{reachable} \text{ from } q, \text{ without reading any input symbols}$$
$$= \{p \in K \mid (q, e) \vdash_M^* (p, e)\}$$

Example:



$E(0) = \{0, 1, 2, 4, 7\}$ $E(1) = \{1, 2, 4\}$
$E(2) = \{2\}$ $E(3) = \{3, 6, 1, 2, 4, 7\}$
$E(4) = \{4\}$ $E(5) = \{5, 6, 7, 1, 2, 4\}$
$E(6) = \{6, 7, 1, 2, 4\}$ $E(7) = \{7\}$
$E(8) = \{8\}$ $E(9) = \{9\}$
$E(10) = \{10\}$

The fundamental theorem

Given an NFA $M = \{K, \Sigma, \Delta, s, F\}$.

We want to construct an equivalent DFA that accepts the same language:

$M' = \{K', \Sigma, \delta', s', F'\}$

- $K' = 2^K$
- $s' = E(s)$
- $F' = \{Q \in K' \mid Q \cap F \neq \emptyset\}$
- for all $Q \in K'$, $\sigma \in \Sigma$,
  $\delta'(Q, \sigma) = \cup_{q \in Q}\{E(p) : (q, \sigma, p) \in \Delta\}$.

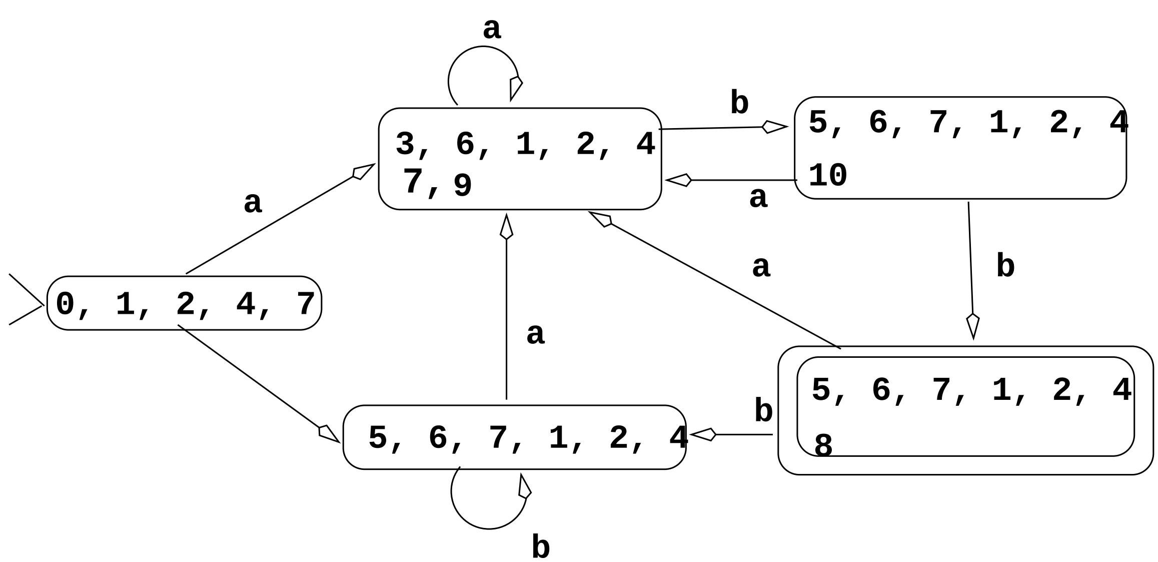
  Note: $\delta'(\emptyset, \sigma) = \emptyset$.

  Note: $\delta'(Q, \sigma)$ may equals $\emptyset$ for some $Q \in K'$, $\sigma \in \Sigma$.

  Note: Some Q's might be unreachable but that's OK.

Refer to the given NFA:

	a	b
$E(0) = \{0, 1, 2, 4, 7\}$	$E(3) \cup E(9) = \{3, 6, 1, 2, 4, 7, 9\}$	$E(5) = \{5, 6, 7, 1, 2, 4\}$
$\{3, 6, 1, 2, 4, 7, 9\}$	$E(3) \cup E(9) = \{3, 6, 1, 2, 4, 7, 9\}$	$E(5) \cup E(10) = \{5, 6, 7, 1, 2, 4, 10\}$
$\{5, 6, 7, 1, 2, 4\}$	$E(9) \cup E(3) = \{3, 6, 1, 2, 4, 7, 9\}$	$E(5) = \{5, 6, 7, 1, 2, 4\}$
$\{5, 6, 7, 1, 2, 4, 10\}$	$E(9) \cup E(3) = \{3, 6, 1, 2, 4, 7, 9\}$	$E(5) \cup E(10) = \{5, 6, 7, 1, 2, 4, 8\}$
$\{5, 6, 7, 1, 2, 4, 8\}$	$E(9) \cup E(3) = \{3, 6, 1, 2, 4, 7, 9\}$	$E(5) = \{5, 6, 7, 1, 2, 4\}$



To be rigorous, we should also show that (i) M' is deterministic and (ii) $L(M) = L(M')$. We only sketch the proof here (refer to textbook for details).

- From the construction process, we notice that δ' is single-valued and well defined on all $Q \in K'$ and $\sigma \in \Sigma$, thus M' is deterministic.

- To show $L(M) = L(M')$, we should show that $\forall w \in \Sigma^*$,

  $$(s, w) \vdash_M^* (f, e) \text{ for some } f \in F$$

  $$<=> (s', w) \vdash_{M'}^* (Q, e) \text{ for some } Q \in F'.$$

  where $s' = E(s)$.

  To prove this statement, it's sufficient to prove the following more general claim.

Claim: $\forall p, q \in K$, $w \in \Sigma^*$,

$$(q, w) \vdash_M^* (p, e)$$

$$<=> (E(q), w) \vdash_{M'}^* (P, e) \text{ for some } P \text{ containing } p.$$

Proof: by induction on $|w|$.

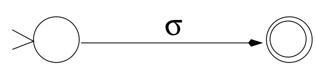
The fundamental theorem

ii) To prove $3 \Rightarrow 2$:
Given a regular expression γ.
Show that there exists an NFA that accepts $L(\gamma)$.

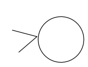
Prove by induction on the number of operations (concatenation, union, Kleene star) in γ.

Basis step: γ has zero operation.

- $\gamma = \sigma$ for some $\sigma \in \Sigma$. The following NFA accepts $L\{\sigma\}$



- $\gamma = \emptyset$. The following NFA accepts $L\{\emptyset\}$



Induction hypothesis: If γ contains k or fewer operations, then there exists an NFA that accepts $L(\gamma)$.

Induction step: Consider a regular expression γ that contains $k+1$ operations. There are three possible cases:

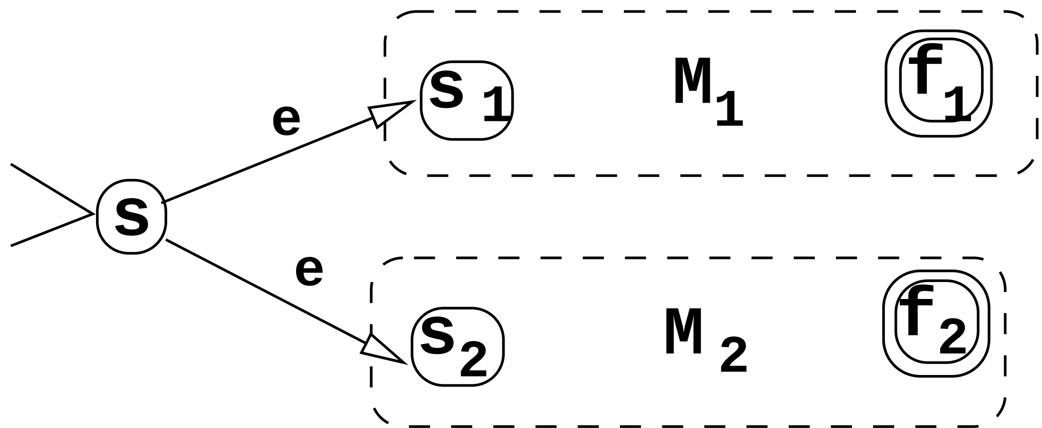
1. $\gamma = \alpha \cup \beta$:

   - By the induction hypothesis, since α and β contain fewer than k operations:

     (a) an NFA M_1 that accepts $L(\alpha)$, and

     (b) an NFA M_2 that accepts $L(\beta)$.

   - We can construct an NFA that accepts $L(\gamma)$ from M_1 and M_2.



Formally, let

$M_1 = \{K_1, \Sigma, \Delta_1, s_1, F_1\}$.

$M_2 = \{K_2, \Sigma, \Delta_2, s_2, F_2\}$ where $K_1 \cap K_2 = \emptyset$.

Then, we construct $M = \{K, \Sigma, \Delta, s, F\}$ where

- $s \notin K_1 \cup K_2$
- $K = K_1 \cup K_2 \cup \{s\}$
- $F = F_1 \cup F_2$
- $\Delta = \Delta_1 \cup \Delta_2 \cup \{(s, e, s_1), (s, e, s_2)\}$

Then, for all $w \in \Sigma^*$,

$$(s, w) \vdash_M^* (q, e) \text{ for some } q \in F$$

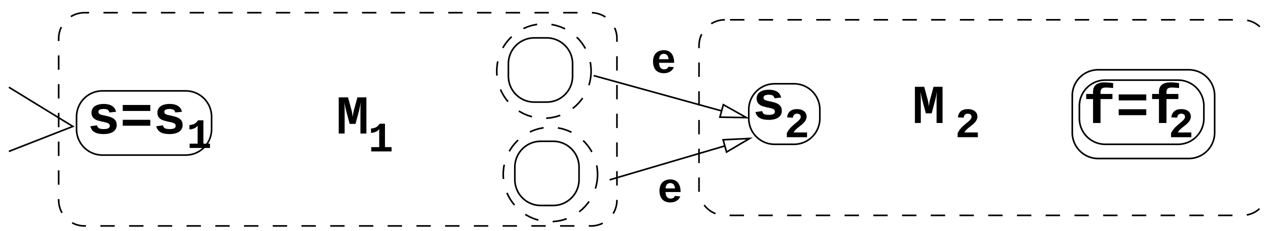
$$<=> \text{ either } (s_1, w) \vdash_{M_1}^* (q, e) \text{ for some } q \in F_1$$

$$\text{or } (s_2, w) \vdash_{M_2}^* (q, e) \text{ for some } q \in F_2$$

Hence, $L(M) = L(M_1) \cup L(M_2)$.
Hence $L(M) = L(\alpha) \cup L(\beta) = L(\alpha \cup \beta) = L(\gamma)$.

2. $\gamma = \alpha\beta$:

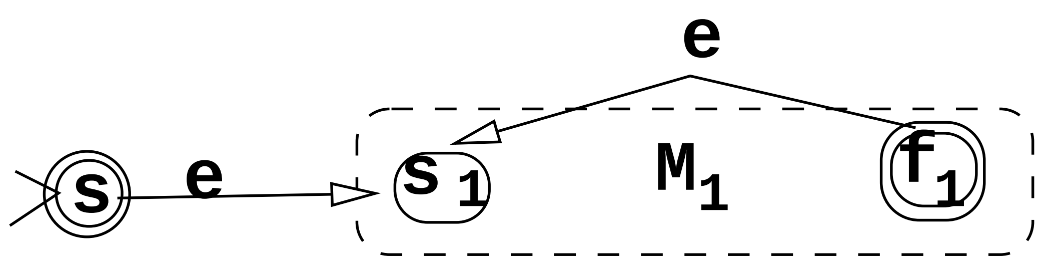
   - By the induction hypothesis, there exist
     (a) an NFA M_1 that accepts $L(\alpha)$, and
     (b) an NFA M_2 that accepts $L(\beta)$.
   - The following NFA accepts $L(\gamma)$.



How do you write M formally as a quintuple?

3. $\gamma = \alpha^*$:

   - By the induction hypothesis, there exists an NFA M_1 that accepts $L(\alpha)$.
   - The following NFA accepts $L(\gamma)$.



How do you write M formally as a quintuple?

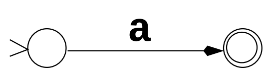
Why do we need to introduce a new start state? Why can't we simply make the original start state a final state?

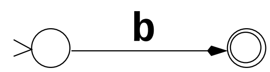
Note that (2-3) serve as proofs for the following lemma: if L_1 and L_2 are accepted by FAs, then L_1^* and $L_1 L_2$ are accepted by FAs.

Example:

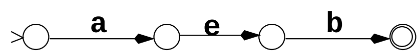
Construct an NFA that accepts $(ab \cup aab)*$

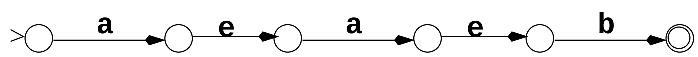
Stage 1

a: 

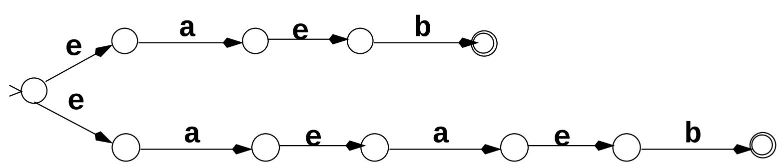
b: 

Stage 2:

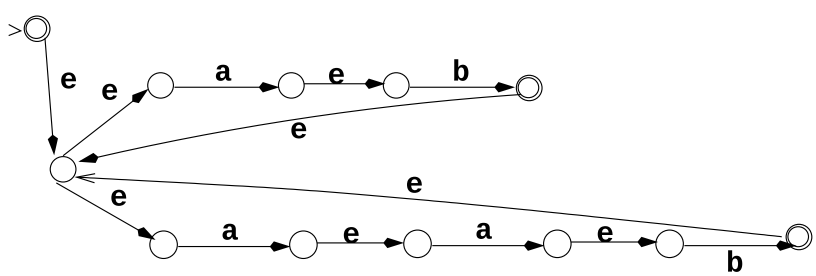
ab: 

aab: 

Stage 3: ab U aab:



Stage4 : (ab U aab) *



ii) $2 \Rightarrow 3$:

Given any NFA, there is an equivalent DFA $M = (K, \Sigma, s, \delta, F)$. Show that there exists a regular expression that generates $L(M)$. We sketch the proof here:

- Assign a number to each state: $K = \{q_1, \ldots, q_n\}$, with the start state $s = q_1$.
- For $i, j = 1, ..., n$, $k = 0, ..., n$, define

  $R(i, j, k)$

  $= \{$ all strings in Σ^* that drive M from q_i to q_j without passing through any intermediate state numbered $> k$ $\}$

  $= \{ \sigma_1\sigma_2 \ldots \sigma_{l-1}$:
  $q_i = q_{r_1} \rightarrow^{\sigma_1} \rightarrow q_{r_2} \ldots \rightarrow q_{r_{l-1}} \rightarrow^{\sigma_{l-1}} \rightarrow q_{r_l} = q_j$
  such that $r_2, \ldots, r_{l-1} \leq k\}$.

  Note:

  - The endpoints q_i and q_j are allowed to be numbered higher than k.
  - $R(i, j, 0)$ is the set of strings that start from state q_i, ends at state q_j, without passing through any intermediate state.

- $R(i, j, k-1) \subseteq R(i, j, k)$ for all i, j, k.

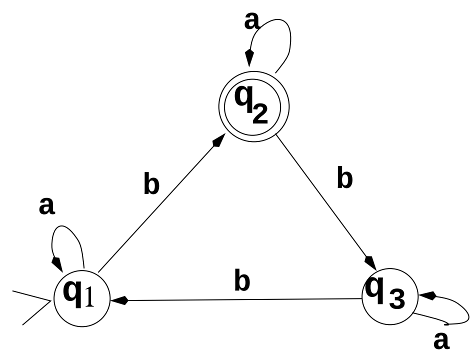
- Note that $R(i, j, n)$ (where n is the number of states) contains all strings that are allowed to pass through any states (since no states are numbered $> n$).

- Hence, $L(M) = \quad \cup\{R(1, j, n) : q_j \in F\}$

- If each $R(1, j, n)$ can be represented by a regular expression, then $L(M)$ can be represented by a regular expression.

- We shall prove that, for all i, j, k, $R(i, j, k)$ can be represented by a regular expression.

Example:



$R(1,1,0) = e \cup a$
$R(1,2,0) = b$
$R(1,3,0) = \emptyset$
$R(2,1,0) = \emptyset$
$R(2,2,0) = e \cup a$
$R(2,3,0) = b$
$R(3,1,0) = b$
$R(3,2,0) = \emptyset$
$R(3,3,0) = e \cup a$

We can contnue to work out the answers for larger k by inspecting the FA diagram, but it gets more tedious as k gets larger.
$R(1,1,1) = a^*$
$R(1,2,1) = a^*b$
$R(1,1,2) = a^*$
$R(1,2,2) = a^*ba^*$
$R(1,3,2) = a^*ba^*b$

Claim:

$$R(i, j, k) = R(i, j, k-1) \cup R(i, k, k-1) R(k, k, k-1)^* R(k, j, k-1)$$

Proof:

- Let $\sigma_1\sigma_2 \ldots \sigma_l \in R(i, j, k)$.
- By definition of $R(i, j, k)$, $\sigma_1\sigma_2 \ldots \sigma_l$ drives M from q_i to q_j without passing through any intermediate state $> k$.
- Case (i): it does not pass through the state numbered k. Then $\sigma_1\sigma_2 \ldots \sigma_l \in R(i, j, k-1)$.
- Case (ii): it passes through the state numbered k.
  - the string can be divided into 3 parts, where the first part drives M from q_i to q_k without passing through states $> k-1$, the second part drives M from q_k to q_k, zero or more times, without passing through states $> k-1$, and the third part drives M from q_k to q_j without passing through states $> k-1$.

  Therefore,
  $\sigma_1\sigma_2 \ldots \sigma_l \in R(i, k, k-1) R(k, k, k-1)^* R(k, j, k-1)$.
- From case (i) and (ii), we have

  $\sigma_1\sigma_2 \ldots \sigma_l \in R(i, j, k-1) \cup R(i, k, k-1) R(k, k, k-1)^* R(k, j, k-1)$

Claim: $R(i,j,k)$ can be represented by a regular expression, for $i,j = 1,...,n, k = 0,1,...,n$.

Proof: By induction on k.

1. *Basis step*: When $k = 0$,

   $$R(i,j,0) = \begin{cases} \{a \in \Sigma \cup \{e\} : (q_i, a, q_j) \in \Delta\} & \text{if } i \neq j \\ \{e\} \cup \{a \in \Sigma \cup \{e\} : (q_i, a, q_j) \in \Delta\} & \text{if } i = j. \end{cases}$$

   Since each set is finite, each $R(i,j,0)$ can be represented by a regular expression.

2. *Induction hypothesis*: Suppose $R(i,j,k-1)$ can be represented by a regular expression for all i,j.

3. *Inductive step*: Consider

   $R(i,j,k) =$

   $$R(i,j,k-1) \cup R(i,k,k-1)R(k,k,k-1)^* R(k,j,k-1)$$

   By the induction hypothesis,
   each of the $R(i,j,k-1)$, $R(i,k,k-1)$, $R(k,k,k-1)$, and $R(k,j,k-1)$ can be represented by a regular expression.

   Since the above expression uses only union, concatenation and Kleene star, $R(i,j,k)$ can be represented by a regular expression.

The constructive proof provides a top-down algorithm to find an equivalent regular expression given an NFA.

The language accepted by the NFA example is $R(1,2,3)$, and

$$R(1,2,3) = R(1,2,2) \cup R(1,3,2)R(3,3,2)^*R(3,2,2)$$

So, we only need the following:

$$R(1,2,2) = R(1,2,1) \cup R(1,2,1)R(2,2,1)^*R(2,2,1)$$
$$R(1,3,2) = R(1,3,1) \cup R(1,2,1)R(2,2,1)^*R(2,3,1)$$
$$R(3,2,2) = R(3,2,1) \cup R(3,2,1)R(2,2,1)^*R(2,2,1)$$
$$R(3,3,2) = R(3,3,1) \cup R(3,2,1)R(2,2,1)^*R(2,3,1)$$

Hence, we need the following:

$$\begin{aligned}
R(1,2,1) &= R(1,2,0) \cup R(1,1,0)R(1,1,0)^*R(1,2,0) \\
&= b \cup (a \cup e)(a \cup e)^* b = a^* b \\
R(1,3,1) &= R(1,3,0) \cup R(1,1,0)R(1,1,0)^*R(1,3,0) \\
&= \emptyset \cup (a \cup e)(a \cup e)^* \emptyset = \emptyset \\
R(2,2,1) &= \cdots = a \cup e \\
R(2,3,1) &= \cdots = b \\
R(3,2,1) &= \cdots = ba^*b \\
R(3,3,1) &= \cdots = a \cup e
\end{aligned}$$

Substituting these results into $R(i,j,2)$:

$R(1,2,2) = a^*ba^*$
$R(1,3,2) = a^*ba^*b$
$R(3,2,2) = ba^*ba^*$
$R(3,3,2) = a \cup e \cup ba^*ba^*b$

Finally, substitute these results into $R(1,2,3)$:
$R(1,2,3) = a^*ba^* \cup a^*ba^*b(a \cup e \cup ba^*ba^*b)^*ba^*ba^*$
which, with some effort, can be simplified to $a^*ba^*(a^*ba^*ba^*ba^*)^*$.