NFA EQUIVALENCE

COMP 4200 – Formal Language

TRY YOURSELF!

closure

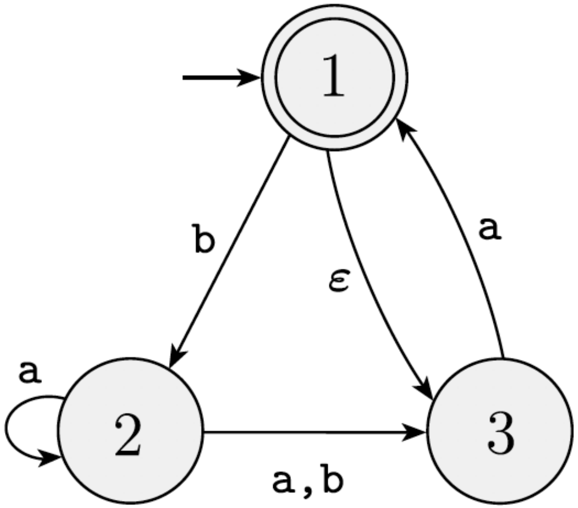
1: {1, 3}

2: {2}

3: {3}

	ε^*	a	ε^*
1	1 3	∅ 1	∅ {1, 3}
2	2	2 3	2 3 {2, 3}
3	3	1	{1, 3}

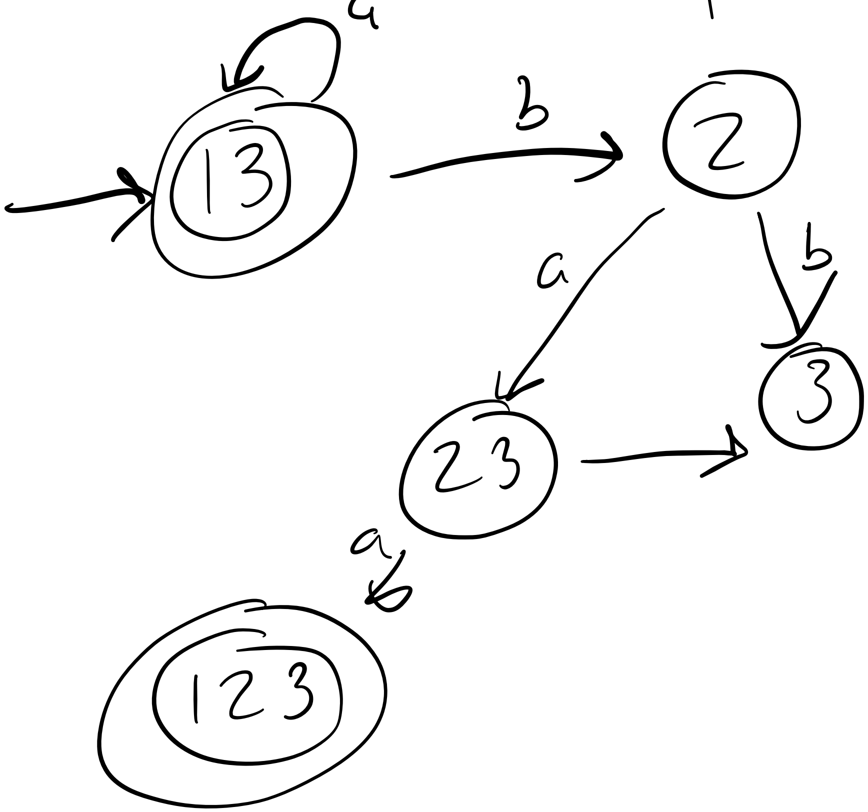
	ε^*	b	ε^*
1	1 3	2 ∅	2
2	2	3	3
3	3	∅	∅



NFA

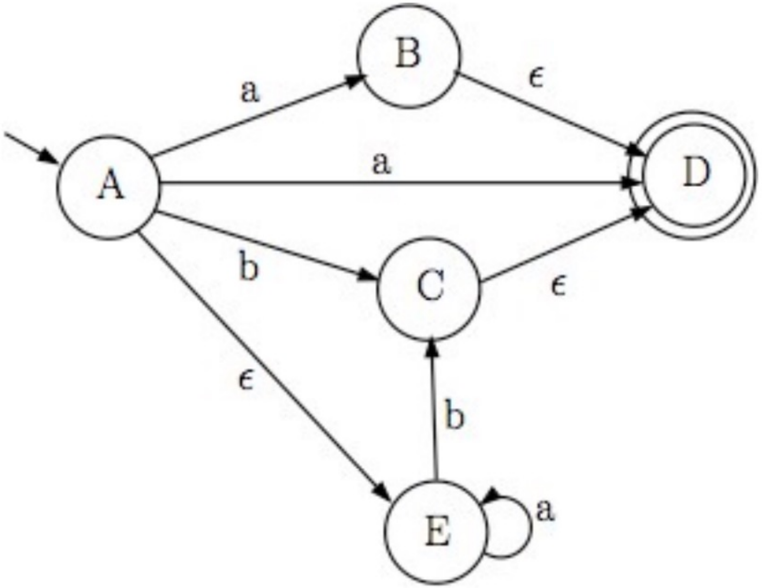
	a	b
1	{1,3}	2
2	{2,3}	3
3	{1,3}	∅

	a	b
~~1~~	~~13~~	~~2~~
*→ 13	13	2
2	23	3
23	123	3
* 123	123	23
3	13	∅
∅	∅	∅



	ϵ^*	a	ϵ^*
A	A E	B D E	{D, B} D E
B	B D	$\emptyset$ $\emptyset$	$\emptyset$
C	C D	$\emptyset$ $\emptyset$	$\emptyset$
D	D	$\emptyset$	$\emptyset$
E	E	E	E

EXAMPLE



closure

A: {A, E}

B: {B, D}

C: {C, D}

D: {D}

E: {E}

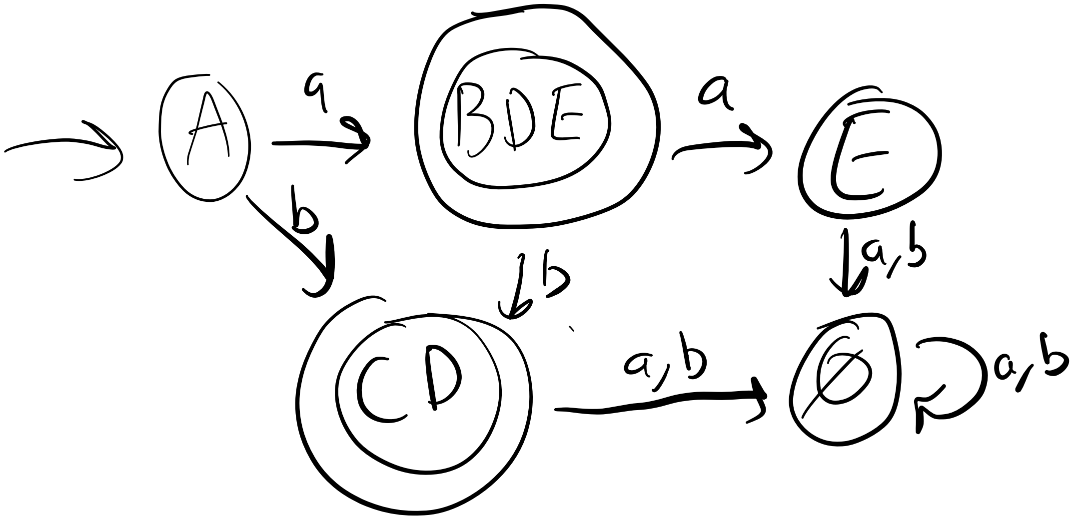
	ϵ^*	b	ϵ^*
A	A, E	C, C	{C, D}
B	B, D	∅, ∅	∅
C	C, D	∅, ∅	∅
D	D	∅	∅
E	E	C	C, D

NFA

	a	b
→ A	{B, D, E}	{C, D}
B	∅	∅
C	∅	∅
* D	∅	∅
E	E	{C, D}

DFA

	a	b
→ A	BDE	CD
* BDE	E	CD
* CD	∅	∅
∅	∅	∅
E	∅	∅

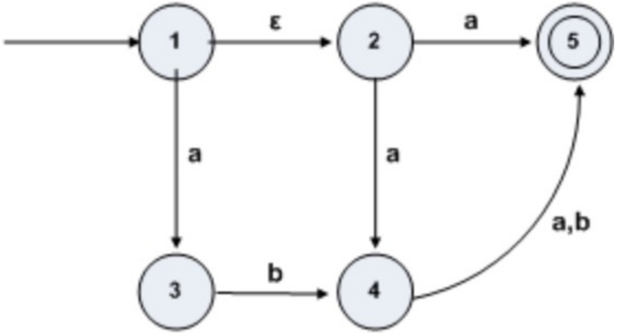


A
a
b
BDE
a
E
a,b
b
CD
a,b
∅
a,b

EXAMPLE

	ϵ^*	a	ϵ^*
1	1, 2	3, {4,5}	3, {4,5}
2	2	4, 5	4, 5
3	3	$\emptyset$	$\emptyset$
4	4	5	5
5	5	$\emptyset$	$\emptyset$

	ϵ^*	b	ϵ^*
1	1, 2	$\emptyset$, $\emptyset$	$\emptyset$
2	2	$\emptyset$	$\emptyset$
3	3	4	4
4	4	5	5
5	5	$\emptyset$	$\emptyset$



closure

1 : {1, 2}

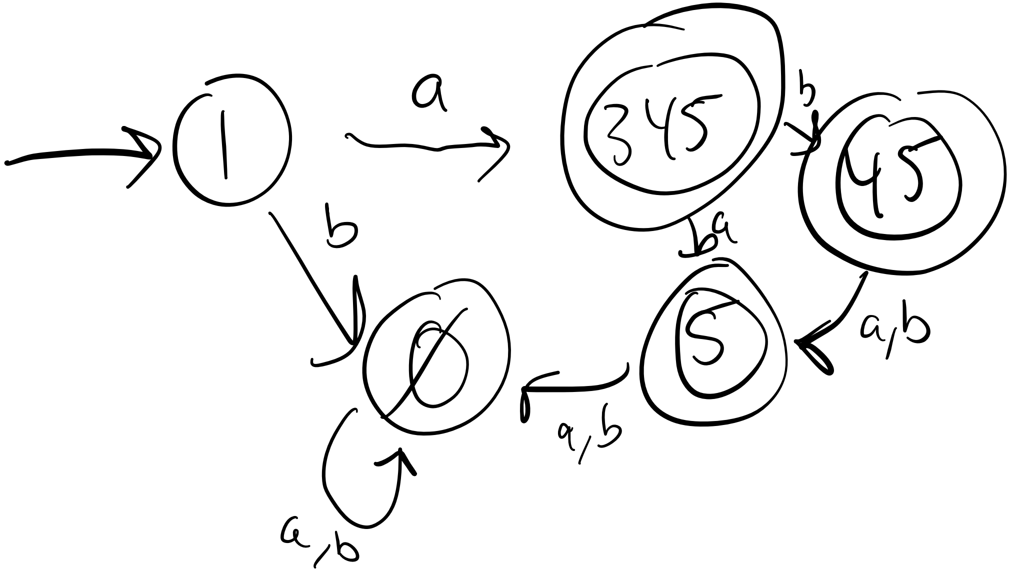
2 : {2}

3 : {3}

4 : {4}

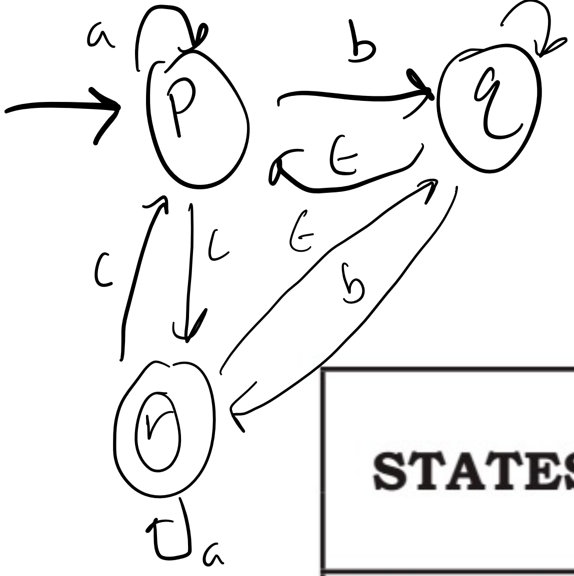
5 : {5}

NFA	a	b
→ 1	{3,4,5}	$\emptyset$
2	{4,5}	$\emptyset$
3	$\emptyset$	4
4	5	5
* 5	$\emptyset$	$\emptyset$



DFA	a	b
→ 1	345	∅
* 345	5	45
* 45	5	5
∅	∅	∅
~~2~~	~~45~~	~~∅~~
~~3~~	~~∅~~	~~4~~
~~4~~	~~5~~	~~5~~
* 5	∅	∅

a



EXAMPLE

STATES	INPUTS			
	ε	a	b	c
=>p	Φ	{p}	{q}	{r}
q	{p}	{q}	{r}	Φ
*r	{q}	{r}	Φ	{p}

closure

p : {p}
q : {q, p}
r : {r, q, p}

	ε*	a	ε*
p	p	p	p
q	q p	q p	{q,p} p
r	r q p	r q p	{r,q,p} {q,p} p

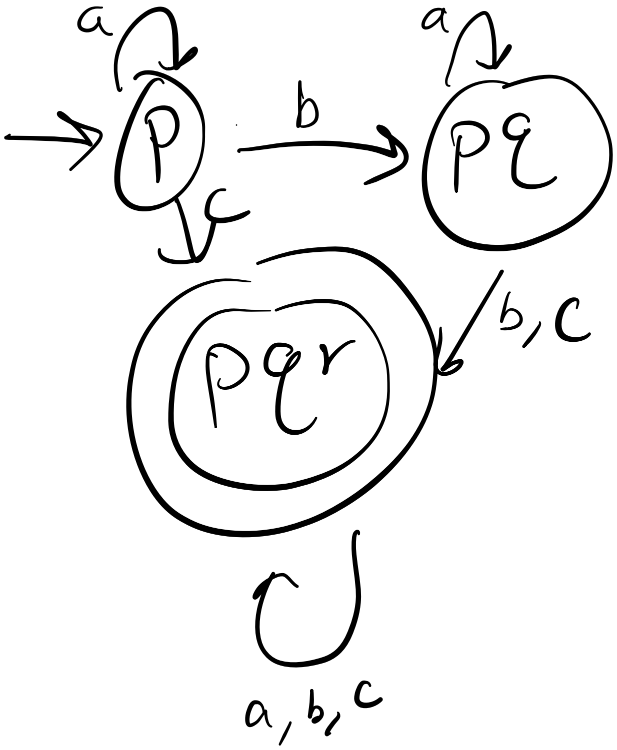
	ϵ^*	b	ϵ^*
p	p	q	$\{q,p\}$
q	q p	r q	$\{r,q,p\}$ $\{q,p\}$
r	r q p	$\emptyset$ r q	$\emptyset$ $\{r,q,p\}$ $\{q,p\}$

	ϵ^*	c	ϵ^*
p	p	r	$\{r,q,p\}$
q	q p	$\emptyset$ r	$\emptyset$ $\{r,q,p\}$
r	r q p	p $\emptyset$ r	p $\emptyset$ $\{r,q,p\}$

NFA

	a	b	c
$\rightarrow p$	p	$\{p,q\}$	$\{p,q,r\}$
q	$\{p,q\}$	$\{p,q,r\}$	$\{p,q,r\}$
$* r$	$\{p,q,r\}$	$\{p,q\}$	$\{p,q,r\}$

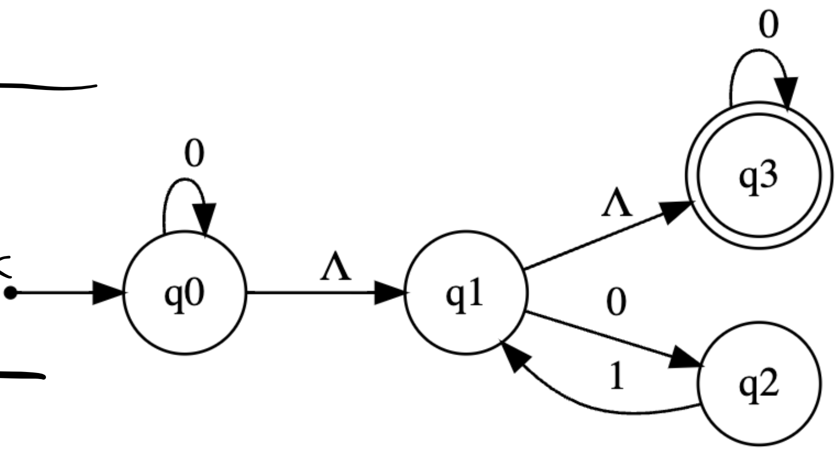
DFA	a	b	c
→ p	p	pq	pqr
pq	pq	pqr	pqr
★ pqr	pqr	pqr	pqr
~~q~~	~~pq~~	~~pqr~~	~~pqr~~
~~r~~	~~pqr~~	~~pqr~~	~~pqr~~



EXAMPLE

	Λ^*	0	Λ^*
q_0	q_0 q_1 q_3	q_0 q_2 q_3	$\{q_0, q_1, q_3\}$ q_2 q_3
q_1	q_1 q_3	q_2 q_3	q_2 q_3
q_2	q_2	$\emptyset$	$\emptyset$
q_3	q_3	q_3	q_3

	Λ^*	1	Λ^*
q_0	q_0 q_1 q_3	$\emptyset$ $\emptyset$ $\emptyset$	$\emptyset$
q_1	q_1 q_3	$\emptyset$ $\emptyset$	$\emptyset$
q_2	q_2	q_1	$\{q_1, q_3\}$
q_3	q_3	$\emptyset$	$\emptyset$



closure

$q_0 : \{q_0, q_1, q_3\}$

$q_1 : \{q_1, q_3\}$

$q_2 : \{q_2\}$

$q_3 : \{q_3\}$

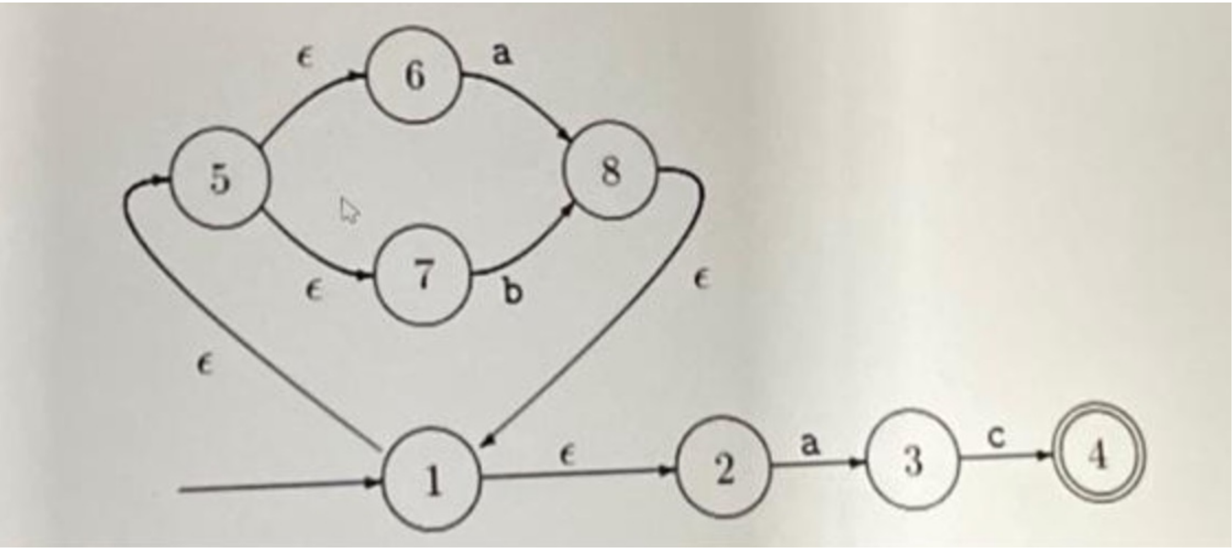
NFA

	0	1
$\rightarrow q_0$	$\{q_0, q_1, q_2, q_3\}$	$\emptyset$
q_1	$\{q_2, q_3\}$	$\emptyset$
q_2	$\emptyset$	$\{q_1, q_3\}$
$* q_3$	q_3	$\emptyset$

DFA

	0	1
$\rightarrow q_0$	q_{0123}	$\emptyset$
$* q_{0123}$	q_{0123}	q_{13}
$* q_{13}$	q_{23}	$\emptyset$
$* q_{23}$	q_3	q_{13}
$\emptyset$	$\emptyset$	$\emptyset$
~~q_2~~	~~$\emptyset$~~	~~q_{13}~~
$* q_3$	q_3	$\emptyset$

EXAMPLE



closure

1: {1, 2, 5, 6, 7}

2: {2}

3: {3}

4: {4}

5: {5, 6, 7}

6: {6}

7: {7}

8: {8, 1, 2, 5, 6, 7}

REGULAR OPERATION FOR NFA

CLOSURE UNDER REGULAR OPERATION

t $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize A_1, and
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognize A_2.

nstruct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize $A_1 \cup A_2$.

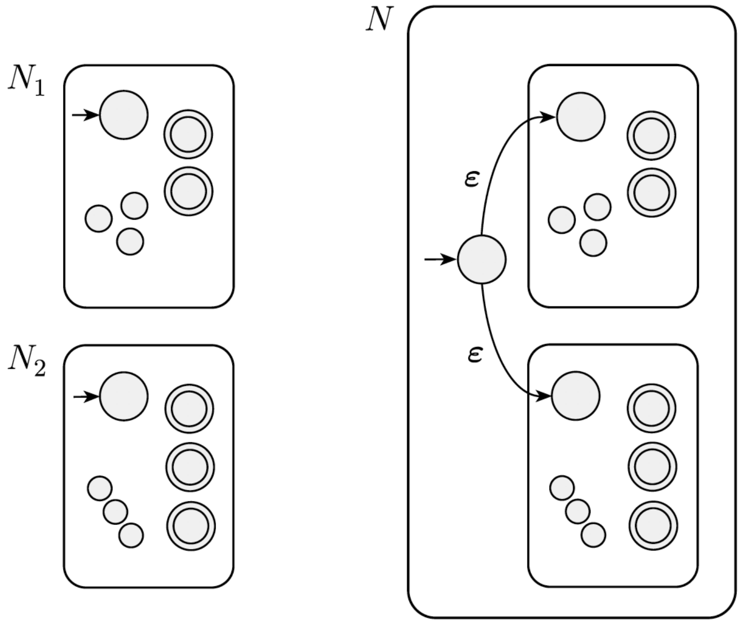
1. $Q = \{q_0\} \cup Q_1 \cup Q_2$.
   The states of N are all the states of N_1 and N_2, with the addition of a start state q_0.
2. The state q_0 is the start state of N.
3. The set of accept states $F = F_1 \cup F_2$.
   The accept states of N are all the accept states of N_1 and N_2. That wa accepts if either N_1 accepts or N_2 accepts.
4. Define δ so that for any $q \in Q$ and any $a \in \Sigma_\varepsilon$,

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \\ \delta_2(q,a) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0 \text{ and } a = \varepsilon \\ \emptyset & q = q_0 \text{ and } a \neq \varepsilon. \end{cases}$$

- The class of regular languages is closed under the union operation.
- Let A_1 and A_2 be regular languages. We want to show A_1 U A_2 is a regular language.
- Since A_1 and A_2 are regular languages there exists an NFA N_1 and there exists an NFA N_2 such that N_1 recognizes A_1 and N_2 recognizes A_2.

CONSTRUCTION OF NFA N TO RECOGNIZE $A_1 \cup A_2$

new start states, no touching final states



REGULAR LANGUAGES ARE CLOSED UNDER CONCATENATION

Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize A_1, and
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognize A_2.

Construct $N = (Q, \Sigma, \delta, q_1, F_2)$ to recognize $A_1 \circ A_2$.

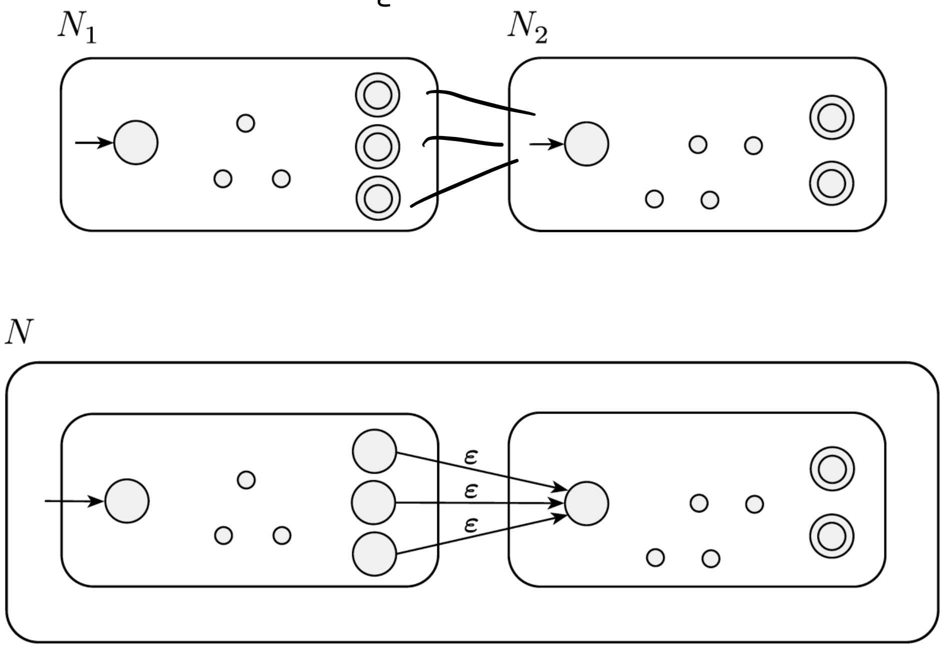
1. $Q = Q_1 \cup Q_2$.
   The states of N are all the states of N_1 and N_2.
2. The state q_1 is the same as the start state of N_1.
3. The accept states F_2 are the same as the accept states of N_2.
4. Define δ so that for any $q \in Q$ and any $a \in \Sigma_\varepsilon$,

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \text{ and } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\} & q \in F_1 \text{ and } a = \varepsilon \\ \delta_2(q,a) & q \in Q_2. \end{cases}$$

CONSTRUCTION OF N TO RECOGNIZE A1 ∘ A2

Hello World

connect



REGULAR LANGUAGES ARE CLOSED UNDER THE STAR OPERATION

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. $Q = \{q_0\} \cup Q_1$.
   The states of N are the states of N_1 plus a new start state.
2. The state q_0 is the new start state.
3. $F = \{q_0\} \cup F_1$.
   The accept states are the old accept states plus the new start state.

4. Define δ so that for any $q \in Q$ and any $a \in \Sigma_\varepsilon$,

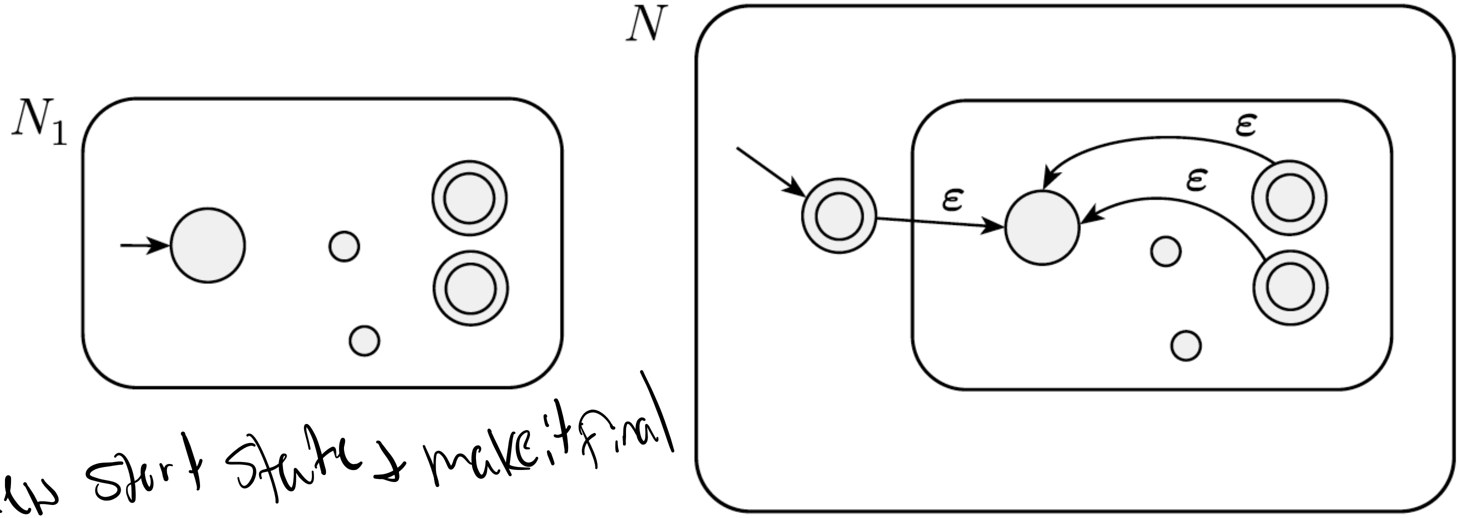
$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \text{ and } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1 \text{ and } a = \varepsilon \\ \{q_1\} & q = q_0 \text{ and } a = \varepsilon \\ \emptyset & q = q_0 \text{ and } a \neq \varepsilon. \end{cases}$$

clean star

CONSTRUCTION OF N TO RECOGNIZE A*

Keep looping

repitition



new start state & make it final

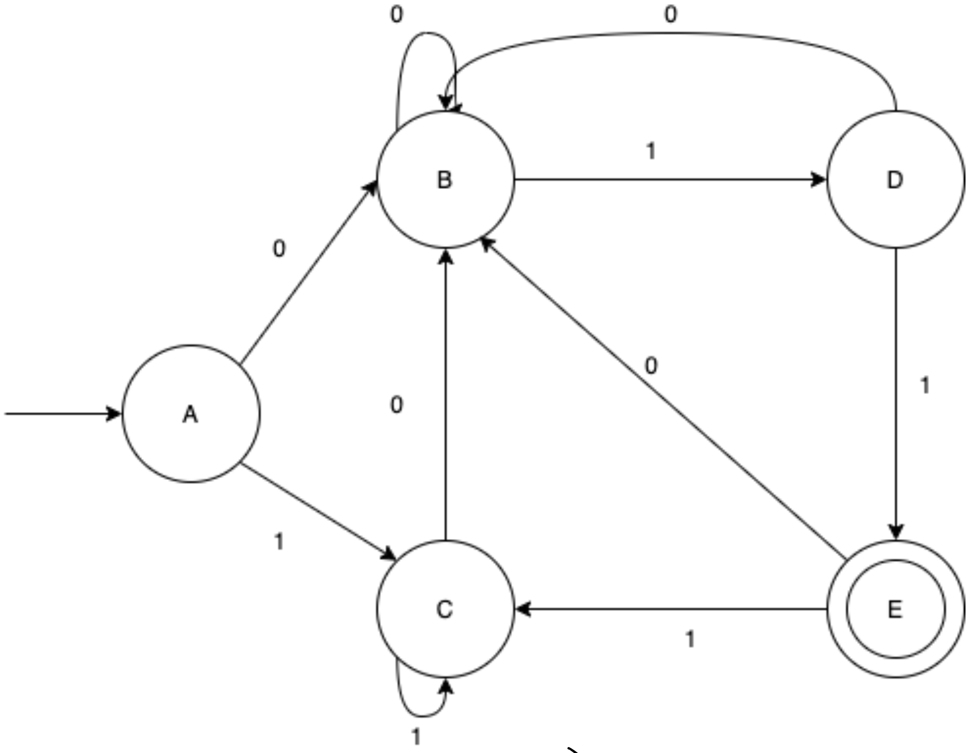
ε

MINIMIZATION OF DFA

- Requirement → a minimal version of any DFA, which consist of minimum number of states possible.
- Lesser the number of states, higher the efficiency.
- But how?
  - Combine two states and make it as one state
  - How?
    - Condition: Two states can be combined only if two states are equivalent.
- Equivalent:
  - Two states are said to be equivalent if
    - $\delta(A,X) \rightarrow F \; and \; \delta(B,X) \rightarrow F \; or \; \delta(A,X) \nrightarrow F \; and \delta(B,X) \nrightarrow F$
    - X is the input string.
    - F is the final state.

- Equivalence types:
  - If |X| = 0, then A and B are said to be 0 equivalent. → length of the input string is 0
  - If |X| = 1, then A and B are said to be 1 equivalent.
  - If |X| = 2, then A and B are said to be 2 equivalent.
  - …..
  - If |X| = n, then A and B are said to be n equivalent.

$\delta(D,0)=B$ $\delta(D,1)=E$



Transition table

		0	1
→	A	B	C
	B	B	D
	C	B	C
	D	B	E
*	E	B	C

0 equv. $\underbrace{\{A,B,C,D\}}_{\text{non-final}}$ $\underbrace{\{E\}}_{\text{Final}}$

1 equv: $\{A,B,C\}$ $\{D\}$ $\{E\}$

2 eqv: $\{A,C\}$ $\{B\}$ $\{D\}$ $\{E\}$

3 equv: $\{A,C\}$ $\{B\}$ $\{D\}$ $\{E\}$

can stop here

$\delta(A,0)=B$ $\delta(A,1)=C$

$\delta(B,0)=B$ $\delta(B,1)=D$

$\delta(C,0)=B$ $\delta(C,1)=C$

Equivalence → Non final states together and rest as a separate set.

- '0' Equivalence → {A,B,C,D} {E}
- '1' Equivalence → {A,B,C} {E} {D}

☑ (A,B): $\delta(A,0) = B, \delta(A,1) = C, \delta(B,0) = B, \delta(B,1) = D$ → They fall in the same set

☑ (A,C): $\delta(A,0) = B, \delta(A,1) = C, \delta(C,0) = B, \delta(C,1) = C$ → They fall in the same set

✗ (A,D): $\delta(A,0) = B, \delta(A,1) = C, \delta(D,0) = B, \delta(D,1) = E$ → They fall in different set

- '2' Equivalence → {A,C} {B} {D} {E}

(A,B): $\delta(A,0) = B, \delta(A,1) = C, \delta(B,0) = B, \delta(B,1) = D$ → They fall in different set

(A,C): $\delta(A,0) = B, \delta(A,1) = C, \delta(C,0) = B, \delta(C,1) = C$ → They fall in the same set

	0	1
A	B	C
B	B	D
C	B	C
D	B	E
E	B	C

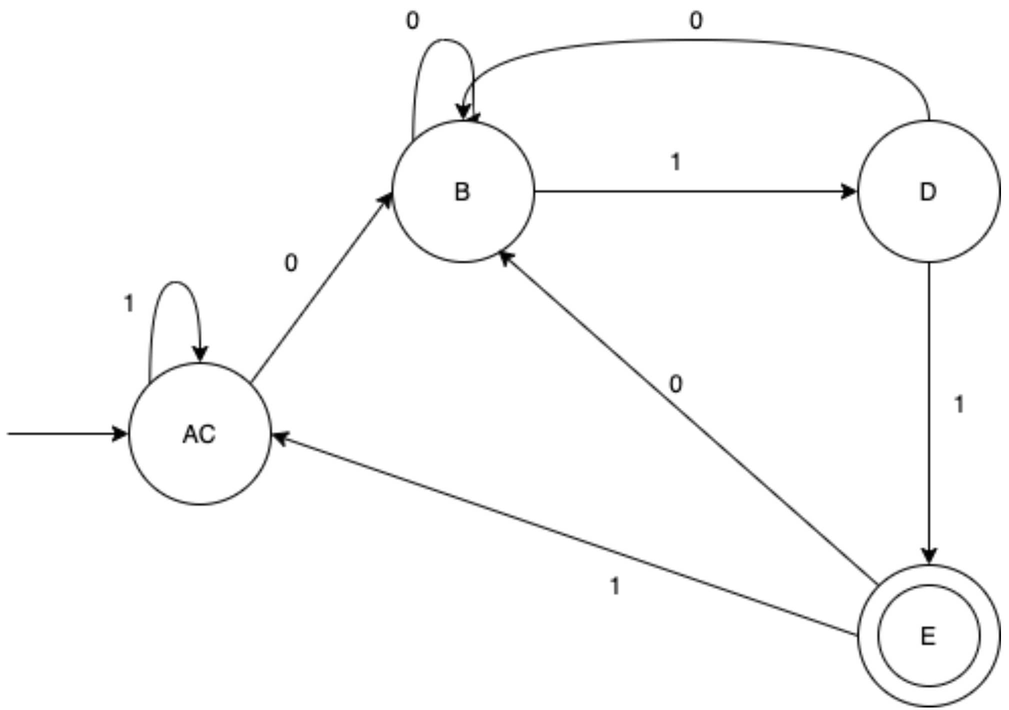
- '3' Equivalence → {A,C} {B} {D} {E}

- Minimal version → {A,C} {B} {D} {E}

Transition table

	0	1
A	B	C
B	B	D
C	B	C
D	B	E
E	B	C

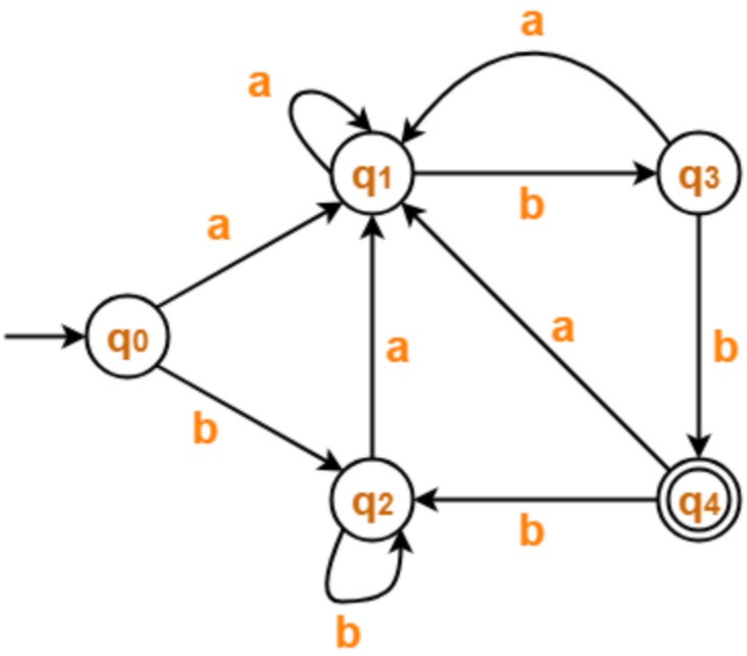
- Minimal version → {A,C} {B} {D} {E}



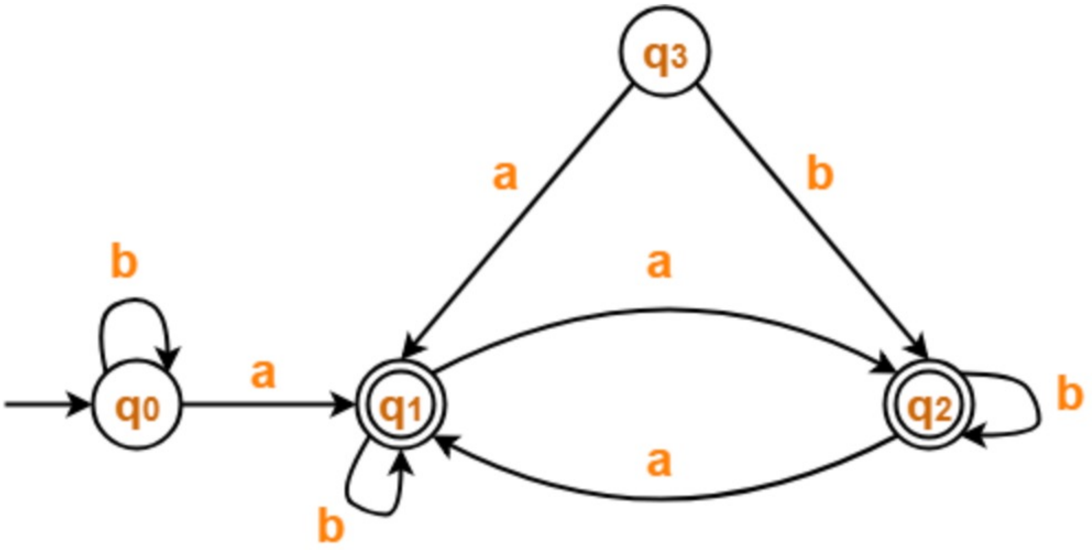
Transition table

	0	1
A	B	C
B	B	D
C	B	C
D	B	E
E	B	C

EXAMPLE



EXAMPLE



EXAMPLE

	0	1
→ A	B	F
B	G	C
C	A	C
D	C	G
E	H	F
F	C	G
G	G	E
H	G	C

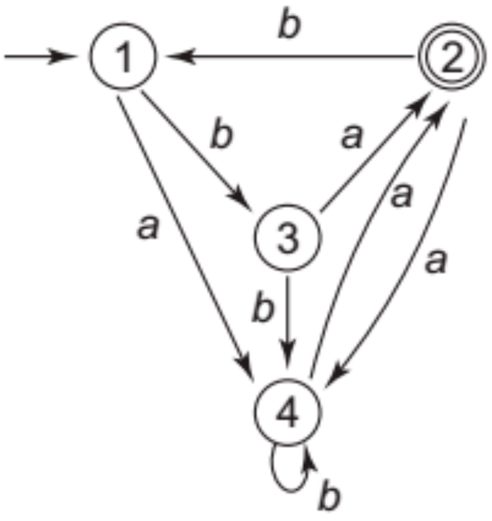
EXAMPLE

State	a	b
$\rightarrow q_0$	q_1	q_2
q_1	q_4	q_3
q_2	q_4	q_3
q_3	q_5	q_6
q_4	q_7	q_6
q_5	q_3	q_6
q_6	q_6	q_6
q_7	q_4	q_6

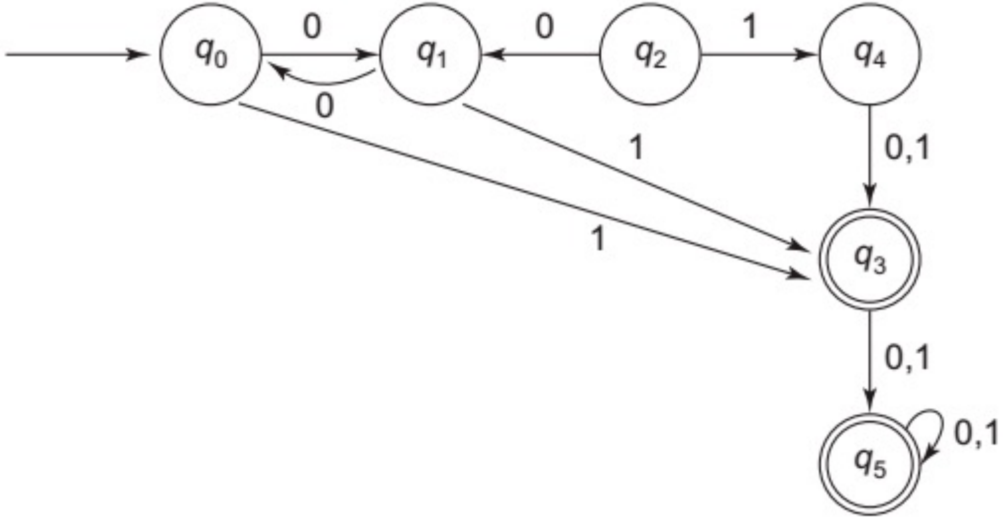
EXAMPLE

Input symbols and states	Next state	
	a	b
→*1	3	2
2	4	1
3	5	4
4	4	4
5	3	2

EXAMPLES TO TRY!



(1)



(2)