# COMPUTATION THEORY

From the Lecture Notes 15-453 FORMAL LANGUAGES, AUTOMATA AND COMPUTABILITY
By Emiratus prof. Edmund M. Clarke
Carnegie Mellon

# The PUMPING LEMMA and REGULAR EXPRESSIONS

# SOME LANGUAGES **ARE NOT** REGULAR

**B = {$0^n1^n$ | n ≥ 0} is NOT regular!**

# **WHICH OF THESE** ARE REGULAR

**C = { w | w has equal number of 1s and 0s}**

**NOT REGULAR**

**D = { w | w has equal number of occurrences of 01 and 10}**

**REGULAR!!!**

Because $\Rightarrow$ Next Slide

Each “01” marks a transition from 0 to 1; each “10” marks a transition from 1 to 0.
By concatenation, transitions alternate such as …01…10…01…10…, etc.

Therefore, the counts of 01 and 10 are either equal or differ by exactly 1, depending only on the endpoints:

If the first and last bits are the same (both 0 or both 1), then #01 = #10.
If they differ, then |#01 − #10| = 1

w ∈ L if and only if w is empty or its first and last symbols are the same.

# THE PUMPING LEMMA

Let L be a regular language with $|L| = \infty$

Then there exists a positive integer P such that

if $w \in L$ and $|w| \geq P$

then w = xyz, where:

1. $|y| > 0$
2. $|xy| \leq P$
3. $xy^iz \in L$ for any $i \geq 0$

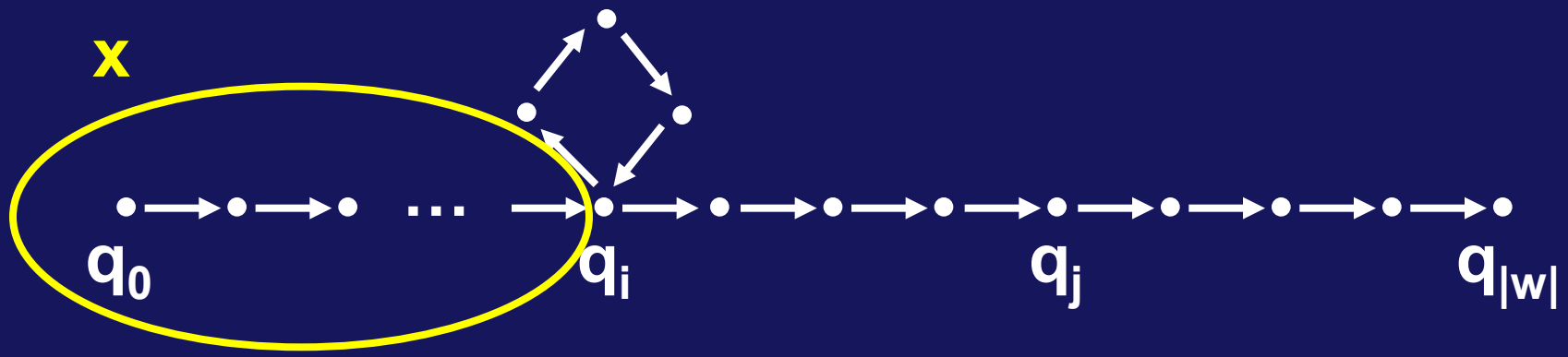
Let M be a DFA that recognizes L

Let P be the number of states in M

Assume $w \in L$ is such that $|w| \geq P$

We show w = xyz

1. $|y| > 0$
2. $|xy| \leq P$
3. $xy^iz \in L$ for any $i \geq 0$




There must be $j > i$ such that $q_i = q_j$

Use the pumping lemma to prove that
C = { w | w has an equal number of 0s and 1s}
is not regular

Hint: Try pumping $s = 0^P1^P$

If C is regular, s can be split into s = xyz,
where for any $i \geq 0$, $xy^iz$ is also in C
and $|xy| \leq P$

# USING THE **PUMPING LEMMA**

**Use the pumping lemma to prove that**
**B = $\{0^n1^n \mid n \geq 0\}$ is not regular**

**Hint: Assume B is regular**

**Let B = L(M), for DFA M,**
**and let P be larger than the**
**number of states in M**

**Try pumping s = $0^P1^P$**

# USING THE **PUMPING LEMMA**

**Use the pumping lemma to prove that**
**$B = \{0^n1^n \mid n \geq 0\}$ is not regular**

**Hint: Assume B is regular, and try pumping $s = 0^P1^P$**

**If B is regular, s can be split into s = xyz, where for any $i \geq 0$, $xy^iz$ is also in B**

**If y is all 0s:** **xyyz has more 0s than 1s**

**If y is all 1s:** **If y has both 1s and 0s:**

**If y has both 1s and 0s:**

**xyyz will have some 1s before some 0s**

# REGULAR **EXPRESSIONS**

**$\sigma$ is a regular expression representing $\{\sigma\}$**

**$\varepsilon$ is a regular expression representing $\{\varepsilon\}$**

**$\varnothing$ is a regular expression representing $\varnothing$**

**If $R_1$ and $R_2$ are regular expressions representing $L_1$ and $L_2$ then:**

**$(R_1R_2)$ represents $L_1 \cdot L_2$**

**$(R_1 \cup R_2)$ represents $L_1 \cup L_2$**

**$(R_1)^*$ represents $L_1^*$**

# PRECEDENCE

| | |
|---|---|
| Tightest | Star (“*”) |
| | Concatenation (“.”, “”) |
| Loosest | Union (“∪”, “+”, “\|”) |

# EXAMPLE

$$R_1 * R_2 \cup R_3 = ((R_1^*)R_2) \cup R_3$$

{ w | w has exactly a single 1 }

**0*10***

**What language does $\emptyset^*$ represent?**

$\{\varepsilon\}$

{ w | w has length ≥ 3 and its 3rd symbol is 0 }

$$000(0 \cup 1)^* \cup 010(0 \cup 1)^* \cup 100(0 \cup 1)^* \cup 110(0 \cup 1)^*$$

$$= (0 \cup 1)(0 \cup 1)0(0 \cup 1)^*$$

{ w | every odd position of w is a 1 }

$$1((0 \cup 1)1)^*(0 \cup 1 \cup \varepsilon) \cup \varepsilon$$

*Also*

$$(1(0 \cup 1))^*(1 \cup \varepsilon)$$

# EQUIVALENCE

**L can be represented by a regexp**

**⇔**

**L is a regular language**

**L can be represented by a regexp**

$\Rightarrow$

**L is a regular language**

**Given regular expression R, we show there exists NFA N such that R represents L(N)**

**Induction on the length of R:**

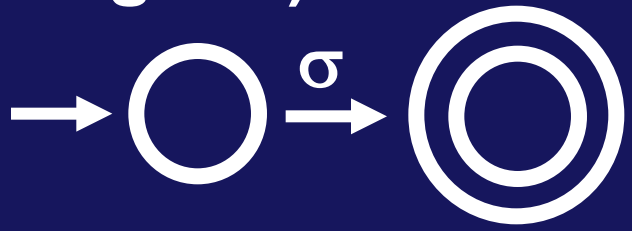
# Given regular expression R, we show there exists NFA N such that R represents L(N)

## Induction on the length of R:

### Base Cases (R has length 1):

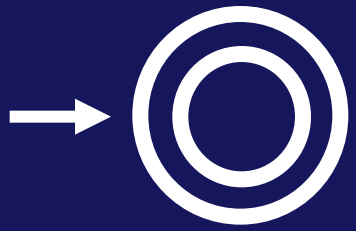
$R = \sigma$

(matches a single symbol)



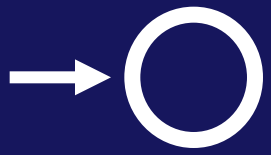

$R = \varepsilon$

(matches the empty string)



$R = \emptyset$

(matches nothing)



## Inductive Step:

Assume R has length k > 1 and that any regular expression of length < k represents a language that can be recognized by an NFA

Three possibilities for R:

$R = R_1 \cup R_2$ **(Union Theorem!)**

$R = R_1 R_2$

$R = (R_1)^*$

## Have Shown

**L can be represented by a regexp**

$\Rightarrow$

**L is a regular language**

**L can be represented by a regexp**

**L is a regular language**

# Cartesian Product of two DAFs

## Product of DFAs

We can run two DFAs in parallel on the same input via the **product construction**, as long as they share the same alphabet.

Suppose $DFA_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ and $DFA_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$

We define $DFA_1 \times DFA_2$ as follows:

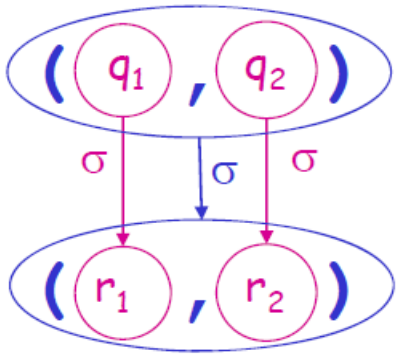
*States:* $Q_{1\times2} = Q_1 \times Q_2$

*Alphabet:* $\Sigma$

*Transitions:*

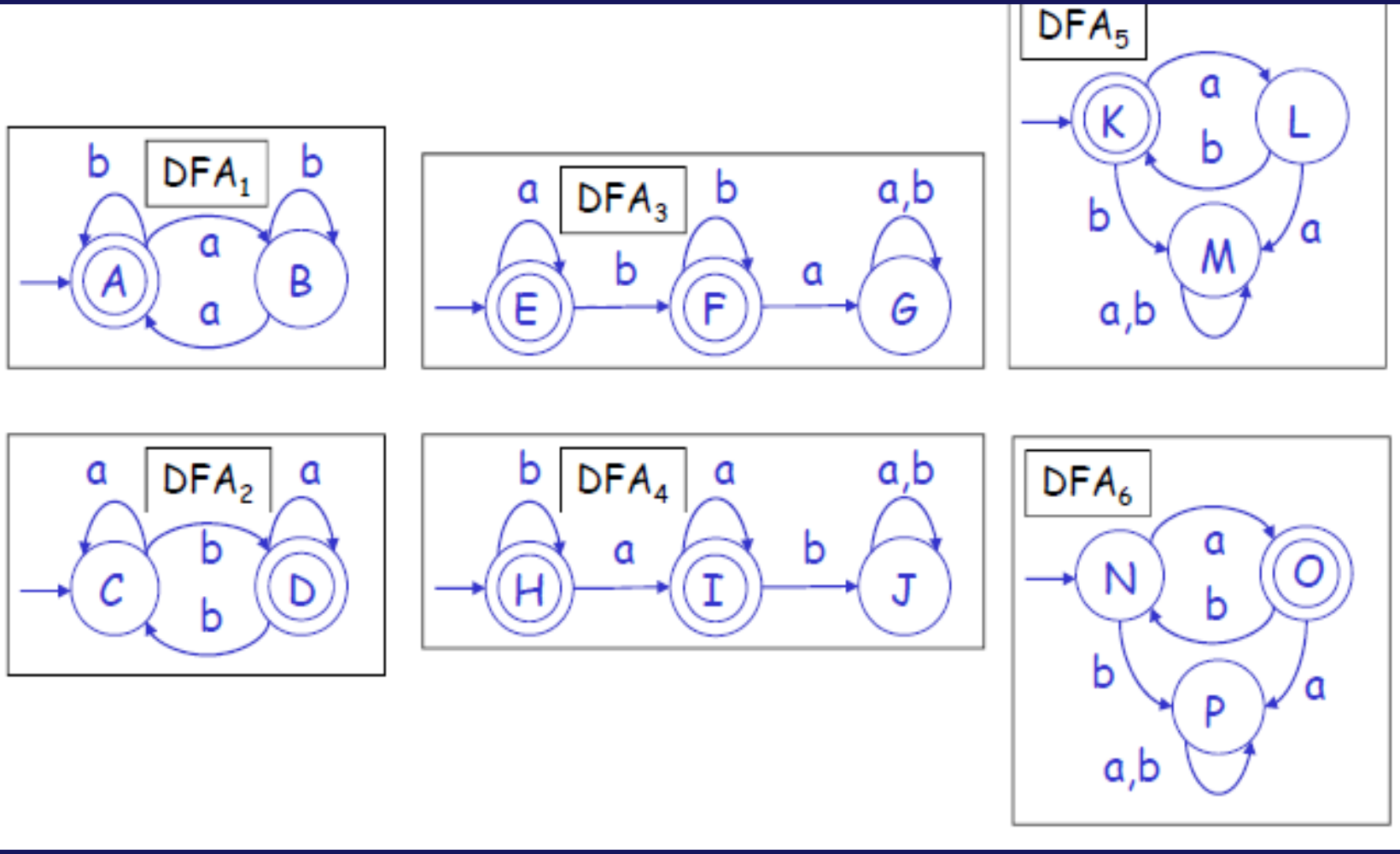
$\delta_{1\times2} \in Q_{1\times2} \times \Sigma \rightarrow Q_{1\times2}$

$\delta_{1\times2}\ (\ ((q_1,q_2), \sigma)\ )$
$= (\ \delta_1(\ (q_1,\sigma)\ ), \delta_2(\ (q_2,\sigma)\ )$



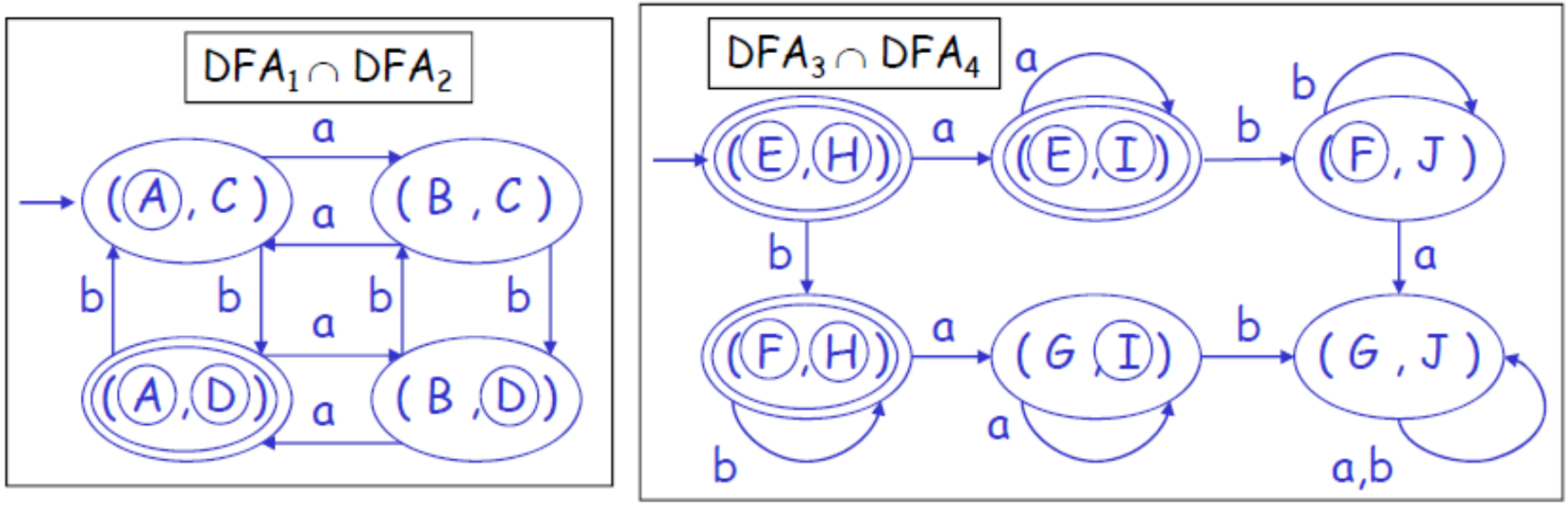

*Start State:* $s_{1\times2} = (s_1, s_2)$

*Final States: Definition depends on how we use product*

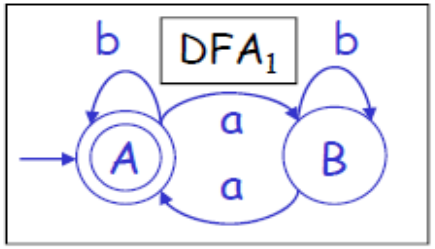

DFA1
b
b
a
a
A
B
DFA2
a
a
b
b
C
D
DFA3
a
b
a,b
a
b
E
F
G
DFA4
b
a
a,b
a
b
H
I
J
DFA5
a
b
K
L
b
a
M
a,b
DFA6
a
b
N
O
b
a
P
a,b

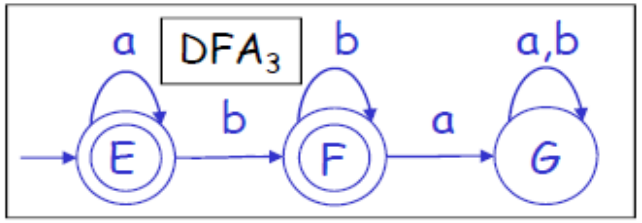
# Intersection of DFAs

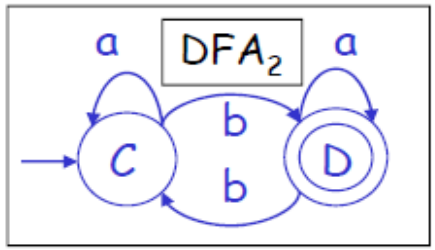
We can intersect $DFA_1$ and $DFA_2$ (written $DFA_1 \cap DFA_2$) by defining the accepting states of $DFA_1$ x $DFA_2$ as those state pairs in which **both** states are final states of their DFAs.

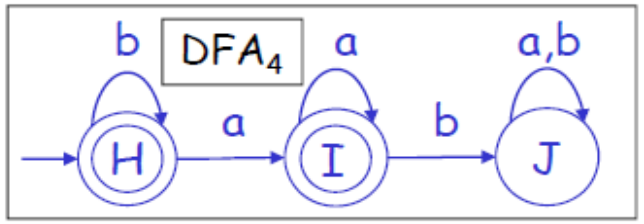


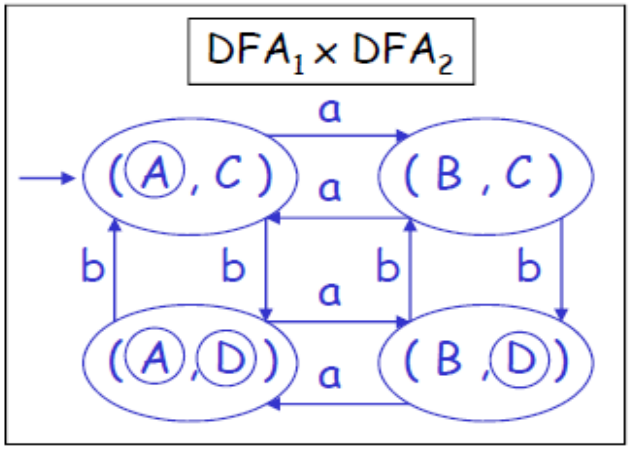
# Sample Products

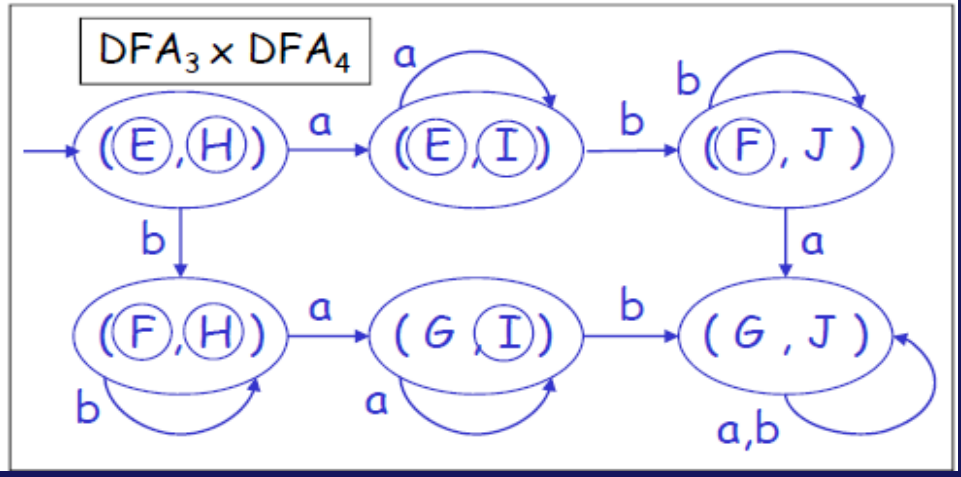


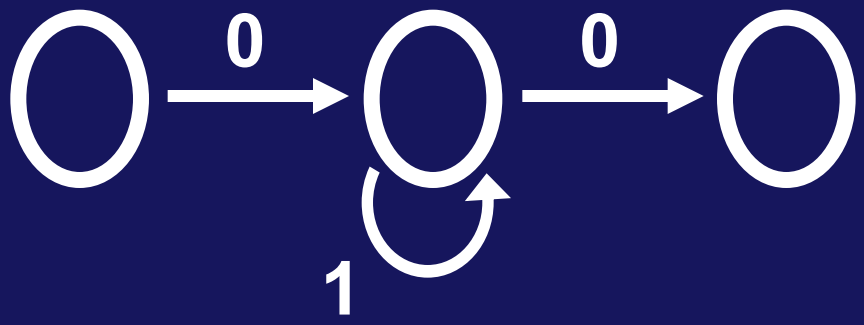



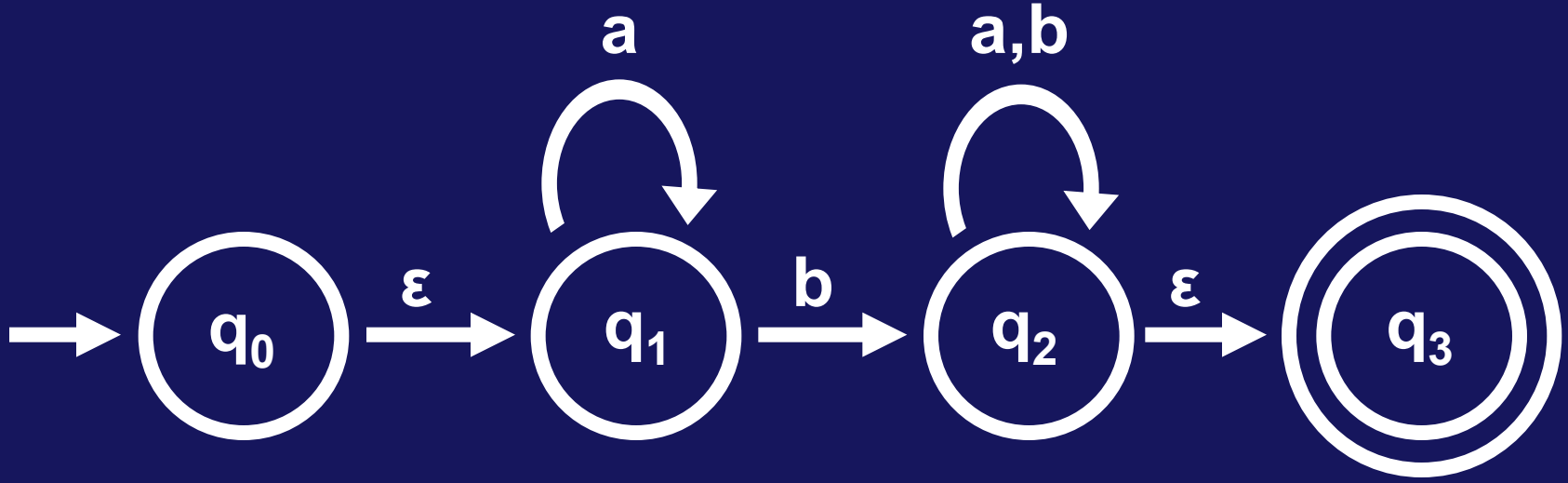

**Add** unique and distinct start and accept states
While machine has more than 2 states:

Pick an internal state, **rip it out and re-label the arrows** with regexps, to account for the missing state

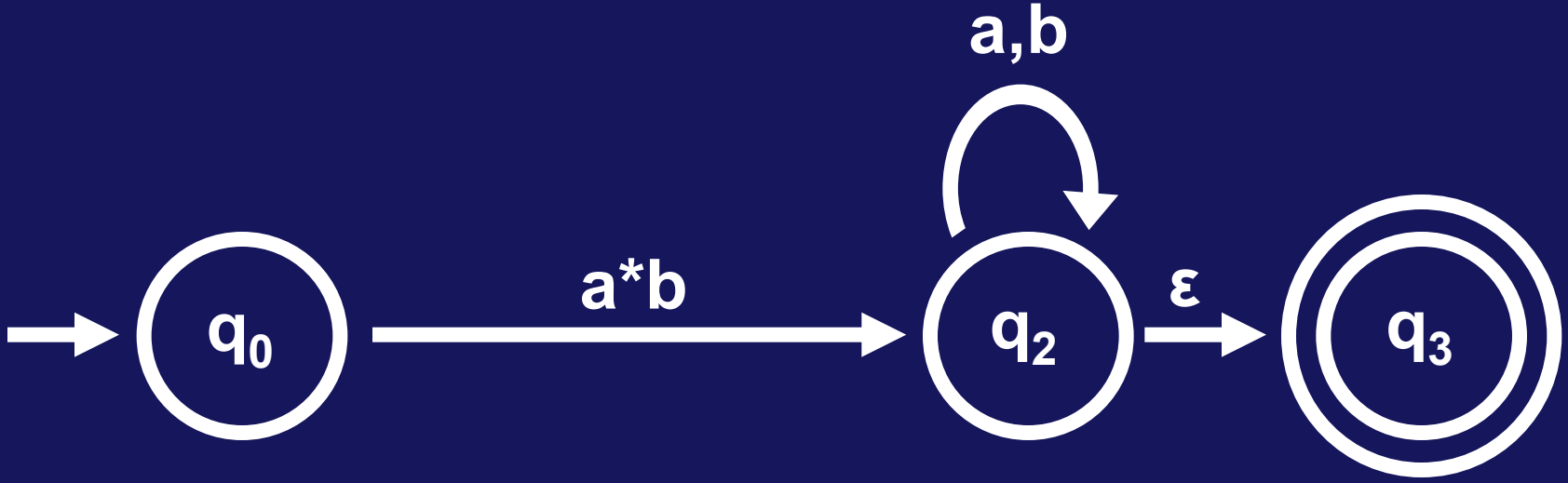



**While machine has more than 2 states:**

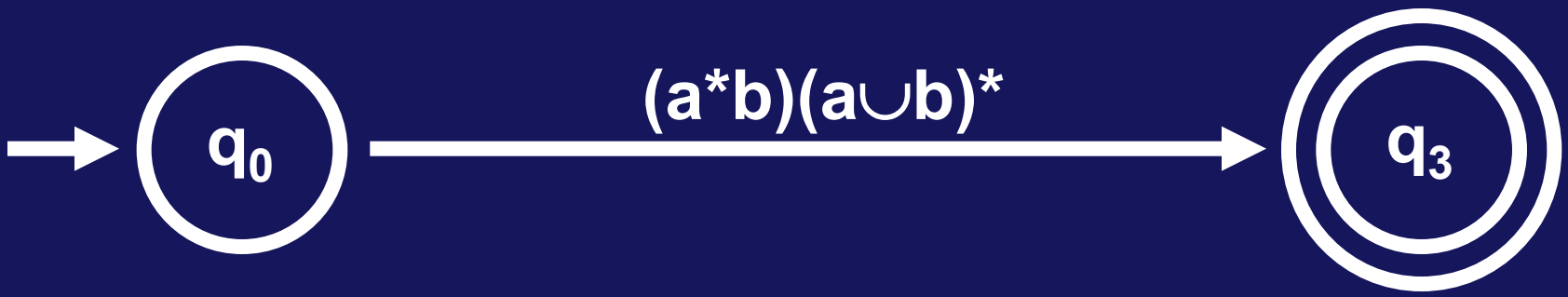
**Pick an internal state, rip it out and re-label the arrows with regexps, to account for the missing state**




$R(q_0,q_3) = (a^*b)(a \cup b)^*$




$$R(q_0,q_3) = (a^*b)(a \cup b)^*$$




$R(q_0,q_3) = (a^*b)(a \cup b)^*$