A Single Final State for Finite Accepters

Observation

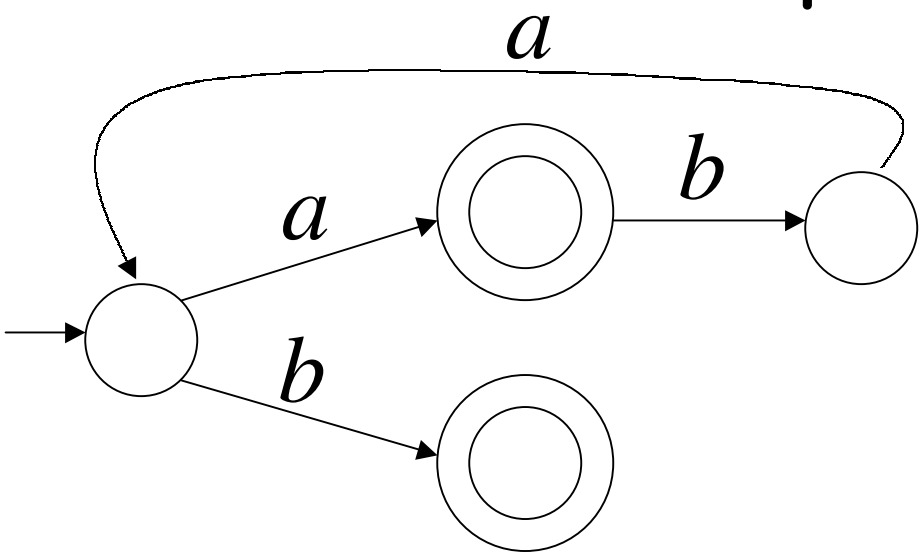
Any Finite Accepter (NFA or DFA)

can be converted to an equivalent NFA

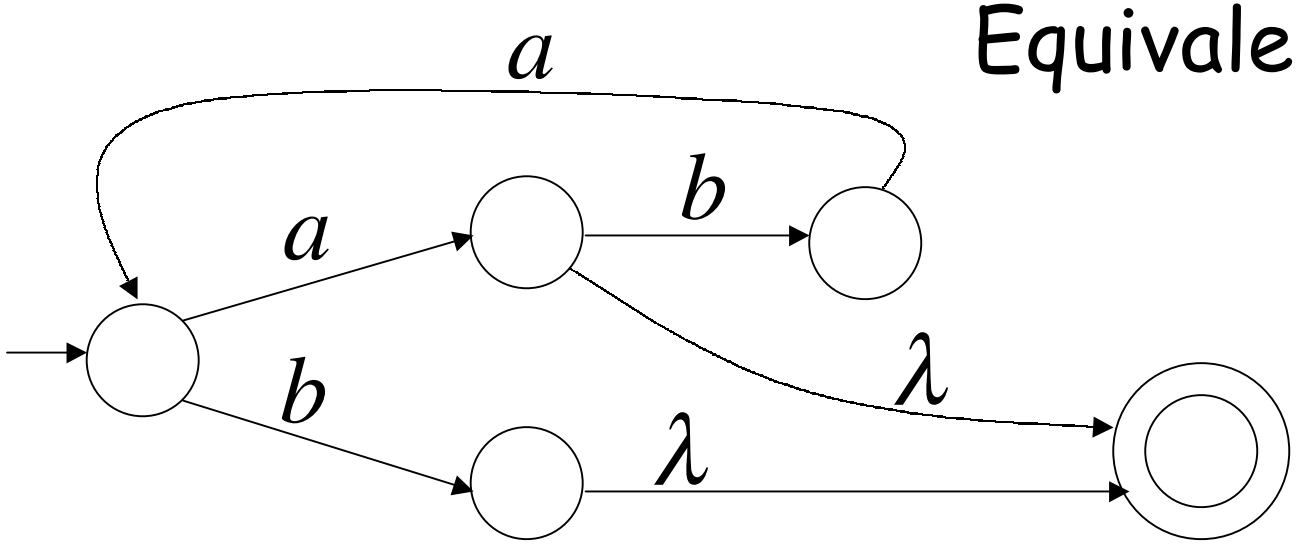
with a single final state

Example

NFA

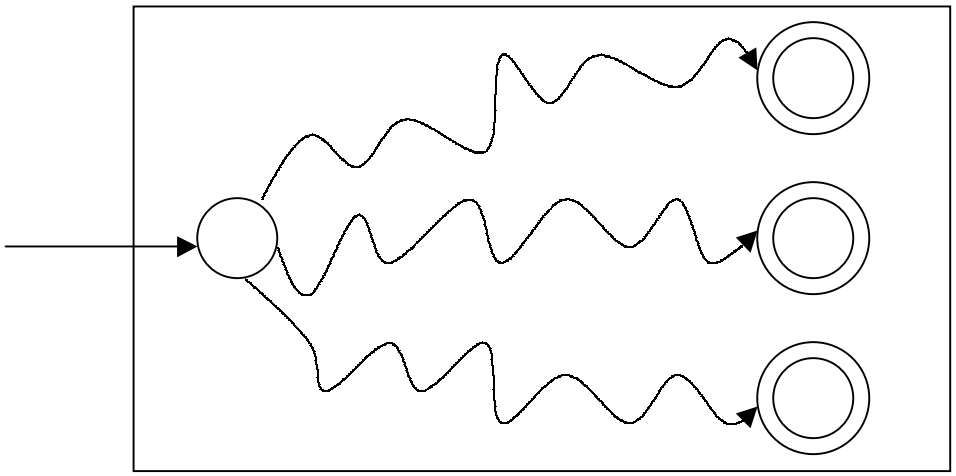


Equivalent NFA

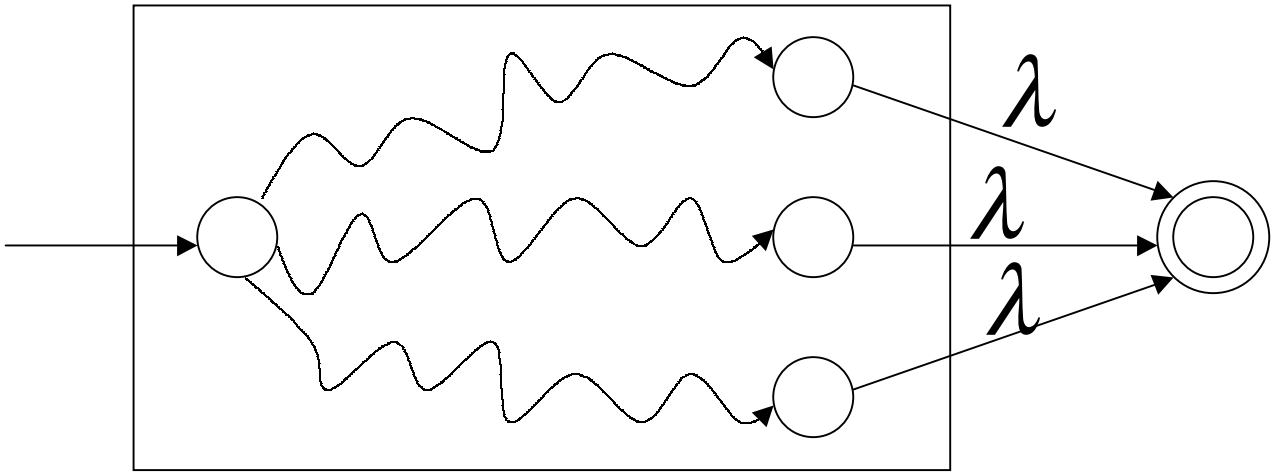


In General

NFA



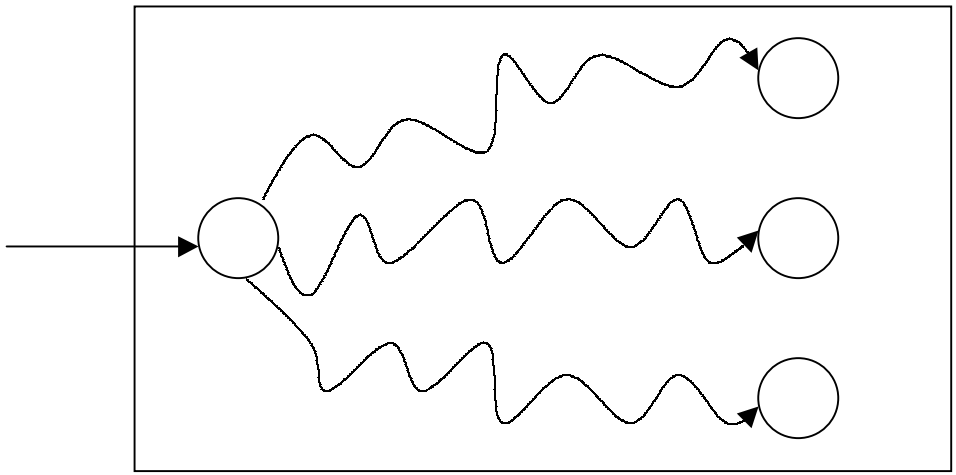
Equivalent NFA

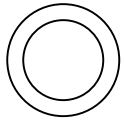


Single final state

Extreme Case

NFA without final state





Add a final state

Properties of Regular Languages

Properties

Take any regular languages L_1 and L_2

We will prove:

Union: $L_1 \cup L_2$

Concatenation: $L_1 L_2$

Star: L_1^*

Complement: $\overline{L_1}$

Intersection: $L_1 \cap L_2$

} Are regular Languages

We Say

Regular Languages are **closed**:

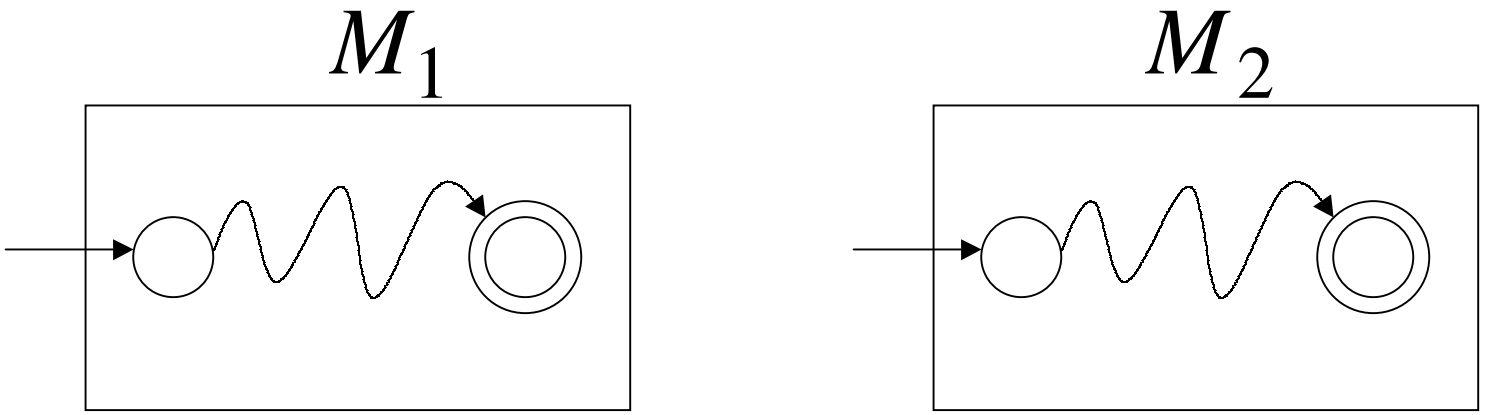
- Under union: $L_1 \cup L_2$
- Under concatenation: $L_1 L_2$
- Under the star operation: $L_1 *$
- Under complement: $\overline{L_1}$
- Under intersection: $L_1 \cap L_2$

For regular languages L_1 and L_2

take NFAs M_1 and M_2 with

$$L(M_1) = L_1$$

$$L(M_2) = L_2$$

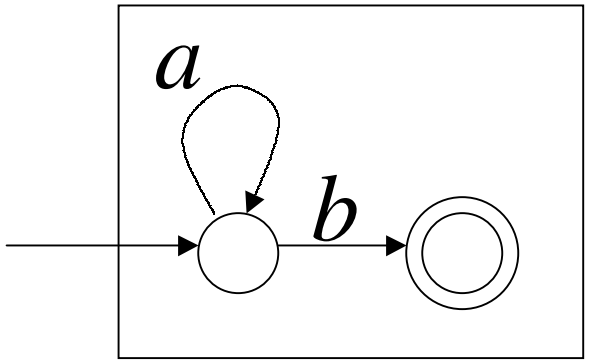


Single final state

Example

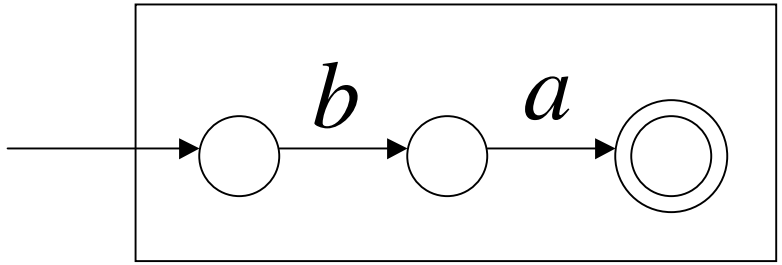
$L_1 = \{a^n b\}$

M_1



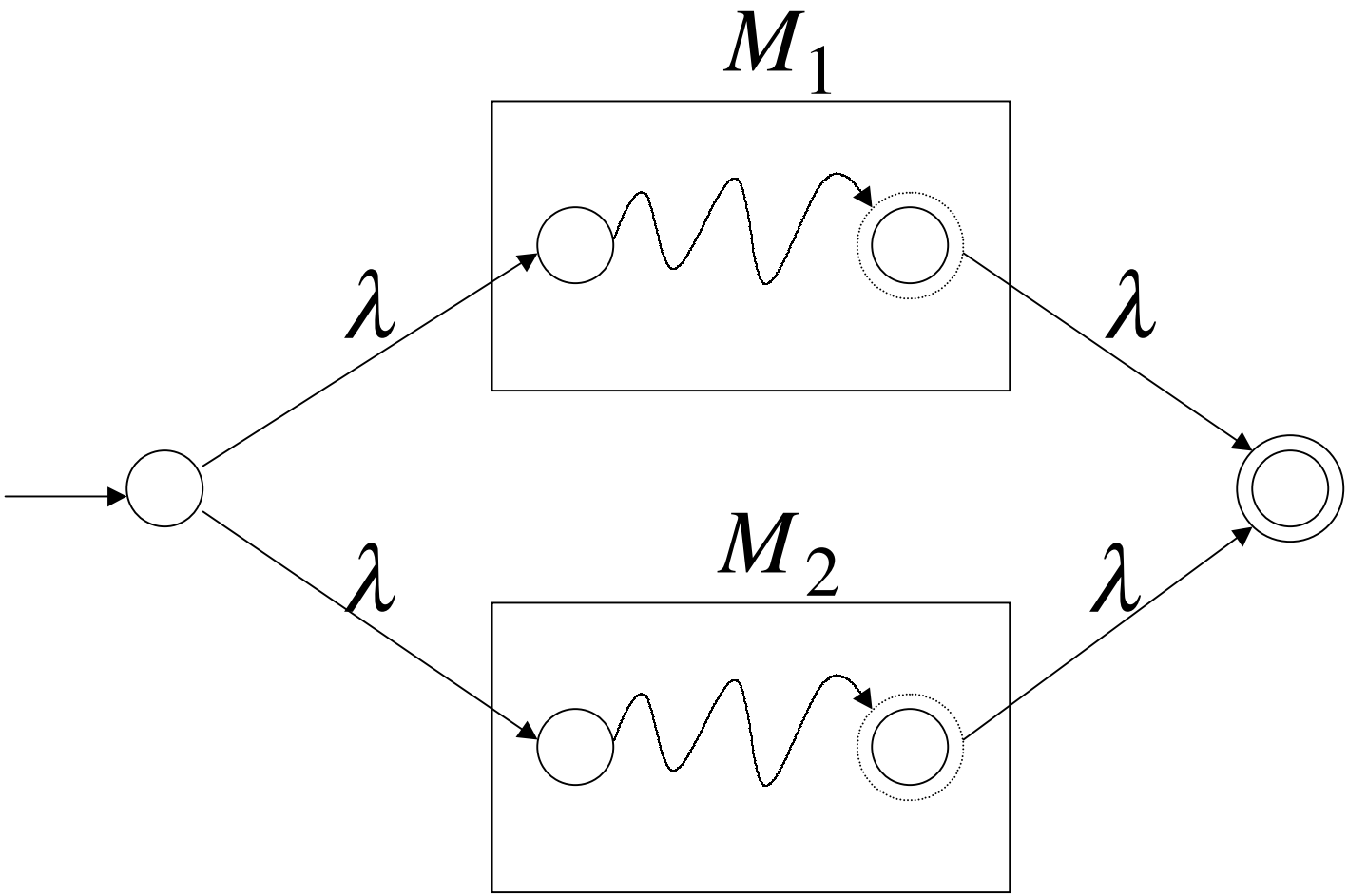
$L_2 = \{ba\}$

M_2



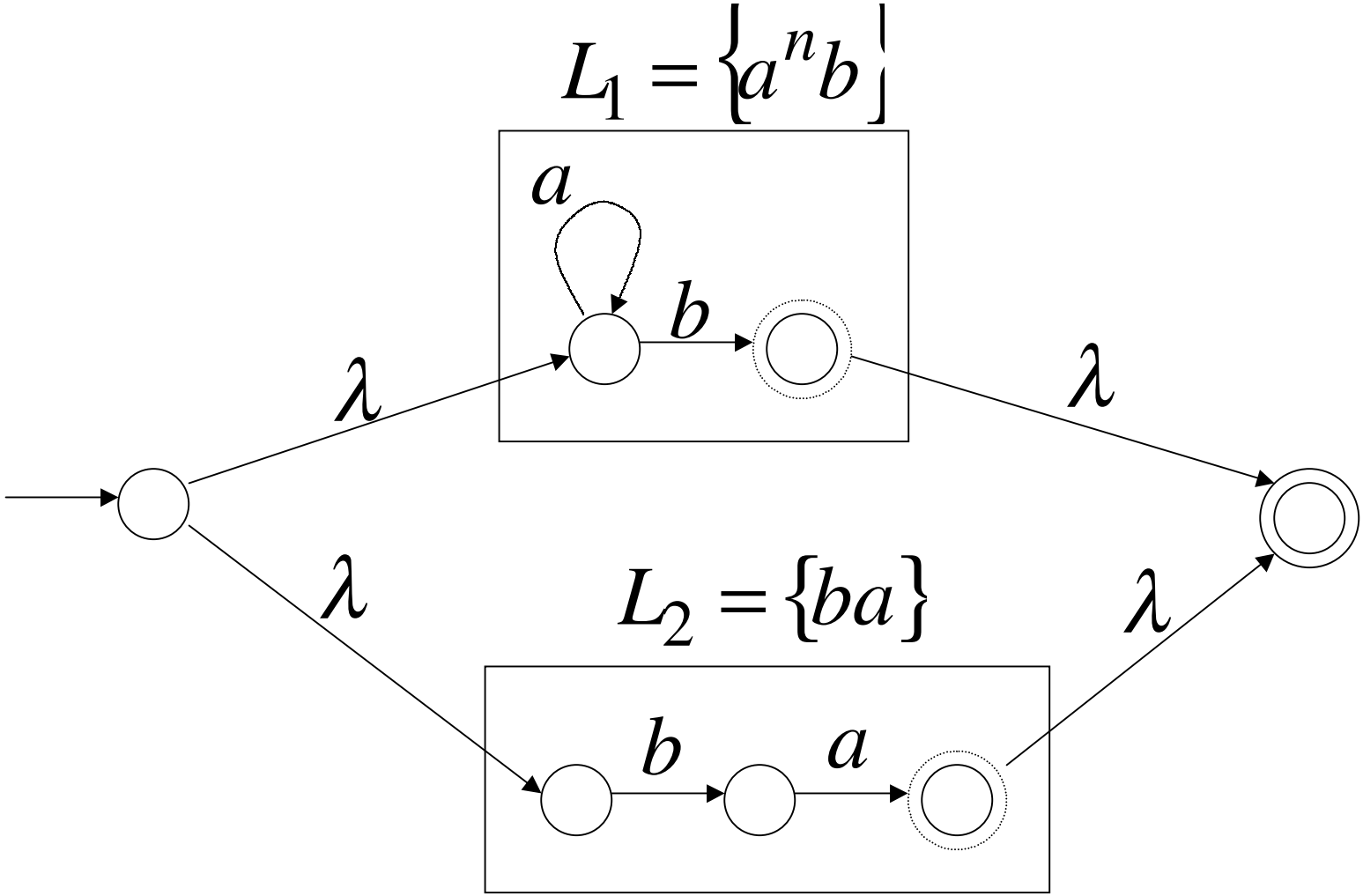
Union

NFA for $L_1 \cup L_2$



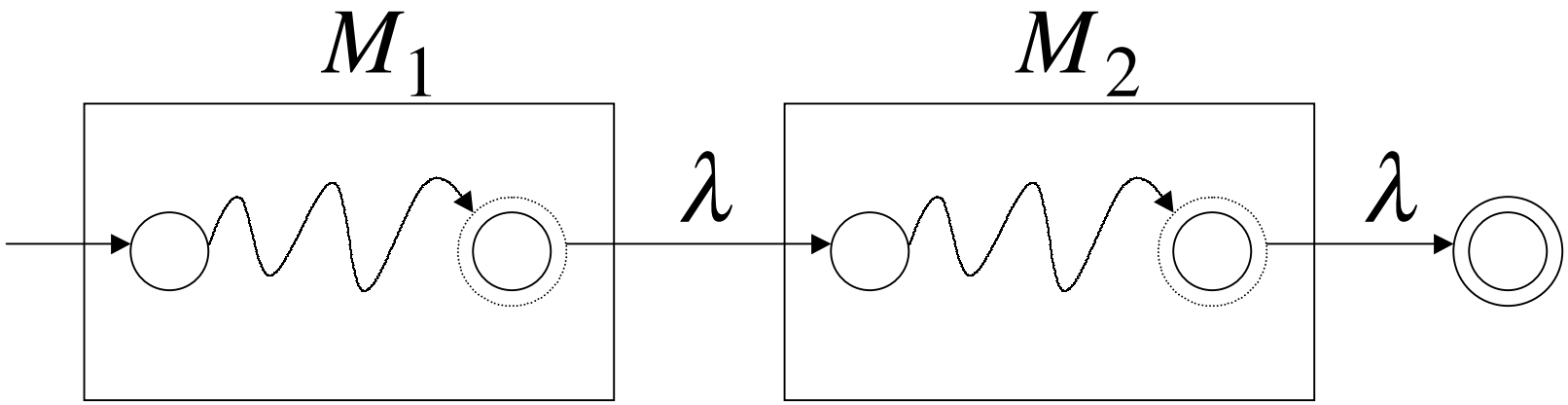
Example

NFA for $L_1 \cup L_2 = \{a^n b\} \cup \{b, a\}$



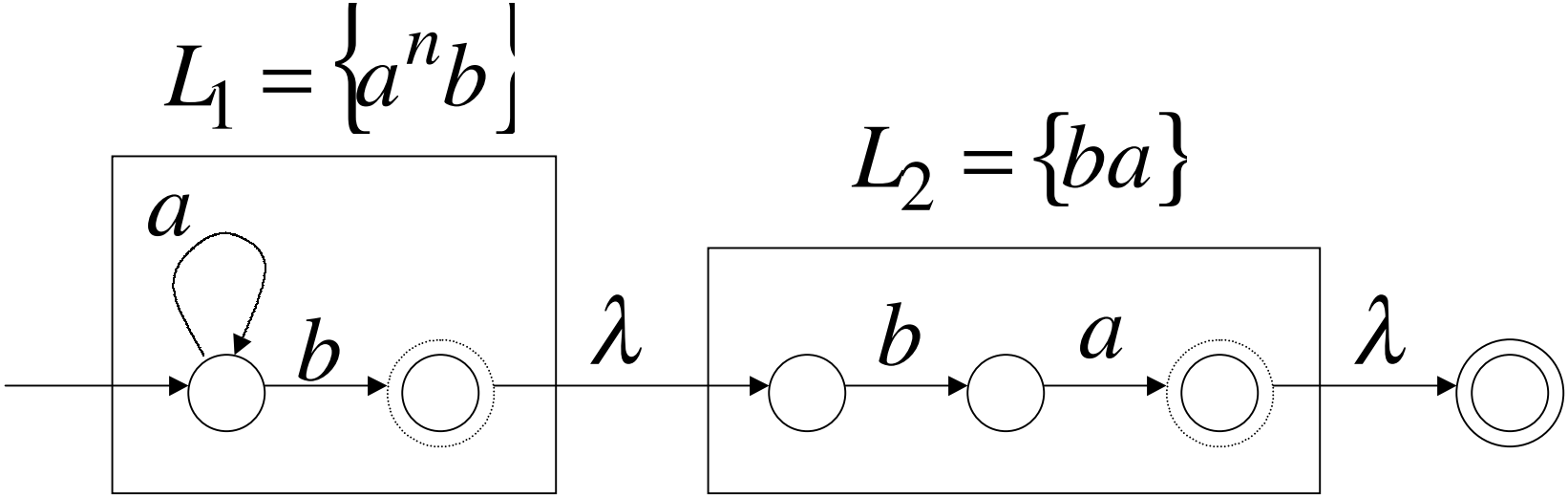
Concatenation

NFA for $L_1 L_2$



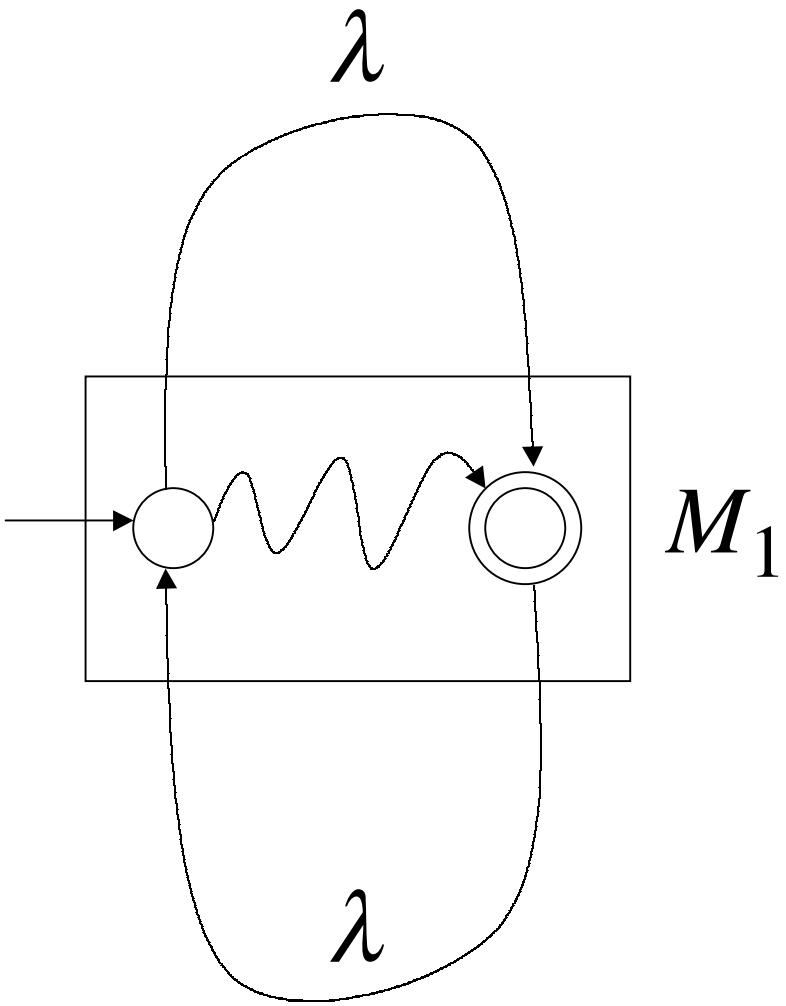
Example

NFA for $L_1L_2 = \{a^n b\}\{ba\} = \{a^n bba\}$



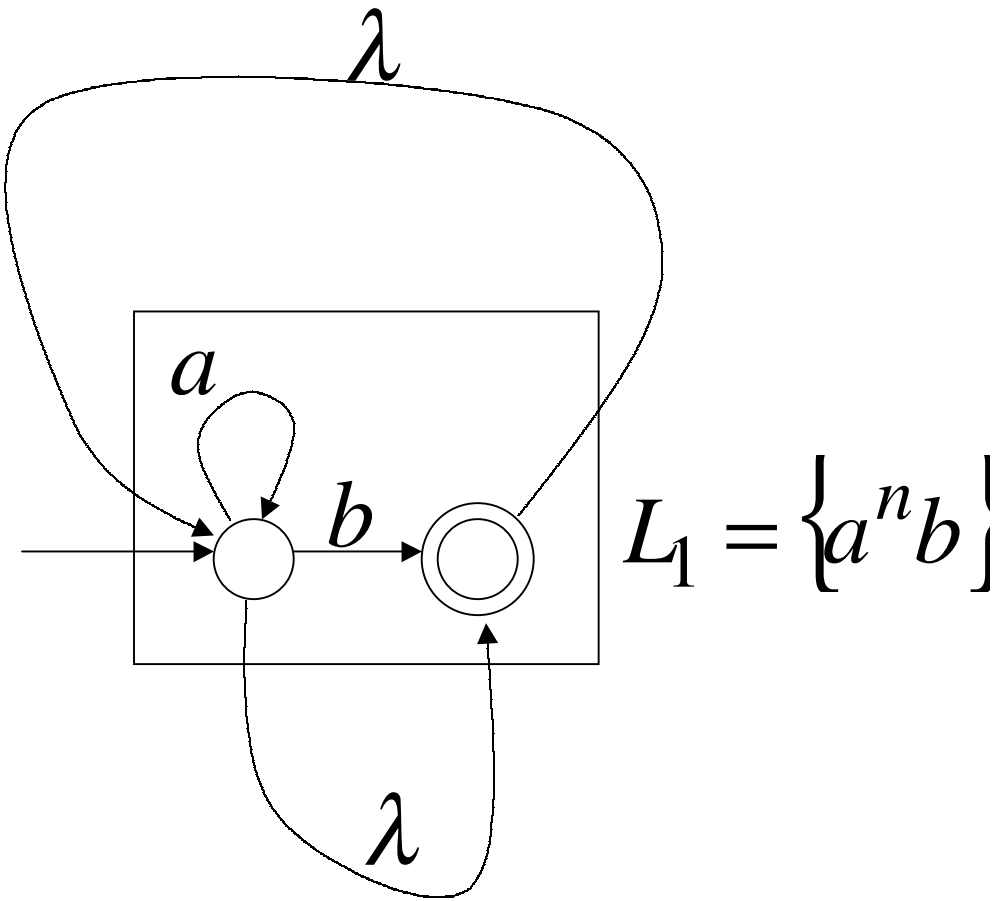
Star Operation

NFA for L_1^*



Example

NFA for $L_1^* = \{a^n b\}^*$



Complement

For the complement $\overline{L}$ of regular language L:

Take the DFA M that accepts L

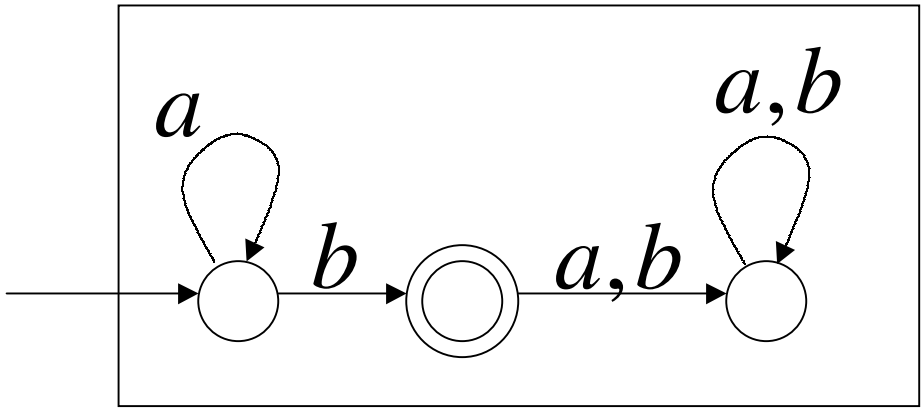
Construct M' such that:

- Each **final** state of M is **nonfinal** in M'

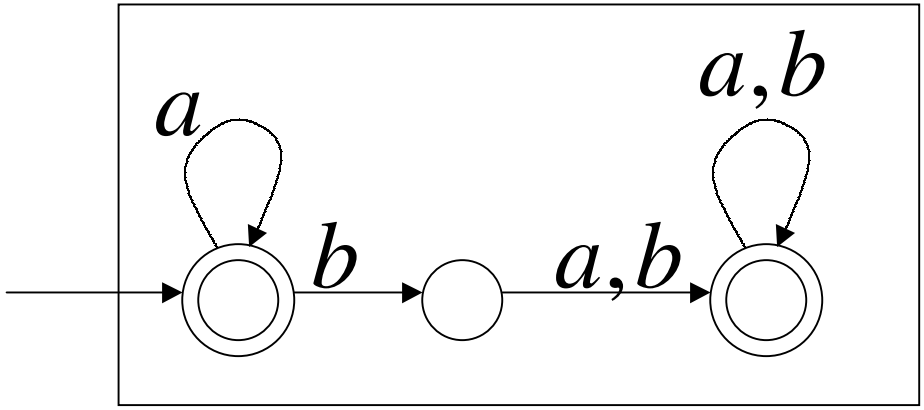
nonfinal **final**

We have: $L(M') = \overline{L(M)} = \overline{L}$

Example



$$L = \{a^n b\}$$



$$\overline{L} = \overline{\{a^n b\}}$$

Intersection

For regular languages L_1 and L_2 :

$$L_1 \cap L_2 = \overline{\overline{L_1 \cap L_2}}$$

$$= \overline{\overline{L_1} \cup \overline{L_2}}$$

regular $L_1 \Rightarrow$ regular $\overline{L_1}$

regular $L_2 \Rightarrow$ regular $\overline{L_2}$

$\Rightarrow \overline{L_1} \cup \overline{L_2}$ (regular) $\Rightarrow \overline{\overline{L_1} \cup \overline{L_2}}$ (regular) $\Rightarrow$

$\Rightarrow L_1 \cap L_2$ (regular)

Example

Regular languages:

$$L_1 = \{a^k b^l\} \qquad L_2 = \{b^m ba\}$$

$$k, l, m \geq 0$$

The language $L_1 \cap L_2 = \{b^m b\}$ is regular

Regular Expressions

Regular Expressions

Regular expressions
are another way of expressing
regular languages

Example: $(a+b\cdot c)^*$

Stands for the language

$$\{a,bc\}^* = \{\lambda, a, bc, aa, abc, bca, \ldots\}$$

Recursive Definition

Regular Expressions:

- Primitive regular expressions: $\emptyset,\ \lambda,\ \alpha$

- Given regular expressions r_1 and r_2

$r_1 + r_2$

$r_1 \cdot r_2$

$r_1 *$

(r_1)

} Are regular expressions

Examples

A regular expression

$$(a + b \cdot c)^* \cdot (c + \varnothing)$$

Not a regular expression

$$(a + b +)$$

Languages of Regular Expressions

$L(r)$: language of regular expression r

Example

$$L((a+b\cdot c)^*)=\{\lambda,a,bc,aa,abc,bca,\ldots\}$$

Definition

For primitive regular expressions:

$$L(\varnothing) = \varnothing$$

$$L(\lambda) = \{\lambda\}$$

$$L(a) = \{a\}$$

Definition (continued)

For regular expressions r_1 and r_2

$$L(r_1 + r_2) = L(r_1) \cup L(r_2)$$

$$L(r_1 \cdot r_2) = L(r_1)\, L(r_2)$$

$$L(r_1 *) = (L(r_1))*$$

$$L((r_1)) = L(r_1)$$

Example

Regular expression: $(a+b)\cdot a^*$

$$
\begin{aligned}
L((a+b)\cdot a^*) &= L((a+b))\,L(a^*) \\
&= L(a+b)\,L(a^*) \\
&= (L(a)\cup L(b))\,(L(a))^* \\
&= (\{a\}\cup\{b\})\,(\{a\})^* \\
&= \{a,b\}\{\lambda,a,aa,aaa,\ldots\} \\
&= \{a,aa,aaa,\ldots,b,ba,baa,\ldots\}
\end{aligned}
$$

Example

Regular expression $r = (a+b)^*(a+bb)$

$$L(r) = \{a, bb, aa, abb, ba, bbb, \ldots\}$$

Example

Regular expression $r = (aa)^*(bb)^*b$

$$L(r) = \left\{ a^{2n} b^{2m} b : \quad n, m \geq 0 \right\}$$

Example

Regular expression $r = (0+1)^*00(0+1)^*$

$L(r)$ = { all strings with at least two consecutive 0 }

Example

Regular expression $r = (1+01)^*(0+\lambda)$

$L(r)$ = { all strings without
two consecutive 0 }

Equivalent Regular Expressions

Definition:

Regular expressions r_1 and r_2

are **equivalent** if $L(r_1) = L(r_2)$

Example

L = { all strings with at least two consecutive 0 }

$$r_1 = (1+01)^*(0+\lambda)$$

$$r_2 = (1^*011^*)^*(0+\lambda)+1^*(0+\lambda)$$

$L(r_1) = L(r_2) = L$ $\Longrightarrow$ r_1 and r_2 are equivalent regular expr.

Regular Expressions and Regular Languages

Theorem

The class of languages
described by Regular expressions
is identical
to the Regular languages

In Other Words

- For any regular expression r the language $L(r)$ is regular

- For any regular language L there is a regular expression r with $L(r) = L$

Proof

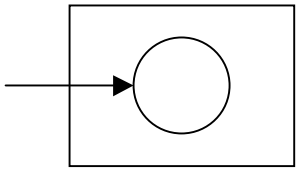
First we prove:

- For any regular expression r the language $L(r)$ is regular

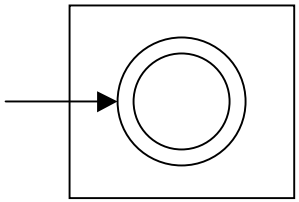
Induction Basis

Primitive Regular Expressions: $\emptyset$, λ, α

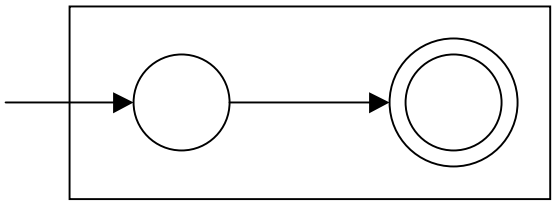
NFAs



$$L(M_1) = \emptyset = L(\emptyset)$$



$$L(M_2) = \{\lambda\} = L(\lambda)$$



$$L(M_3) = \{a\} = L(a)$$

regular languages

Inductive Hypothesis

Assume

for regular expressions r_1 and r_2

that

$L(r_1)$ and $L(r_2)$ are regular languages

Inductive Step

We will prove that:

$L(r_1 + r_2)$

$L(r_1 \cdot r_2)$

$L(r_1 *)$

$L((r_1))$

Are regular Languages

By definition of regular expressions:

$$L(r_1 + r_2) = L(r_1) \cup L(r_2)$$

$$L(r_1 \cdot r_2) = L(r_1)\, L(r_2)$$

$$L(r_1 *) = (L(r_1))*$$

$$L((r_1)) = L(r_1)$$

By inductive hypothesis

$L(r_1)$ and $L(r_2)$ are regular languages

We know:

Regular languages

are closed under

union	$L(r_1) \cup L(r_2)$
concatenation	$L(r_1)\,L(r_2)$
star operation	$(L(r_1))^*$

Therefore:

$$L(r_1 + r_2) = L(r_1) \cup L(r_2)$$

$$L(r_1 \cdot r_2) = L(r_1)\, L(r_2)$$

$$L(r_1 *) = (L(r_1))*$$

Are regular languages

And trivially:

$L((r_1))$ is a regular language

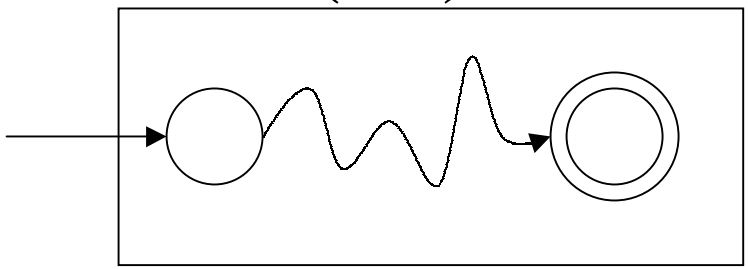
Proof - Second Part

Now we want to prove:

- For any regular language L there is a regular expression r with $L(r) = L$

Since L is regular take the NFA M that accepts it

$$L(M) = L$$



Single final state

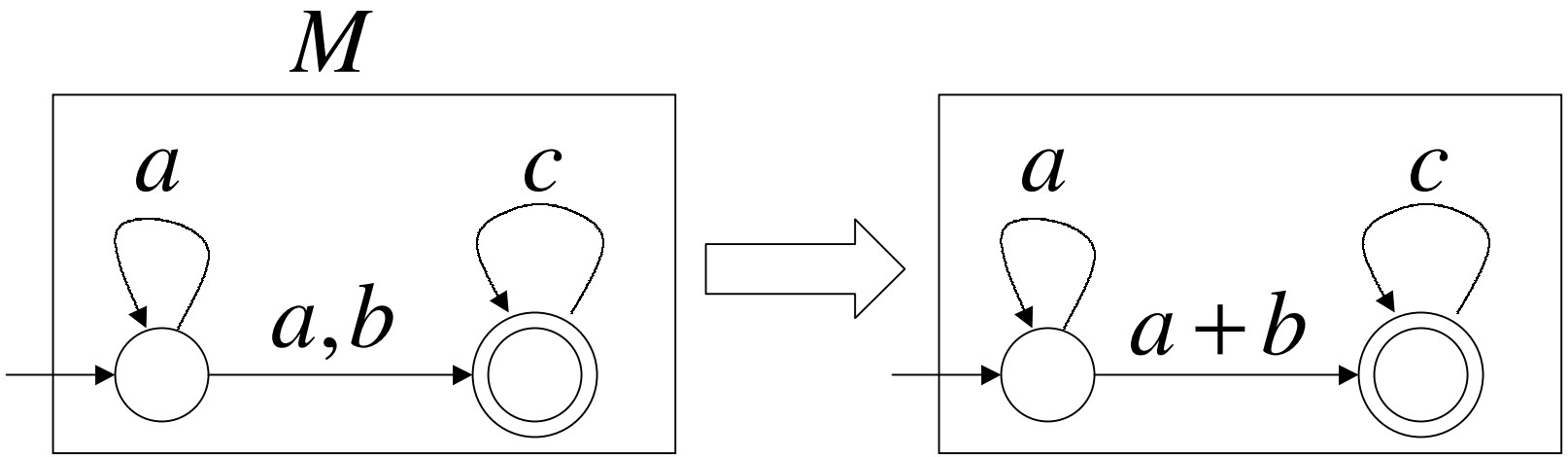
From M Construct the equivalent

Generalized Transition Graph

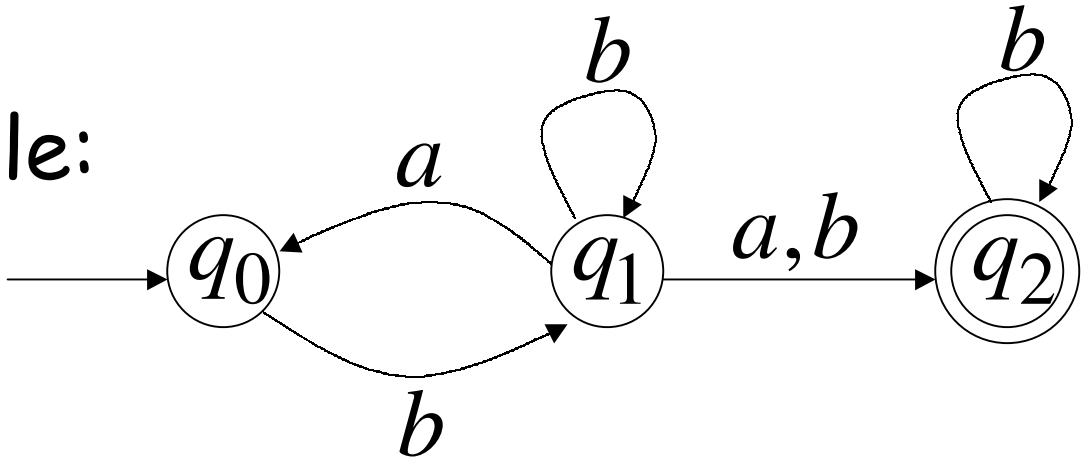
labels of transitions

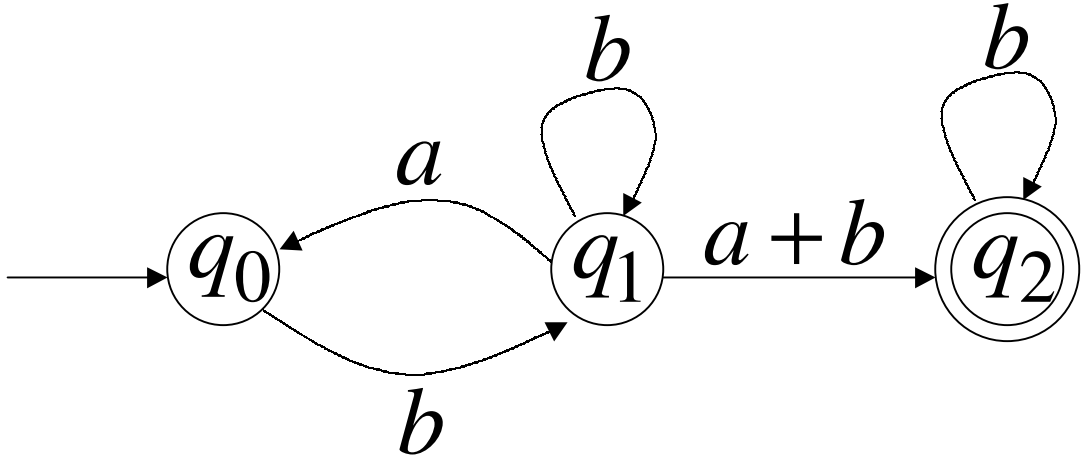
are regular expressions

Example:

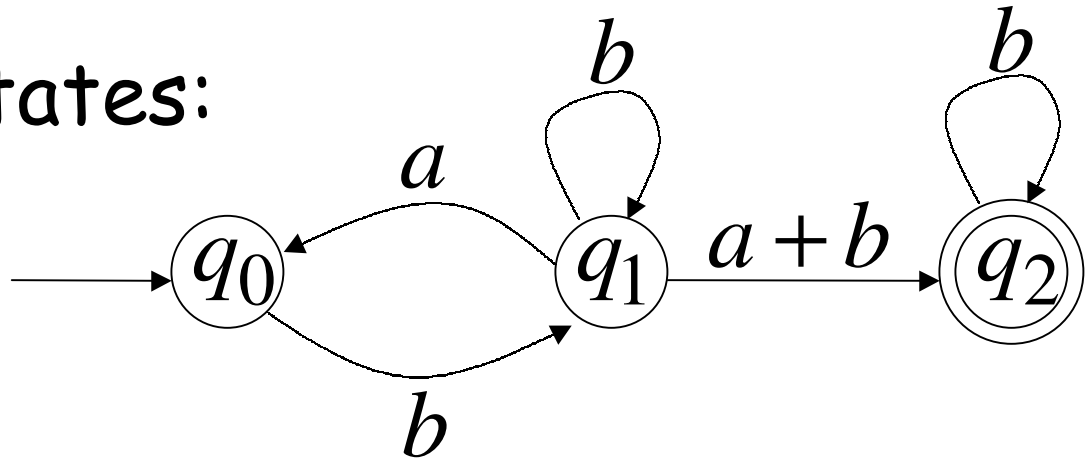


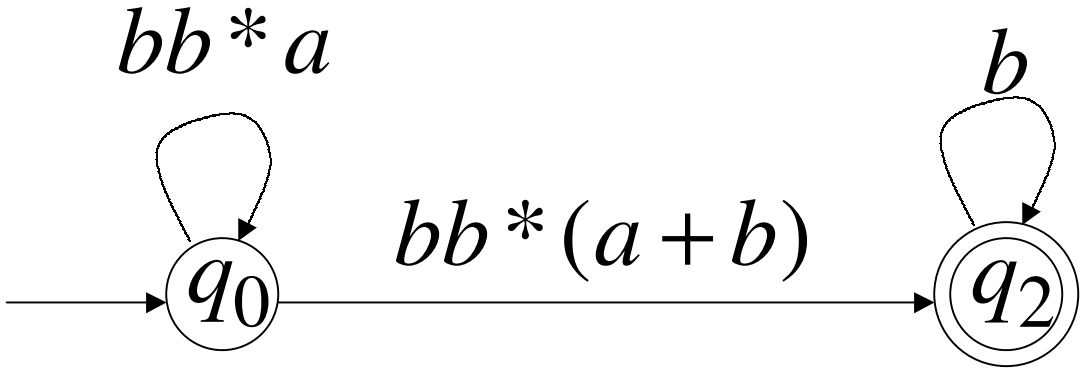
Another Example:



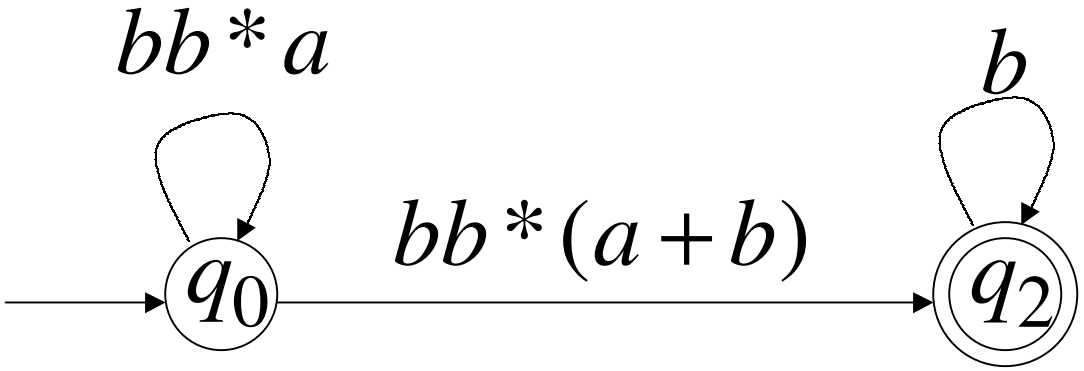


Reducing the states:





Resulting Regular Expression:

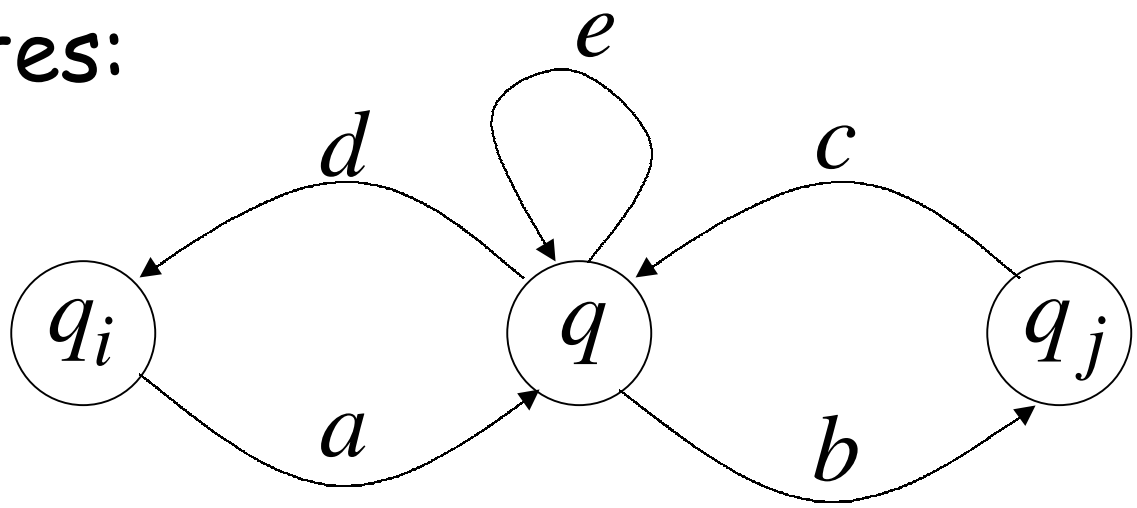


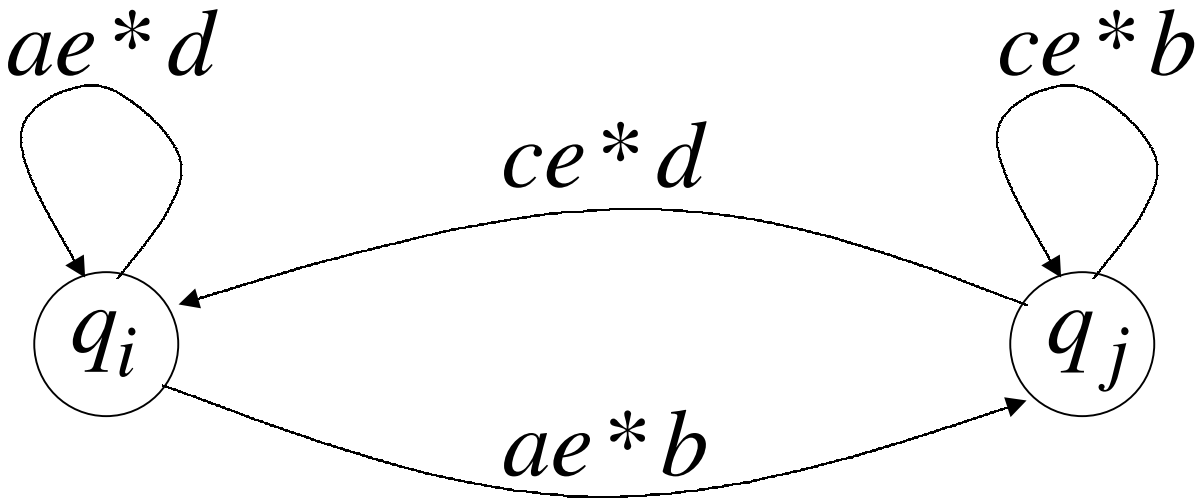
$$r = (bb^*a)^*bb^*(a+b)b^*$$

$$L(r) = L(M) = L$$

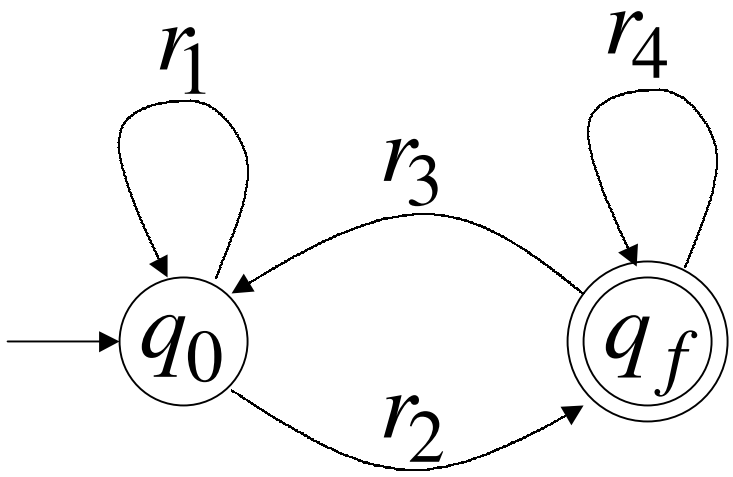
In General

Removing states:





Obtaining the final regular expression:



$$r = r_1^* r_2 (r_4 + r_3 r_1^* r_2)^*$$

$$L(r) = L(M) = L$$