Regular Expressions

Regular Expressions

- Regular expressions describe regular languages

- **Example**

$(a + b \cdot c)^*$ describes the language

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

Regular Expressions

Validation

- checking that an input string is in valid format
- **example 1**: checking format of email address on web entry form
- **example 2**: UNIX *regex* command

Search and selection

- looking for strings that match a certain pattern
- **example**: UNIX *grep* command

Tokenization

- converting sequence of characters (a string) into sequence of tokens (e.g., keywords, identifiers)
- used in lexical analysis phase of compiler

Recursive Definition

- Primitive regular expressions: $\varnothing, \quad \lambda, \quad a$
- Given regular expressions r_1 and r_2

$$\left.\begin{array}{l} r_1 + r_2 \\ r_1 \cdot r_2 \\ r_1^* \\ (r_1) \end{array}\right\} \text{ are regular expressions}$$

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

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: $(a+b)^* \cdot (a+bb)$

$$L\left((a+b)^* \cdot (a+bb)\right)?$$

Example

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

Example

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

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

Example

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

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

Exercises

Let $\Sigma = \{a, b, c\}$. Given the language, find a regular expression.

- All strings with no c that start with b
- All strings containing exactly two a's
- All strings containing no more than three a's

Exercises

Let $\Sigma = \{a, b, c\}$. Given the language, find a regular expression.

- All strings with no c that start with b

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

- All strings containing exactly two a's

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

- All strings containing no more than three a's

$$r = (b + c)^* (\lambda + a)(b + c)^* (\lambda + a)$$
$$(b + c)^* (\lambda + a)(b + c)^*$$

Exercises

Let $\Sigma = \{a, b\}$. Given a regular expression, find the language.

- $r = a^* b$
- $r = (aaa + bba)$
- $r = a(\emptyset + \emptyset ab)^*$

Exercises

Let $\Sigma = \{a, b\}$. Given a regular expression, find the language.

- $r = a^*b$

$$L(r) = \{a^i b: \ i \geq 0\}$$

- $r = (aaa + bba)$

$$L(r) = \{aaa, bba\}$$

- $r = a(\emptyset + \emptyset ab)^*$

$$L(r) = \{a\}(\{\quad\} \cup \{\quad\}\{a\}\{b\})^*$$
$$= \{a\}(\{\quad\}\{a\}\{b\})^* = \{a\}(\{\quad\})^*$$
$$= \{a\}\{\lambda\} = \{a\}$$

Exercises

Let $\Sigma = \{0, 1\}$. Given the language, find a regular expression.

- One or more 0's followed by a 1
- Two or more symbols followed by 3 or more 0's
- All strings that do not end with 01

Exercises

Let $\Sigma = \{0, 1\}$. Given the language, find a regular expression.

- One or more 0's followed by a 1

$$r = 00^*1$$

- Two or more symbols followed by 3 or more 0's

$$r = (0 + 1)(0 + 1)(0 + 1)^*0000^*$$

- All strings that do not end with 01

$$r = \lambda + 0 + 1 + (0 + 1)^*00 + (0 + 1)^*10 + (0 + 1)^*11$$

Regular Expressions and Regular Languages

Theorem

{ Languages generated by reg.exp. } = { Regular languages }

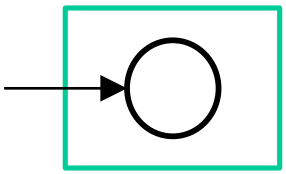
Proof : Part 1

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

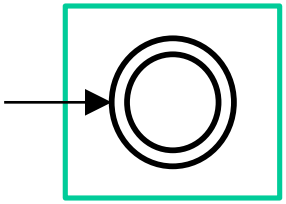
- Proof by induction on the size of r

Induction Basis

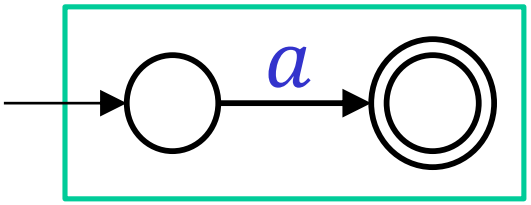
- Primitive regular expressions: $\varnothing$, λ, a
- NFAs:



$$L(M_1) = \varnothing = L(\varnothing)$$



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



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

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:

$$L(r_1 + r_2)$$
$$L(r_1 \cdot r_2)$$
$$L(r_1^*)$$
$$L((r_1))$$

are regular languages

Inductive Step

- By definition of regular expressions:

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

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

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

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

Inductive Step

- By inductive hypothesis we know that

$$L(r_1), \quad L(r_2)$$

are regular languages.

- We also know that regular languages are closed under
  - union $L(r_1) \cup L(r_2)$
  - concatenation $L(r_1)\,L(r_2)$
  - star $(L(r_1))^*$

Inductive Step

- 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.

Example

- Find an NFA that accepts $L(r)$, where

$$r = (a + bb)^*(ba^* + \lambda).$$

Proof : Part 2

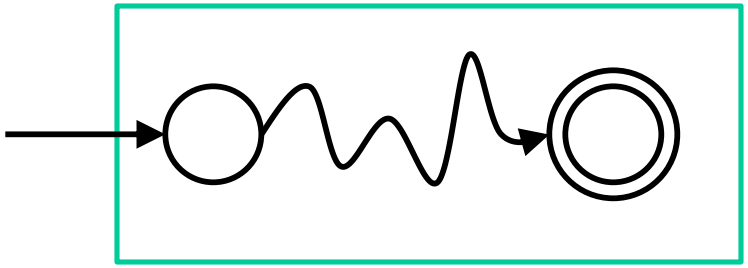
- For any regular language L there is a regular expression r with $L(r) = L$.

- Proof by construction of regular expression.

Proof : Part 2

- Since L is regular then there is an NFA M which accepts the language L.

$$L(M) = L$$

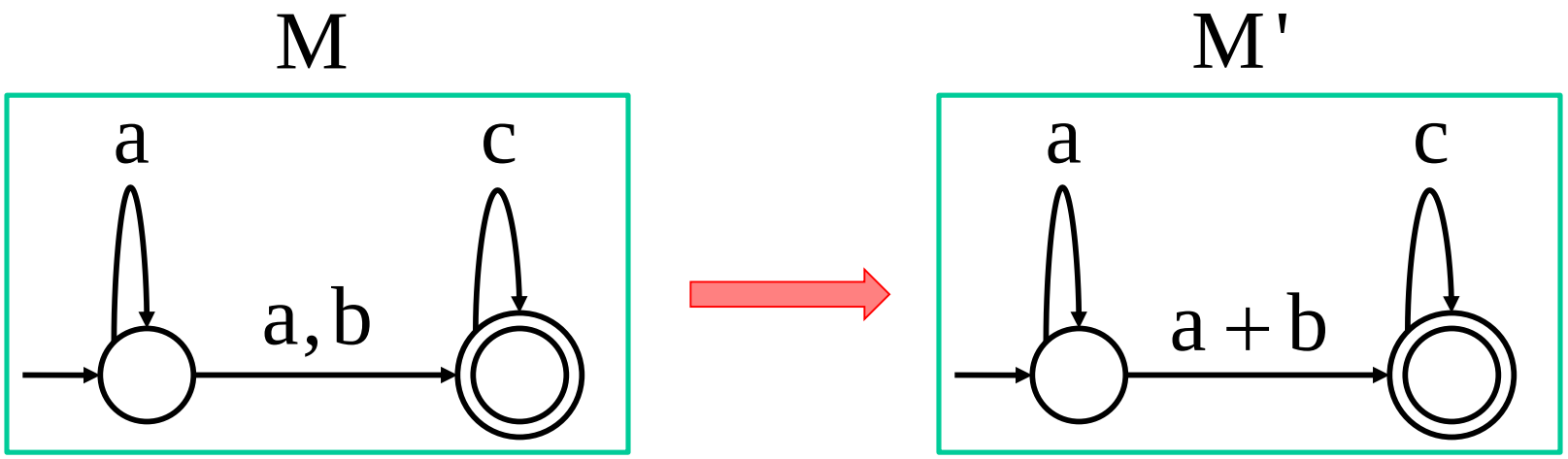


(single final state)

Proof : Part 2

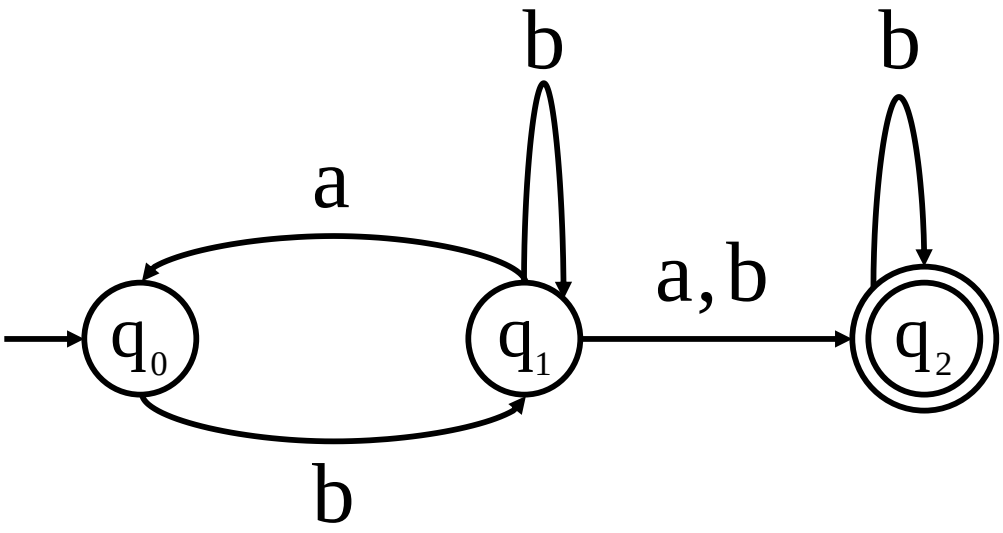
- From M construct the equivalent **Generalized Transition Graph** whose transition labels are regular expressions.

- Example:



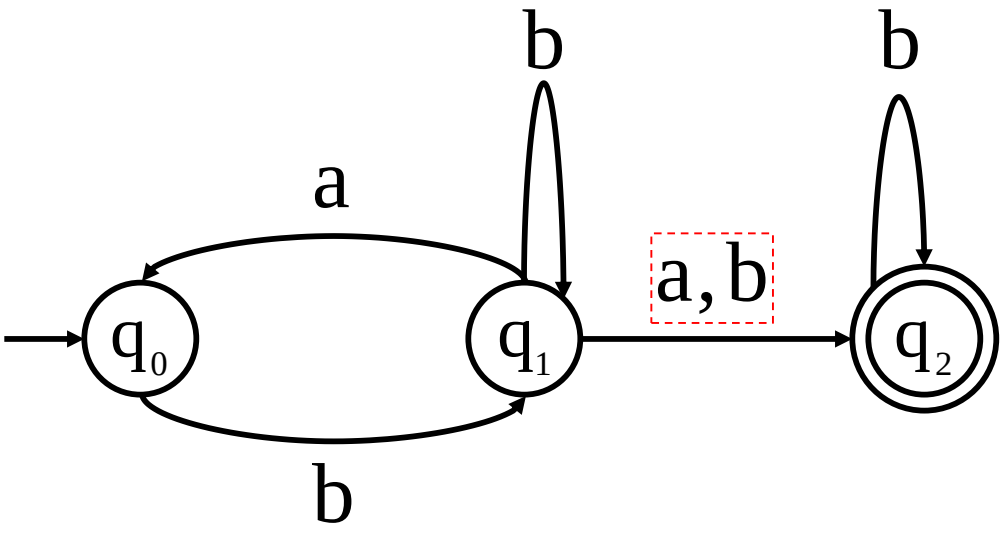
Proof : Part 2

- Example:



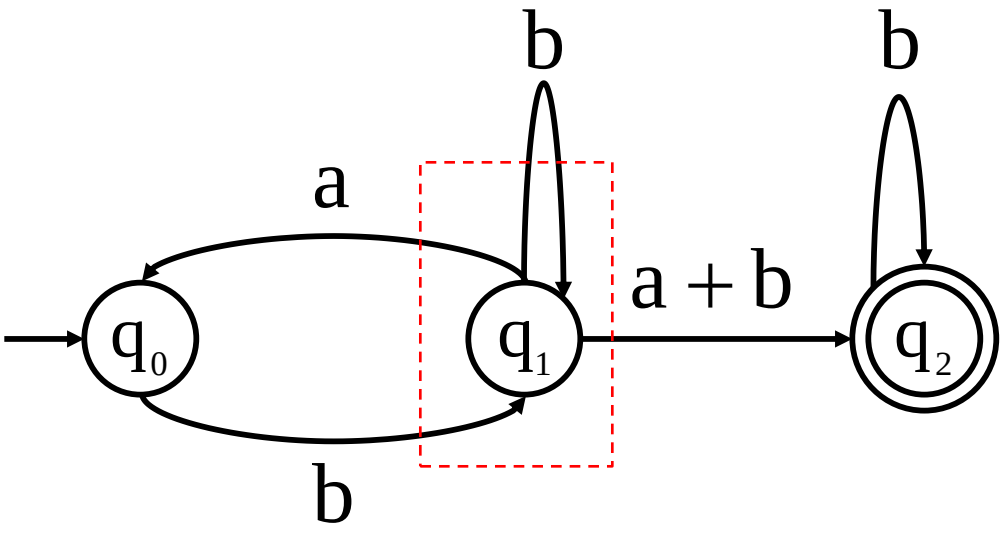
Proof : Part 2

- Example:



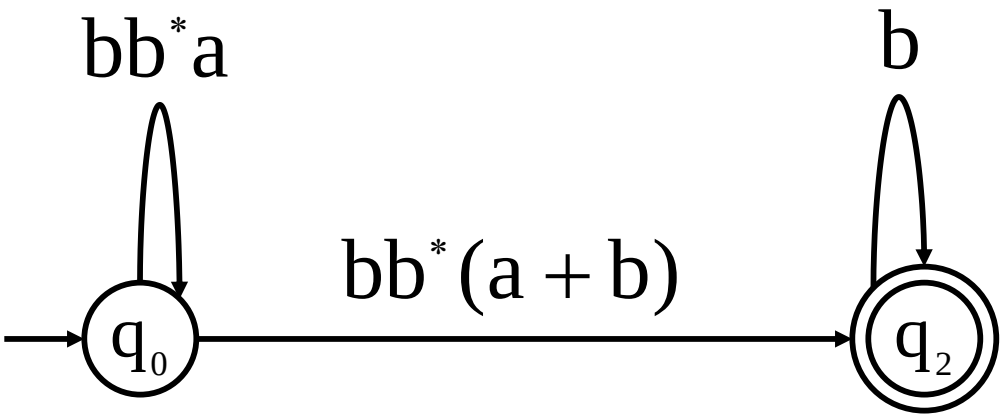
Proof : Part 2

- Example:



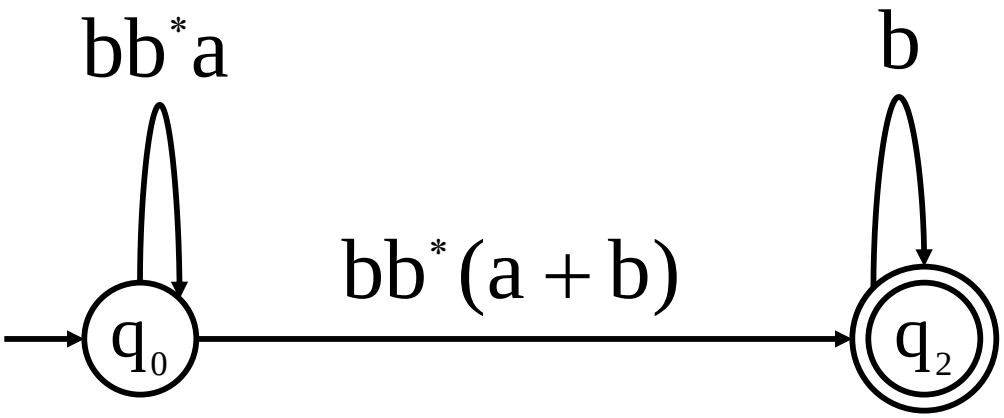
Proof : Part 2

- Reducing the states:



Proof : Part 2

- Resulting Regular Expression:

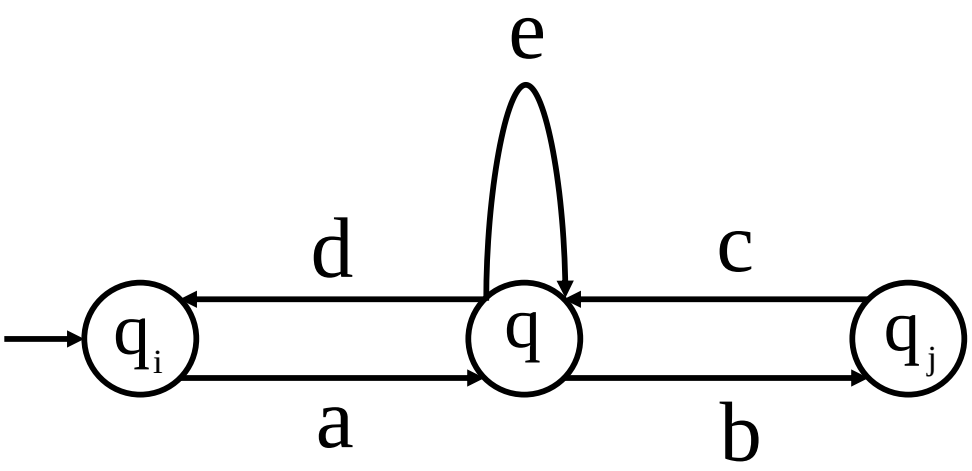


$$r = (bb^*a)^* bb^*(a+b)b^*$$

$$\boxed{L(r) = L(M) = L}$$

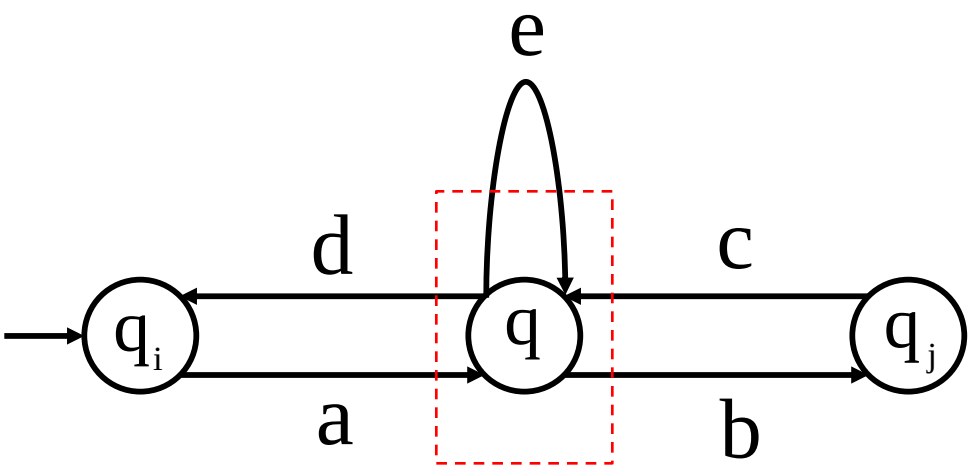
Proof : Part 2

- In general:



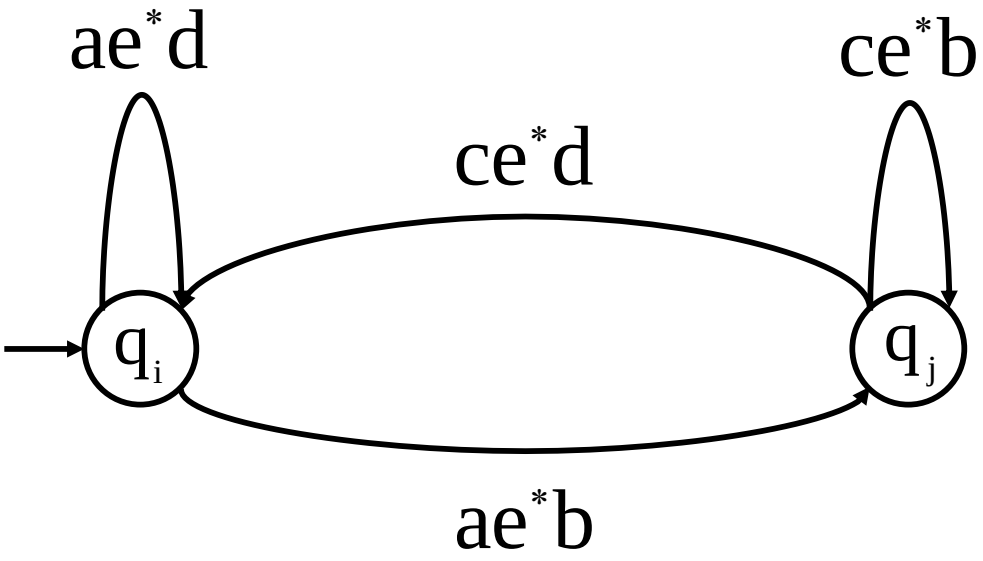
Proof : Part 2

- Removing states:



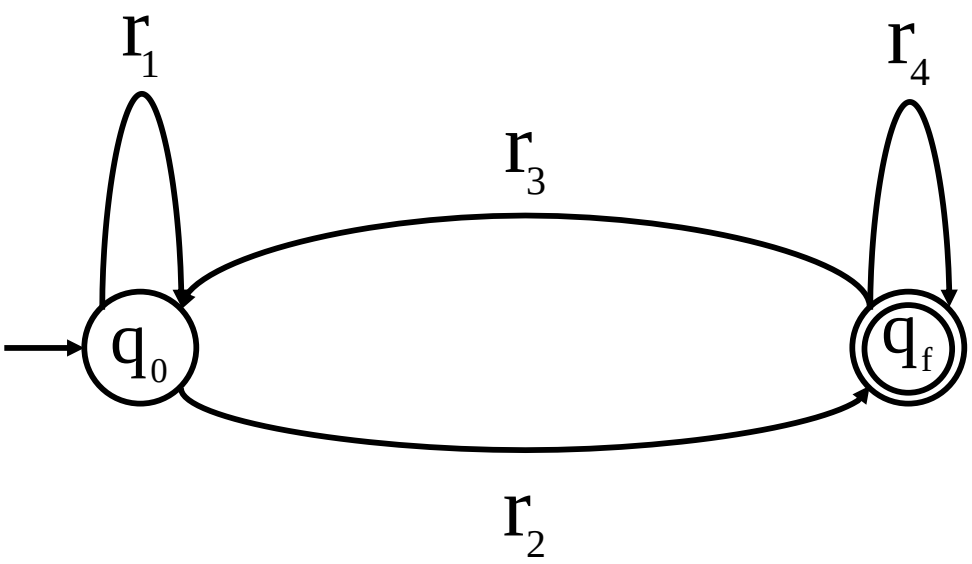
Proof : Part 2

- Removing states:



Proof : Part 2

- The final transition graph:



- The resulting regular expression:

$$r = r_1^* r_2 (r_4 + r_3 r_1^* r_2)^* \qquad L(r) = L(M) = L$$

Exercise

- Find NFA that accept the following language:

$$L(ab^*aa + bba^*ab)$$

- Find a regular expression for the language accepted by the following automaton: