COS210 - Theoretical Computer Science
Finite Automata and Regular Languages (Part 8)

Regular Expressions

The following theorem holds:

Theorem (1)

Let L be a language, then:

$$L \text{ is regular} \iff \text{there exists a regular expression } R \text{ that describes } L$$

$\Longleftarrow$:

Theorem (1A)

Every regular expression R describes a language $L(M)$ where M is a finite automaton.

$\Longrightarrow$:

Theorem (1B)

For every finite automaton M, the language $L(M)$ can be described by a regular expression R.

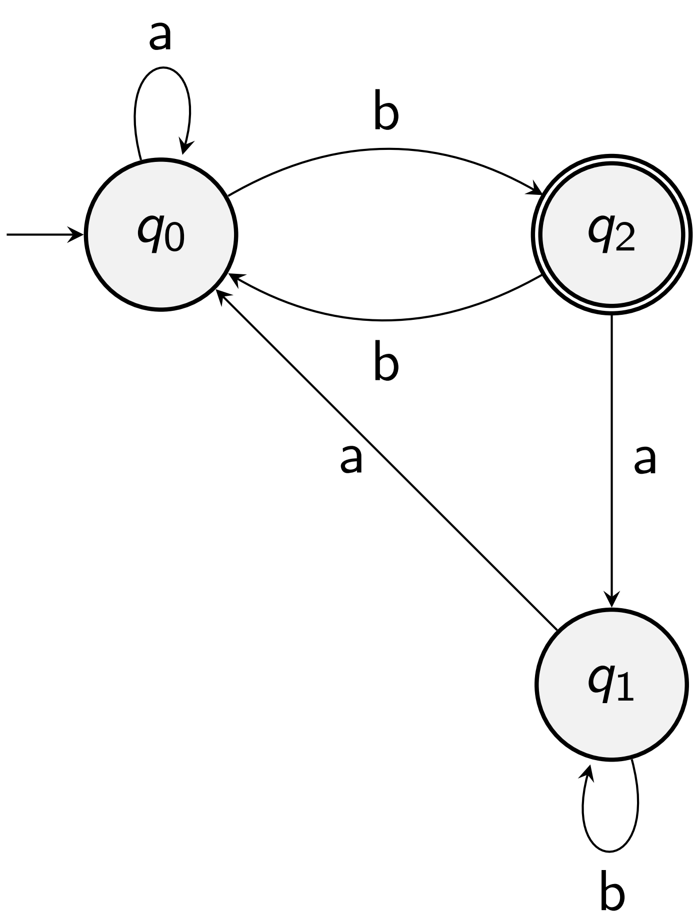
Converting DFA to a Regular Expression

Equations for languages L_r:

$$\boxed{\begin{aligned} L_r &= \bigcup_{a\in\Sigma} a \cdot L_{\delta(r,a)} \quad \text{if } r \notin F \\ L_r &= \left(\bigcup_{a\in\Sigma} a \cdot L_{\delta(r,a)}\right) \cup \epsilon \quad \text{if } r \in F \end{aligned}}$$

For our example:

$$\begin{aligned} L_{q_0} &= a \cdot L_{q_0} \cup b \cdot L_{q_2} \\ L_{q_1} &= a \cdot L_{q_0} \cup b \cdot L_{q_1} \\ L_{q_2} &= a \cdot L_{q_1} \cup b \cdot L_{q_0} \cup \epsilon \end{aligned}$$

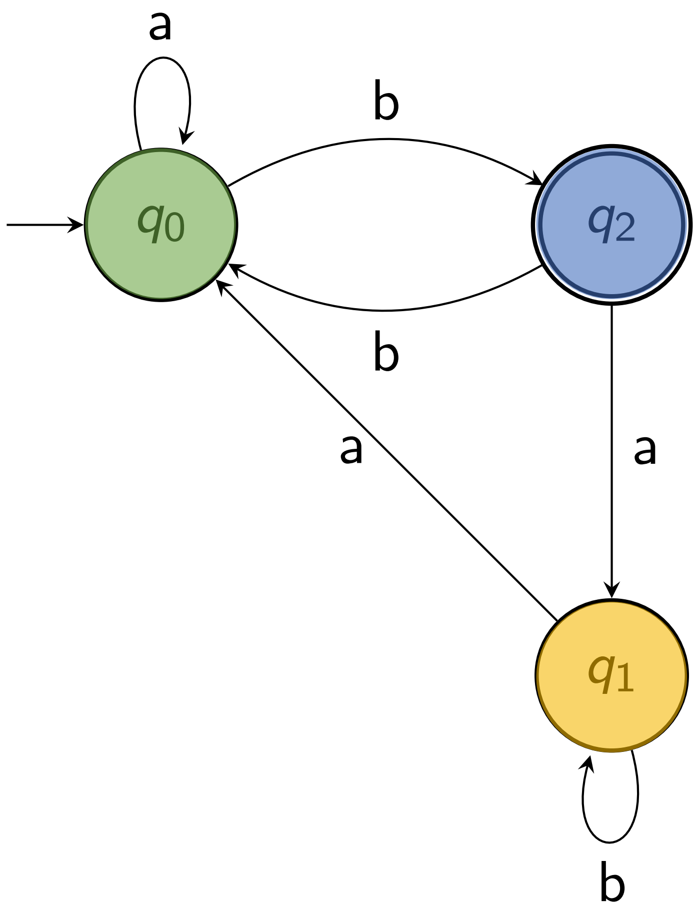


Converting DFA to a Regular Expression

Equations for languages L_r:

$$L_r = \bigcup_{a\in\Sigma} a \cdot L_{\delta(r,a)} \quad \text{if } r \notin F$$

$$L_r = \left(\bigcup_{a\in\Sigma} a \cdot L_{\delta(r,a)}\right) \cup \epsilon \quad \text{if } r \in F$$



For our example:

$$\boxed{L_{q_0} = a \cdot L_{q_0} \cup b \cdot L_{q_2}}$$

$$\boxed{L_{q_1} = a \cdot L_{q_0} \cup b \cdot L_{q_1}}$$

$$\boxed{L_{q_2} = a \cdot L_{q_1} \cup b \cdot L_{q_0} \cup \epsilon}$$

Converting DFA to a Regular Expression

Lemma (2.8.2 (Textbook))

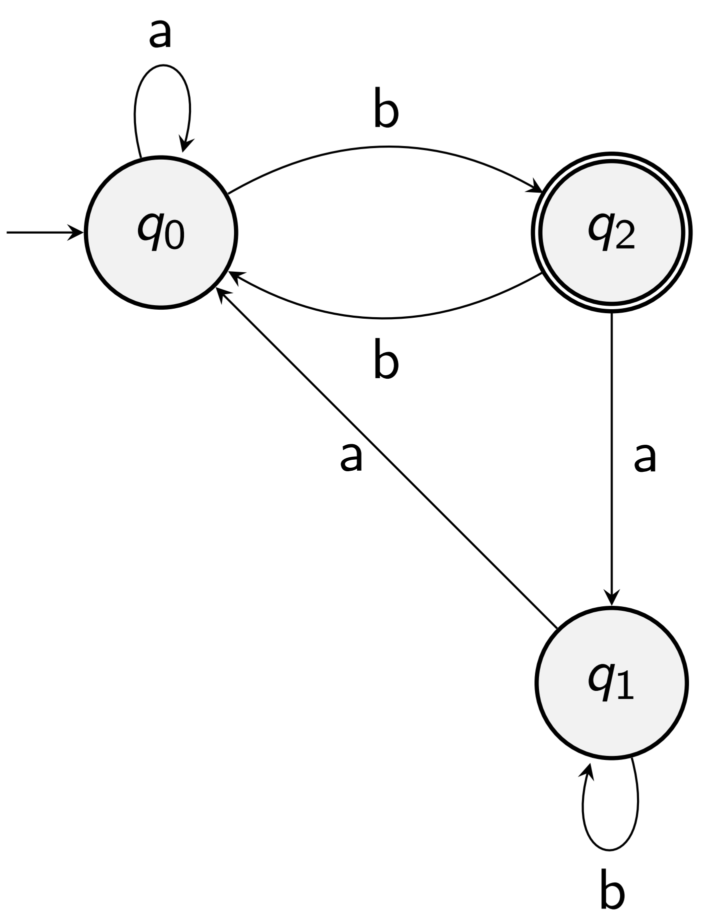
$$L = BL \cup C \quad \Rightarrow \quad L = B^*C$$

Equation system to be solved:

$$\begin{aligned}
\underbrace{L_{q_0}}_{L} &= \underbrace{a \cdot L_{q_0}}_{BL} \cup \underbrace{b \cdot L_{q_2}}_{C} \\
&= \underbrace{a^*}_{B^*} \; \underbrace{b \cdot L_{q_2}}_{C} \\
L_{q_1} &= a \cdot L_{q_0} \cup b \cdot L_{q_1} \\
L_{q_2} &= a \cdot L_{q_1} \cup b \cdot L_{q_0} \cup \epsilon
\end{aligned}$$

Solution for L_{q_0}:

$$L_{q_0} = (a \cup bb \cup bab^*a)^*b$$



Converting DFA to a Regular Expression

Lemma (2.8.2 (Textbook))

$$\boxed{L = BL \cup C} \quad \Rightarrow \quad L = B^* C$$

Equation system to be solved:

$$\boxed{\underbrace{L_{q_0}}_{L} = \underbrace{a \cdot L_{q_0}}_{BL} \cup \underbrace{b \cdot L_{q_2}}_{C}}$$

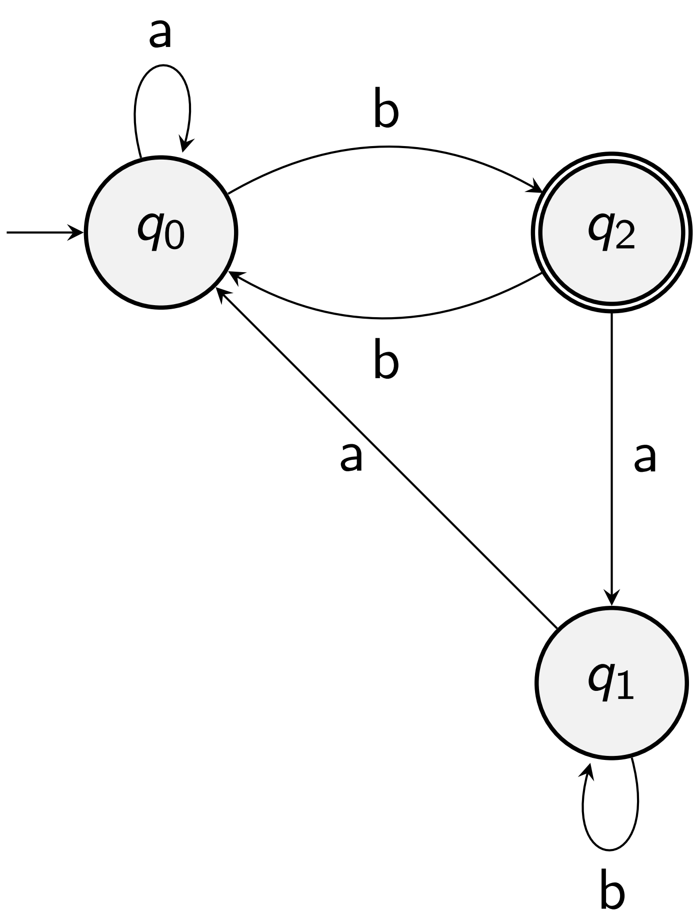
$$= \underbrace{a^*}_{B^*} \; \underbrace{b \cdot L_{q_2}}_{C}$$

$$L_{q_1} = a \cdot L_{q_0} \cup b \cdot L_{q_1}$$

$$L_{q_2} = a \cdot L_{q_1} \cup b \cdot L_{q_0} \cup \epsilon$$

Solution for L_{q_0}:

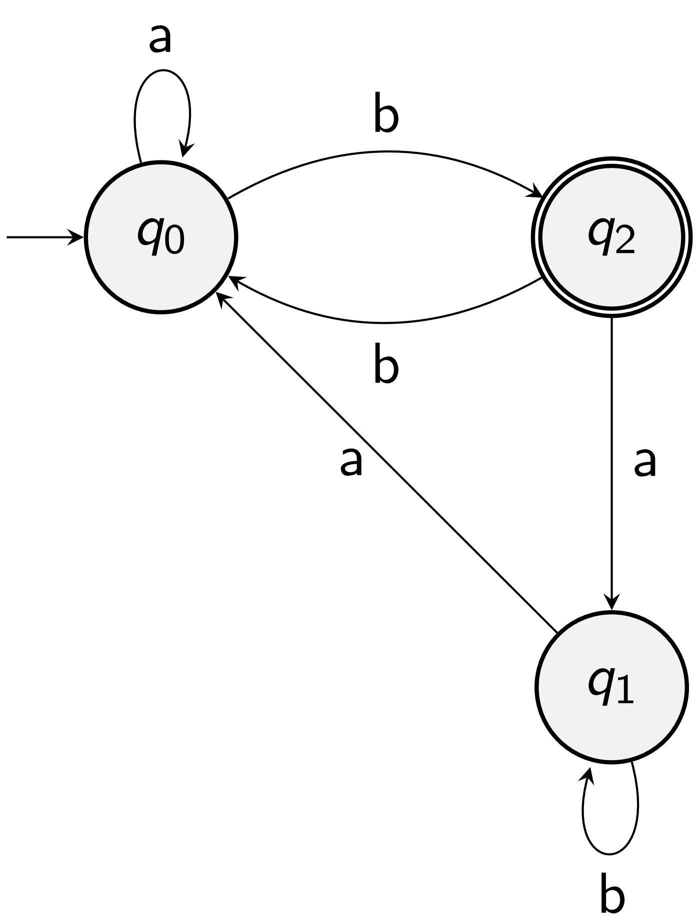
$$L_{q_0} = (a \cup bb \cup bab^* a)^* b$$



Converting DFA to a Regular Expression

Lemma (2.8.2 (Textbook))

$$L = BL \cup C \quad \Rightarrow \quad \boxed{L = B^* C}$$



Equation system to be solved:

$$\begin{aligned}
\underbrace{L_{q_0}}_{L} &= \underbrace{a \cdot L_{q_0}}_{BL} \cup \underbrace{b \cdot L_{q_2}}_{C} \\
&= \boxed{\underbrace{a^*}_{B^*} \underbrace{b \cdot L_{q_2}}_{C}} \\
L_{q_1} &= a \cdot L_{q_0} \cup b \cdot L_{q_1} \\
L_{q_2} &= a \cdot L_{q_1} \cup b \cdot L_{q_0} \cup \epsilon
\end{aligned}$$

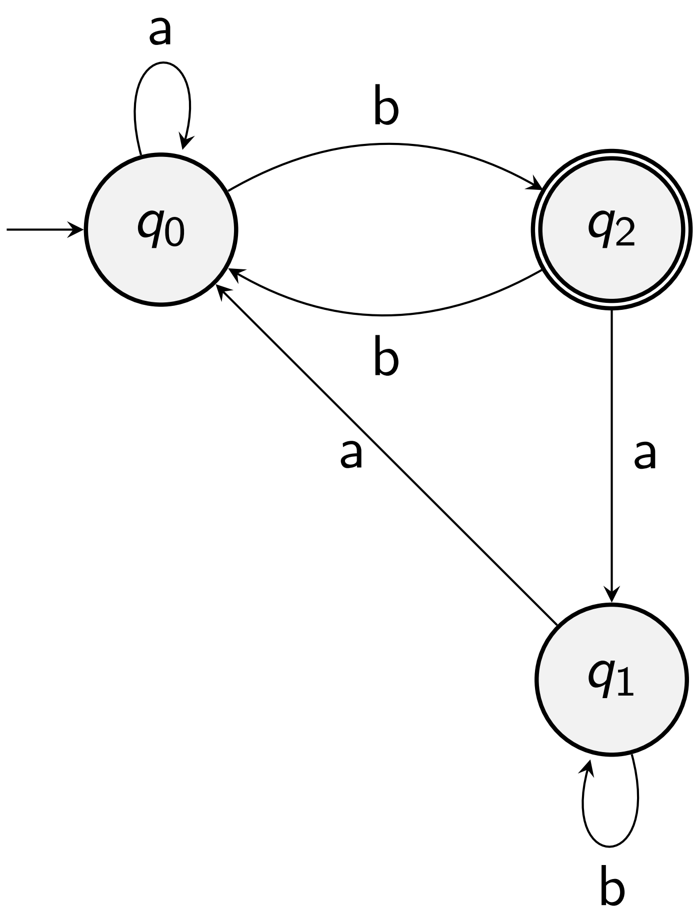
Solution for L_{q_0}:

$$L_{q_0} = (a \cup bb \cup bab^* a)^* b$$

Converting DFA to a Regular Expression

We showed that for the **example** DFA the corresponding equation system can be solved.

We still need to prove that for **arbitrary** DFA's the corresponding equation system always has a solution.



Generalised Equation System

Equations for languages L_r:

$$L_r = \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \text{ if } r \notin F$$

$$L_r = \left(\bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \right) \cup \epsilon \text{ if } r \in F$$

Generalised Equation System

Equations for languages L_r:

$$L_r = \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \quad \text{if } r \notin F$$

$$L_r = \left(\bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \right) \cup \epsilon \quad \text{if } r \in F$$

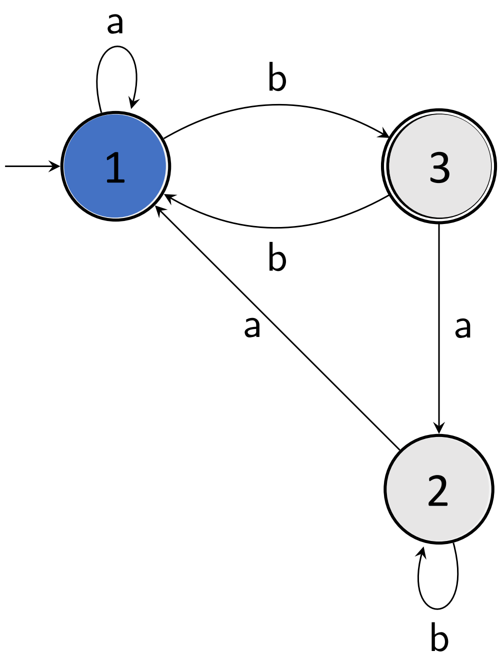
Generalised equations:
Let $Q = \{i : 1 \leq i \leq n\}$ be the states of a DFA, then for each $i \in Q$:

$$L_i = \left(\bigcup_{j=1}^{n} B_{ij} L_j \right) \cup C_i$$

- union over all states
- B_{ij}: symbol of transition from i to j, or $\emptyset$ if there is no transition
- C_i: ϵ if $i \in F$, $\emptyset$ otherwise

Generalised Equation System

Equations for languages L_r:



$$L_r = \bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \quad \text{if } r \notin F$$

$$L_r = \left(\bigcup_{a \in \Sigma} a \cdot L_{\delta(r,a)} \right) \cup \epsilon \quad \text{if } r \in F$$

Generalised equations:
Let $Q = \{i : 1 \leq i \leq n\}$ be the states of a DFA, then for each $i \in Q$:

$$L_i = \left(\bigcup_{j=1}^{n} B_{ij} L_j \right) \cup C_i$$

$L_1 = aL_1 \cup \emptyset L_2 \cup bL_3 \cup \emptyset$

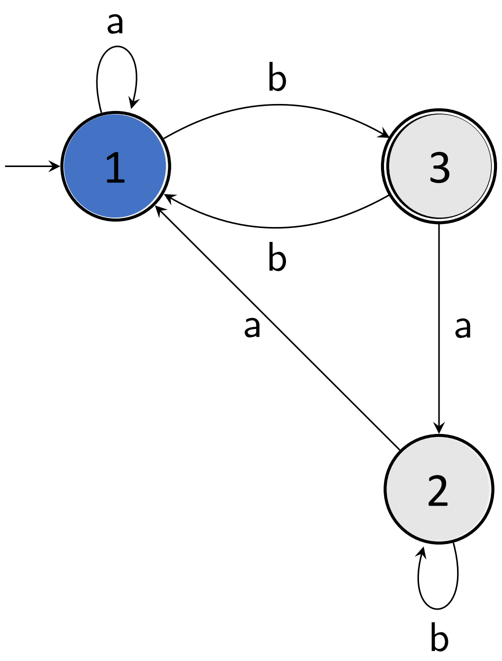
- union over all states
- B_{ij}: symbol of transition from i to j, or $\emptyset$ if there is no transition
- C_i: ϵ if $i \in F$, $\emptyset$ otherwise

Generalised Equation System

Equations for languages L_r:

$$L_r = \bigcup_{a\in\Sigma} a \cdot L_{\delta(r,a)} \quad \text{if } r \notin F$$

$$L_r = \left(\bigcup_{a\in\Sigma} a \cdot L_{\delta(r,a)}\right) \cup \epsilon \quad \text{if } r \in F$$



Generalised equations:
Let $Q = \{i : 1 \le i \le n\}$ be the states of a DFA, then for each $i \in Q$:

$$L_i = \left(\bigcup_{j=1}^{n} B_{ij} L_j\right) \cup C_i$$

$$L_1 = aL_1 \cup \emptyset L_2 \cup bL_3 \cup \emptyset$$

$$= aL_1 \cup bL_3$$

- union over all states
- B_{ij}: symbol of transition from i to j, or $\emptyset$ if there is no transition
- C_i: ϵ if $i \in F$, $\emptyset$ otherwise

Solution to a System of Regular Expressions

Lemma (2.8.3 (Textbook))

Let $n \in \mathbb{N}$, let B_{ij} and C_i be regular expressions where $\epsilon \notin B_{ij}$ and $1 \leq i, j \leq n$, and let $L_1, \ldots, L_n$ be languages that satisfy

$$L_i = \left(\bigcup_{j=1}^{n} B_{ij} L_j \right) \cup C_i \quad \text{for} \quad 1 \leq i \leq n.$$

Then the language L_1 can be written as a regular expression over B_{ij} and C_i only.

Solution to a System of Regular Expressions

Proof by Induction:

Base case: $n = 1$

- If $n = 1$, then there is only one equation:

$$L_1 \;=\; B_{11} L_1 \cup C_1$$

Solution to a System of Regular Expressions

Proof by Induction:

Base case: $n = 1$

- If $n = 1$, then there is only one equation:

$$L_1 = B_{11} L_1 \cup C_1$$

- We can use lemma 2.8.2:

$$\begin{aligned} \underbrace{L_1}_{L} &= \underbrace{B_{11} L_1}_{BL} \cup \underbrace{C_1}_{C} \\ &= \underbrace{B_{11}^*}_{B^*} \underbrace{C_1}_{C} \end{aligned}$$

- It follows that the base case holds.

Solution to a System of Regular Expressions

Hypothesis:

- Assume that for languages $L_1, \ldots, L_{n-1}$, the lemma holds (i.e. the language L_1 can be written as a regular expression).

Solution to a System of Regular Expressions

Hypothesis:

- Assume that for languages $L_1, \ldots, L_{n-1}$, the lemma holds (i.e. the language L_1 can be written as a regular expression).

Inductive Step (informal):

- Show that under the hypothesis the lemma also holds for languages $L_1, \ldots, L_{n-1}, L_n$:

Solution to a System of Regular Expressions

Hypothesis:

- Assume that for languages $L_1, \ldots, L_{n-1}$, the lemma holds (i.e. the language L_1 can be written as a regular expression).

Inductive Step (informal):

- Show that under the hypothesis the lemma also holds for languages $L_1, \ldots, L_{n-1}, L_n$:
  - consider the equation for language L_n
  - apply **equivalence transformations** to the equation for L_n such that L_n occurs on the left-hand side only
  - **substitute** the L_n into the equations for $L_1, \ldots, L_{n-1}$
  - apply **equivalence transformations** to equations for $L_1, \ldots, L_{n-1}$ such that each is of the form

$$L_i = \left(\bigcup_{j=1}^{n-1} B_{ij} L_j \right) \cup C_i$$

  - it follows from the hypothesis that L_1 can be written as a regular expression

Solution to a System of Regular Expressions

Inductive Step (formal):
consider the equation for language L_n:

$$L_n = \left(\bigcup_{j=1}^{n} B_{nj} L_j\right) \cup C_n$$

$$\boxed{= B_{n1}L_1 \cup \ldots \cup B_{nn}\boldsymbol{L_n} \cup C_n}$$

Solution to a System of Regular Expressions

Inductive Step (formal):
consider the equation for language L_n:

$$\begin{aligned} L_n &= \left(\bigcup_{j=1}^{n} B_{nj} L_j\right) \cup C_n && \text{(move } B_{nn}L_n \text{ outside brackets)} \\ &= B_{nn}L_n \cup \left(\bigcup_{j=1}^{n-1} B_{nj} L_j\right) \cup C_n \end{aligned}$$

Solution to a System of Regular Expressions

Inductive Step (formal):
consider the equation for language L_n:

$$\begin{aligned} L_n &= \left(\bigcup_{j=1}^{n} B_{nj} L_j \right) \cup C_n && \text{(move } B_{nn} L_n \text{ outside brackets)} \\ &= \underbrace{B_{nn} L_n}_{BL} \cup \underbrace{\left(\bigcup_{j=1}^{n-1} B_{nj} L_j \right) \cup C_n}_{C} && \text{(apply } BL \cup C \;\Rightarrow\; B^* C\text{)} \\ &= B_{nn}^{*} \left(\left(\bigcup_{j=1}^{n-1} B_{nj} L_j \right) \cup C_n \right) \end{aligned}$$

Solution to a System of Regular Expressions

Inductive Step (formal):
consider the equation for language L_n:

$$\begin{aligned}
L_n &= \left(\bigcup_{j=1}^{n} B_{nj}L_j\right) \cup C_n && \text{(move } B_{nn}L_n \text{ outside brackets)} \\
&= \underbrace{B_{nn}L_n}_{BL} \cup \underbrace{\left(\bigcup_{j=1}^{n-1} B_{nj}L_j\right) \cup C_n}_{C} && \text{(apply } BL \cup C \;\Rightarrow\; B^*C\text{)} \\
&= B_{nn}^*\left(\left(\bigcup_{j=1}^{n-1} B_{nj}L_j\right) \cup C_n\right) && \text{(apply } b(a \cup c) = ba \cup bc\text{)} \\
&= B_{nn}^*\left(\bigcup_{j=1}^{n-1} B_{nj}L_j\right) \cup B_{nn}^*C_n
\end{aligned}$$

Solution to a System of Regular Expressions

Inductive Step (formal):
consider the equation for language L_n:

$$
\begin{aligned}
L_n &= \left(\bigcup_{j=1}^{n} B_{nj} L_j\right) \cup C_n && \text{(move } B_{nn}L_n \text{ outside brackets)} \\
&= \underbrace{B_{nn}L_n}_{BL} \cup \underbrace{\left(\bigcup_{j=1}^{n-1} B_{nj} L_j\right) \cup C_n}_{C} && \text{(apply } BL \cup C \;\Rightarrow\; B^{*}C\text{)} \\
&= B_{nn}^{*}\left(\left(\bigcup_{j=1}^{n-1} B_{nj} L_j\right) \cup C_n\right) && \text{(apply } b(a \cup c) = ba \cup bc\text{)} \\
&= B_{nn}^{*}\left(\bigcup_{j=1}^{n-1} B_{nj} L_j\right) \cup B_{nn}^{*} C_n && \text{(apply } b(a \cup c) = ba \cup bc\text{)} \\
&= \left(\bigcup_{j=1}^{n-1} B_{nn}^{*} B_{nj} L_j\right) \cup B_{nn}^{*} C_n
\end{aligned}
$$

Solution to a System of Regular Expressions

consider each equation L_i where $1 \leq i \leq n-1$:

$$\begin{aligned} L_i &= \left(\bigcup_{j=1}^{n} B_{ij} L_j\right) \cup C_i && \text{(move } B_{in} L_n \text{ outside brackets)} \\ &= B_{in} L_n \cup \left(\bigcup_{j=1}^{n-1} B_{ij} L_j\right) \cup C_i \end{aligned}$$

Solution to a System of Regular Expressions

consider each equation L_i where $1 \leq i \leq n-1$:

$$\begin{aligned}
L_i &= \Big(\bigcup_{j=1}^{n} B_{ij} L_j\Big) \cup C_i && \text{(move } B_{in} L_n \text{ outside brackets)} \\
&= B_{in} L_n \cup \Big(\bigcup_{j=1}^{n-1} B_{ij} L_j\Big) \cup C_i && \text{(substitute } L_n \text{ into equation)} \\
&= B_{in}\bigg(\Big(\bigcup_{j=1}^{n-1} B_{nn}^{*} B_{nj} L_j\Big) \cup B_{nn}^{*} C_n\bigg) \cup \Big(\bigcup_{j=1}^{n-1} B_{nj} L_j\Big) \cup C_i
\end{aligned}$$

Solution to a System of Regular Expressions

consider each equation L_i where $1 \leq i \leq n-1$:

$$
\begin{aligned}
L_i &= \Big(\bigcup_{j=1}^{n} B_{ij}L_j\Big) \cup C_i && \text{(move } B_{in}L_n \text{ outside brackets)}\\
&= B_{in}L_n \cup \Big(\bigcup_{j=1}^{n-1} B_{ij}L_j\Big) \cup C_i && \text{(substitute } L_n \text{ into equation)}\\
&= B_{in}\bigg(\Big(\bigcup_{j=1}^{n-1} B_{nn}^*B_{nj}L_j\Big) \cup B_{nn}^*C_n\bigg) \cup \Big(\bigcup_{j=1}^{n-1} B_{nj}L_j\Big) \cup C_i && \text{(apply } b(a \cup c) = ba \cup bc\text{)}\\
&= \Big(\bigcup_{j=1}^{n-1} B_{in}B_{nn}^*B_{nj}L_j\Big) \cup B_{in}B_{nn}^*C_n \cup \Big(\bigcup_{j=1}^{n-1} B_{nj}L_j\Big) \cup C_i
\end{aligned}
$$

Solution to a System of Regular Expressions

consider each equation L_i where $1 \le i \le n-1$:

$$
\begin{aligned}
L_i &= \left(\bigcup_{j=1}^{n} B_{ij} L_j\right) \cup C_i && \text{(move } B_{in} L_n \text{ outside brackets)} \\
&= B_{in} L_n \cup \left(\bigcup_{j=1}^{n-1} B_{ij} L_j\right) \cup C_i && \text{(substitute } L_n \text{ into equation)} \\
&= B_{in}\left(\left(\bigcup_{j=1}^{n-1} B_{nn}^{*} B_{nj} L_j\right) \cup B_{nn}^{*} C_n\right) \cup \left(\bigcup_{j=1}^{n-1} B_{nj} L_j\right) \cup C_i && \text{(apply } b(a \cup c) = ba \cup bc\text{)} \\
&= \left(\bigcup_{j=1}^{n-1} B_{in} B_{nn}^{*} B_{nj} L_j\right) \cup B_{in} B_{nn}^{*} C_n \cup \left(\bigcup_{j=1}^{n-1} B_{nj} L_j\right) \cup C_i && \text{(combine the two unions } \textstyle\bigcup_{j=1}^{n-1}\text{)} \\
&= \left(\bigcup_{j=1}^{n-1} \boxed{\underbrace{(B_{in} B_{nn}^{*} B_{nj} \cup B_{nj})}_{B'_{ij}}} L_j\right) \cup \boxed{\underbrace{B_{in} B_{nn}^{*} C_n \cup C_i}_{C'_i}}
\end{aligned}
$$

Solution to a System of Regular Expressions

consider each equation L_i where $1 \leq i \leq n-1$:

$$
\begin{aligned}
L_i &= \Big(\bigcup_{j=1}^{n} B_{ij}L_j\Big) \cup C_i && \text{(move } B_{in}L_n \text{ outside brackets)}\\
&= B_{in}L_n \cup \Big(\bigcup_{j=1}^{n-1} B_{ij}L_j\Big) \cup C_i && \text{(substitute } L_n \text{ into equation)}\\
&= B_{in}\Big(\Big(\bigcup_{j=1}^{n-1} B_{nn}^{*}B_{nj}L_j\Big) \cup B_{nn}^{*}C_n\Big) \cup \Big(\bigcup_{j=1}^{n-1} B_{nj}L_j\Big) \cup C_i && \text{(apply } b(a \cup c) = ba \cup bc\text{)}\\
&= \Big(\bigcup_{j=1}^{n-1} B_{in}B_{nn}^{*}B_{nj}L_j\Big) \cup B_{in}B_{nn}^{*}C_n \cup \Big(\bigcup_{j=1}^{n-1} B_{nj}L_j\Big) \cup C_i && \text{(combine the two unions } \bigcup_{j=1}^{n-1}\text{)}\\
&= \Big(\bigcup_{j=1}^{n-1} \boxed{\underbrace{(B_{in}B_{nn}^{*}B_{nj} \cup B_{nj})}_{B'_{ij}}}\, L_j\Big) \cup \boxed{\underbrace{B_{in}B_{nn}^{*}C_n \cup C_i}_{C'_i}}
\end{aligned}
$$

- now each equation is of the form $L_i = (\bigcup_{j=1}^{n-1} B_{ij}L_j) \cup C_i$ where $1 \leq i \leq n-1$
- L_n does not occur any more in the equations
- it follows from the hypothesis that L_1 can be written as a regular expression over B_{ij}'s and C_i's. □

Regular Expressions

The following theorem holds:

Theorem (1)

Let L be a language, then:

$$L \text{ is regular} \\ \Leftrightarrow \\ \text{there exists a regular expression } R \text{ that describes } L$$

$\Longleftarrow$:

Theorem (1A)

Every regular expression R describes a language $L(M)$ where M is a finite automaton.

$\Longrightarrow$:

Theorem (1B)

For every finite automaton M, the language $L(M)$ can be described by a regular expression R.

Learned about Regular Languages

- can be accepted by deterministic and non-deterministic finite automata
- can be described by regular expressions
- are closed under union, intersection, complement, concatenation, star
- techniques for proving that languages are regular exist

Learned about Regular Languages

- can be accepted by deterministic and non-deterministic finite automata
- can be described by regular expressions
- are closed under union, intersection, complement, concatenation, star
- techniques for proving that languages are regular exist

Further facts:

- given a regular language, the amount of **memory** that is needed to determine whether some string is in the language is **finite** and independent of the length of the string
- if a regular language consists infinitely many strings, then the language contains infinite subsets with a **repetitive structure**

Non-Regular Languages

Examples:

$$L_1 = \{0^n 1^n : n \geq 0\} = \{\epsilon, 01, 0011, 000111, \ldots\}$$

Infinite amount of memory may be required in order to remember how many 0s have occurred when the first 1 is read in.

Non-Regular Languages

Examples:

$$L_1 \;=\; \{0^n 1^n : n \geq 0\} \;=\; \{\epsilon, 01, 0011, 000111, \ldots\}$$

Infinite amount of memory may be required in order to remember how many 0s have occurred when the first 1 is read in.

$$L_2 \;=\; \{0^n : \text{n is a prime number}\} \;=\; \{00, 000, 00000, 0000000, \ldots\}$$

There are infinitely many prime numbers and prime numbers do not have a repetitive structure.

How to prove that a language is non-regular?

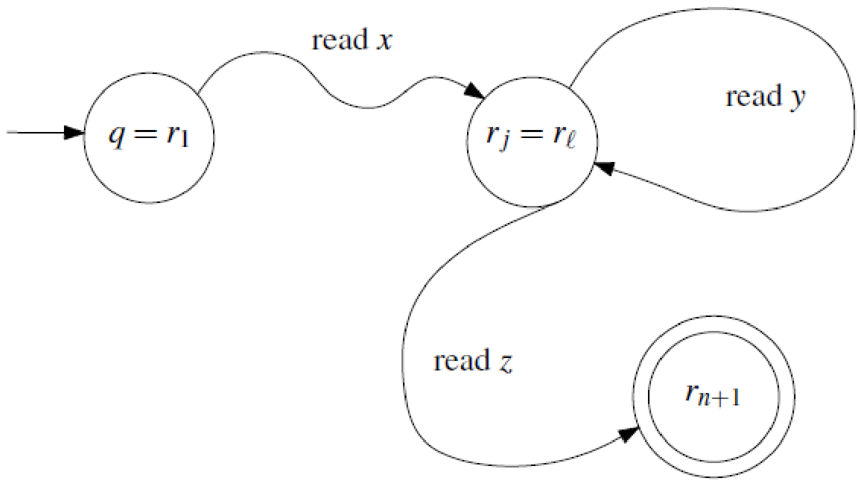
Pumping Lemma

The pumping lemma is a lemma that can be used for **proving or disproving** that a given language is regular:

Theorem (Pumping Lemma for Regular Languages)

Let A be a regular language. Then there exists a natural number $p \geq 1$, called the pumping length, such that the following holds: Every string $w \in A$, with length $|w| \geq p$, can be written as $w = xyz$, such that

1. $y \neq \epsilon$ *(non-empty middle part y)*
2. $|xy| \leq p$ *(finite prefix xy)*
3. *and* $xy^i z \in A$ *for all* $i \geq 0$ *(repeatable middle part)*



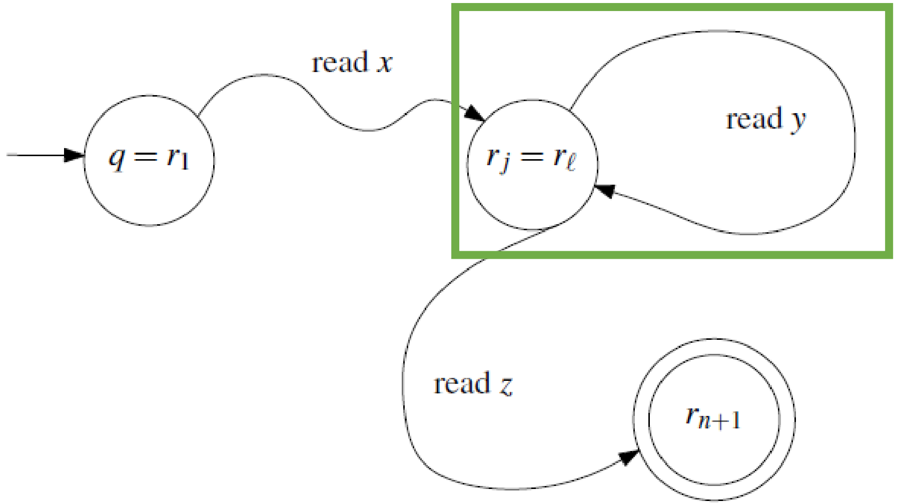
Pumping Lemma

The pumping lemma is a lemma that can be used for **proving or disproving** that a given language is regular:

Theorem (Pumping Lemma for Regular Languages)

Let A be a regular language. Then there exists a natural number $p \geq 1$, called the pumping length, such that the following holds: Every string $w \in A$, with length $|w| \geq p$, can be written as $w = xyz$, such that

1. $\boxed{y \neq \epsilon}$ *(non-empty middle part y)*
2. $|xy| \leq p$ *(finite prefix xy)*
3. *and $xy^i z \in A$ for all $i \geq 0$ (repeatable middle part)*



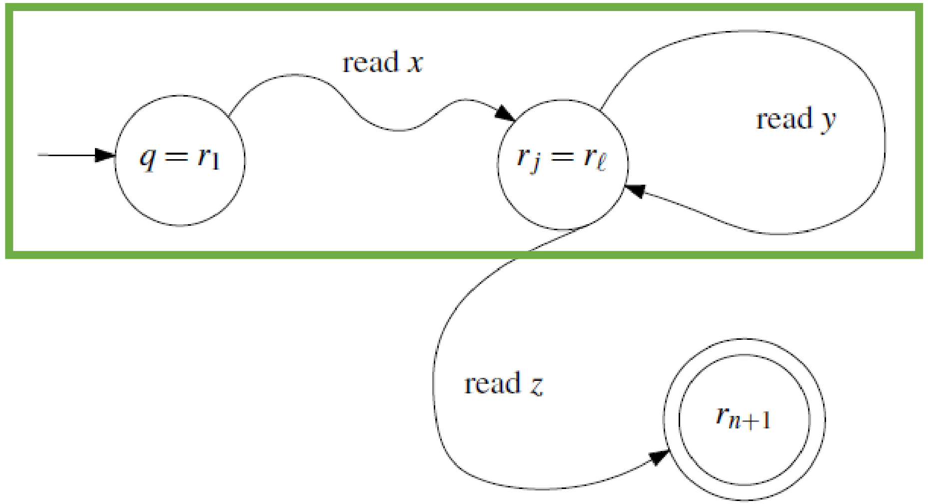
Pumping Lemma

The pumping lemma is a lemma that can be used for **proving or disproving** that a given language is regular:

Theorem (Pumping Lemma for Regular Languages)

Let A be a regular language. Then there exists a natural number $p \geq 1$, called the pumping length, such that the following holds: Every string $w \in A$, with length $|w| \geq p$, can be written as $w = xyz$, such that

1. $y \neq \epsilon$ *(non-empty middle part y)*
2. $\boxed{|xy| \leq p}$ *(finite prefix xy)*
3. *and* $xy^i z \in A$ *for all* $i \geq 0$ *(repeatable middle part)*



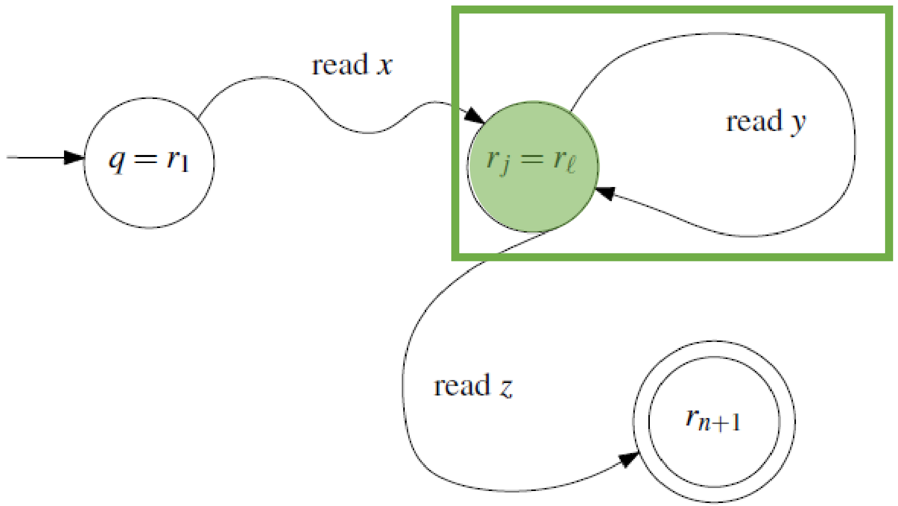
Pumping Lemma

The pumping lemma is a lemma that can be used for **proving or disproving** that a given language is regular:

Theorem (Pumping Lemma for Regular Languages)

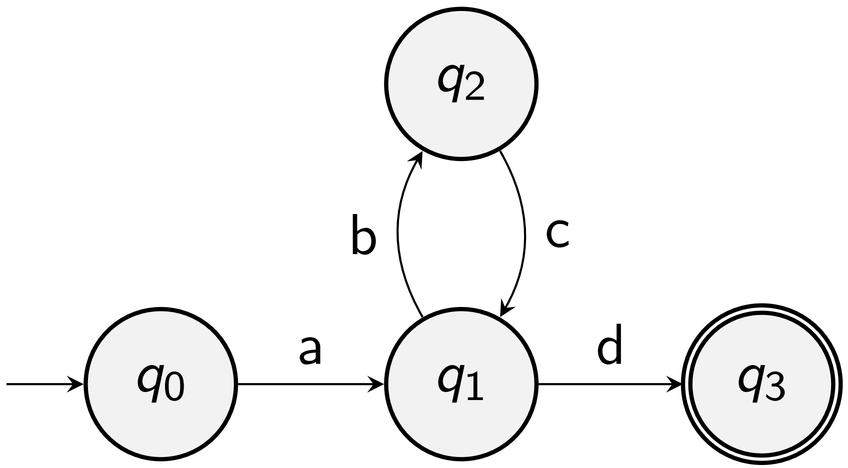
Let A be a regular language. Then there exists a natural number $p \geq 1$, called the pumping length, such that the following holds: Every string $w \in A$, with length $|w| \geq p$, can be written as $w = xyz$, such that

1. $y \neq \epsilon$ *(non-empty middle part y)*
2. $|xy| \leq p$ *(finite prefix xy)*
3. $\boxed{\text{and } xy^i z \in A \text{ for all } i \geq 0}$ *(repeatable middle part)*



Correctness of the Pumping Lemma - Exemplified

- $L = a(bc)^*d$ is regular
- We can construct finite automaton M that accepts L:



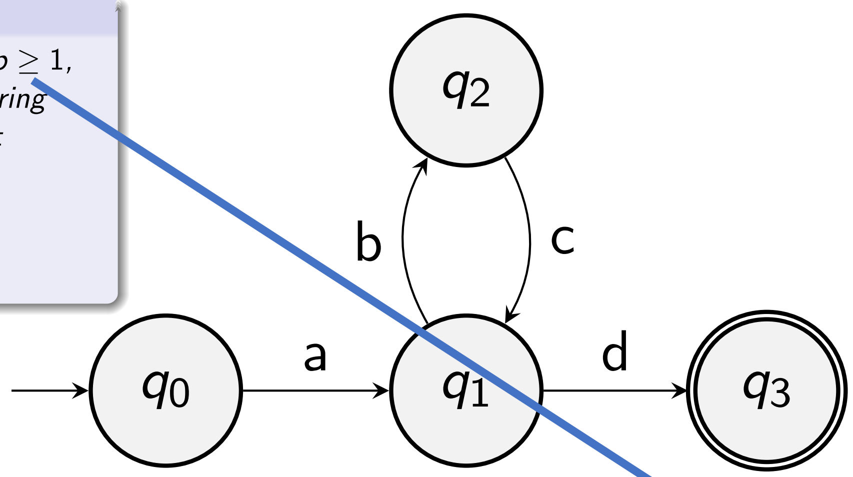
Correctness of the Pumping Lemma - Exemplified

- $L = a(bc)^*d$ is regular
- We can construct finite automaton M that accepts L:

Theorem (Pumping Lemma for Regular Languages)

Let A be a regular language. Then there exists a natural number $p \geq 1$, called the pumping length, such that the following holds: Every string $w \in A$, with length $|w| \geq p$, can be written as $w = xyz$, such that

1. $y \neq \epsilon$ *(non-empty middle part y)*
2. $|xy| \leq p$ *(finite prefix xy)*
3. *and $xy^iz \in A$ for all $i \geq 0$ (repeatable middle part)*



- M has four states. Choose pumping length $p = 4$

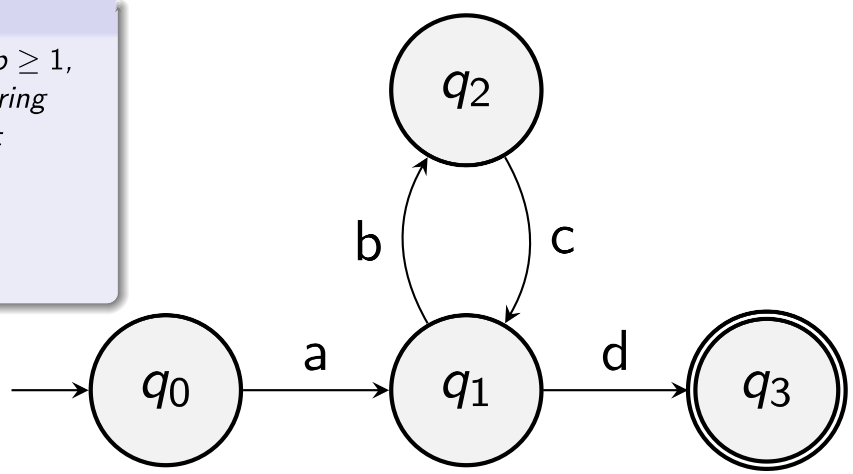
Correctness of the Pumping Lemma - Exemplified

- $L = a(bc)^*d$ is regular
- We can construct finite automaton M that accepts L:

Theorem (Pumping Lemma for Regular Languages)

Let A be a regular language. Then there exists a natural number $p \geq 1$, called the pumping length, such that the following holds: Every string $w \in A$, with length $|w| \geq p$, can be written as $w = xyz$, such that

1. $y \neq \epsilon$ *(non-empty middle part y)*
2. $|xy| \leq p$ *(finite prefix xy)*
3. *and* $xy^iz \in A$ *for all* $i \geq 0$ *(repeatable middle part)*



- M has four states. Choose pumping length $p = 4$
- $w = abcd$ is an accepted string of length 4

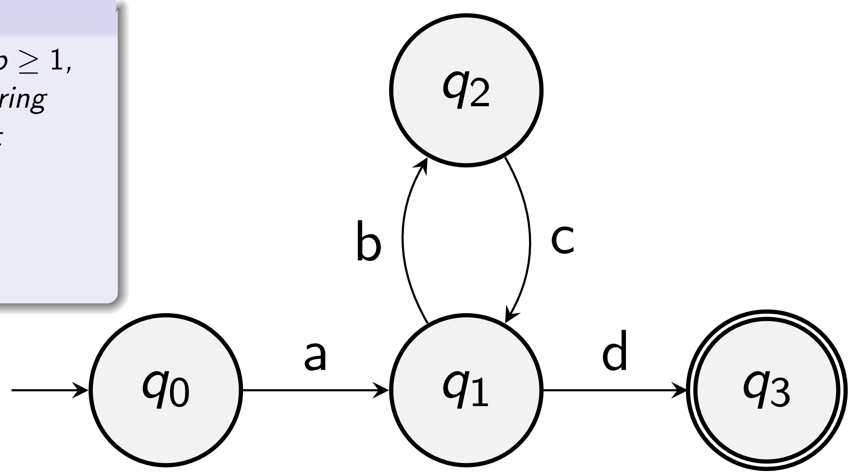
Correctness of the Pumping Lemma - Exemplified

- $L = a(bc)^*d$ is regular
- We can construct finite automaton M that accepts L:

Theorem (Pumping Lemma for Regular Languages)

Let A be a regular language. Then there exists a natural number $p \geq 1$, called the pumping length, such that the following holds: Every string $w \in A$, with length $|w| \geq p$, can be written as $w = xyz$, such that

1. $y \neq \epsilon$ *(non-empty middle part y)*
2. $|xy| \leq p$ *(finite prefix xy)*
3. *and* $xy^iz \in A$ *for all* $i \geq 0$ *(repeatable middle part)*



- M has four states. Choose pumping length $p = 4$
- $w = abcd$ is an accepted string of length 4
- run over w must have at least one repeated state, which is q_1 here
- the sub-string bc takes M from q_1 to q_1 again.

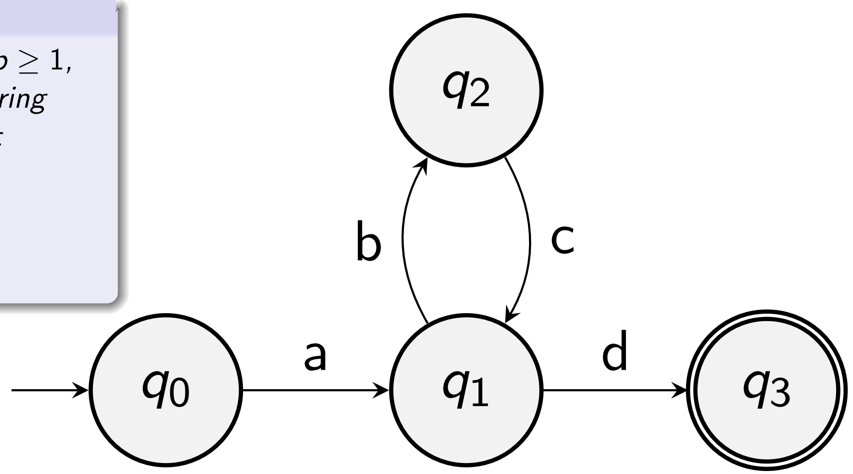
Correctness of the Pumping Lemma - Exemplified

- $L = a(bc)^*d$ is regular
- We can construct finite automaton M that accepts L:

Theorem (Pumping Lemma for Regular Languages)

Let A be a regular language. Then there exists a natural number $p \geq 1$, called the pumping length, such that the following holds: Every string $w \in A$, with length $|w| \geq p$, can be written as $w = xyz$, such that

1. $y \neq \epsilon$ *(non-empty middle part y)*
2. $|xy| \leq p$ *(finite prefix xy)*
3. *and $xy^iz \in A$ for all $i \geq 0$* *(repeatable middle part)*



- M has four states. Choose pumping length $p = 4$
- $w = abcd$ is an accepted string of length 4
- run over w must have at least one repeated state, which is q_1 here
- the sub-string bc takes M from q_1 to q_1 again.
- choose $y = bc$. Consequently $x = a$ and $z = d$

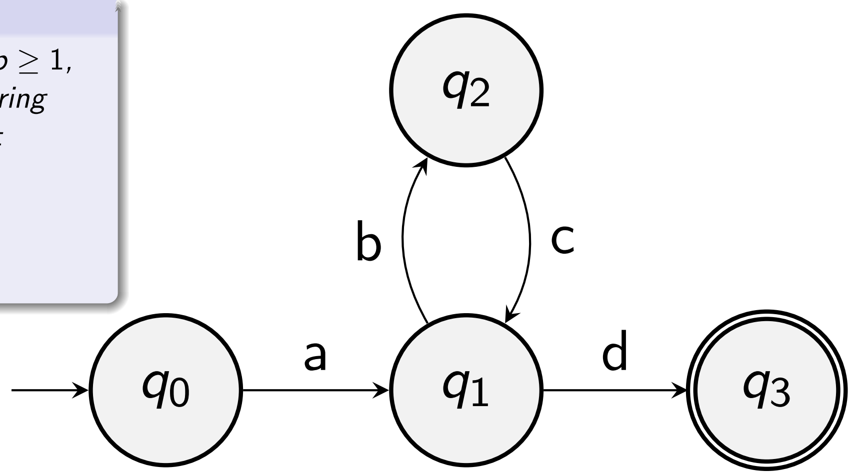
Correctness of the Pumping Lemma - Exemplified

- $L = a(bc)^*d$ is regular
- We can construct finite automaton M that accepts L:

Theorem (Pumping Lemma for Regular Languages)

Let A be a regular language. Then there exists a natural number $p \geq 1$, called the pumping length, such that the following holds: Every string $w \in A$, with length $|w| \geq p$, can be written as $w = xyz$, such that

1. $y \neq \epsilon$ *(non-empty middle part y)*
2. $|xy| \leq p$ *(finite prefix xy)*
3. *and $xy^iz \in A$ for all $i \geq 0$ (repeatable middle part)*



- M has four states. Choose pumping length $p = 4$
- $w = abcd$ is an accepted string of length 4
- run over w must have at least one repeated state, which is q_1 here
- the sub-string bc takes M from q_1 to q_1 again.
- choose $y = bc$. Consequently $x = a$ and $z = d$
- M accepts $ad, abcd, abcbcd, \ldots$
- all conditions of the pumping lemma are satisfied

$\boxed{|xy| = |abc| = 3}$

How to Prove that a Language in Non-Regular?

- Assume that a given language A is regular
- Show that the properties of the pumping lemma would lead to a **contradiction**
- It can be followed that A is not regular

