Regular Expressions and the Pumping Lemma

Dr. Chuck Rocca
roccac@wcsu.edu
http://sites.wcsu.edu/roccac

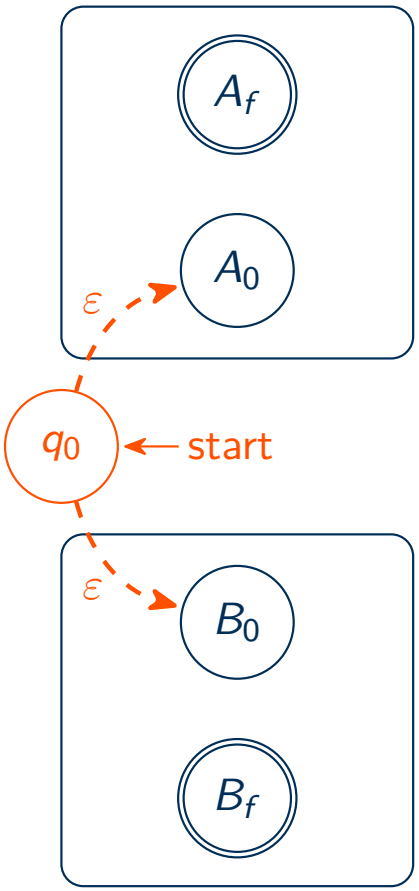
WESTERN CONNECTICUT STATE UNIVERSITY
MACRICOSTAS SCHOOL OF ARTS & SCIENCES

Table of Contents

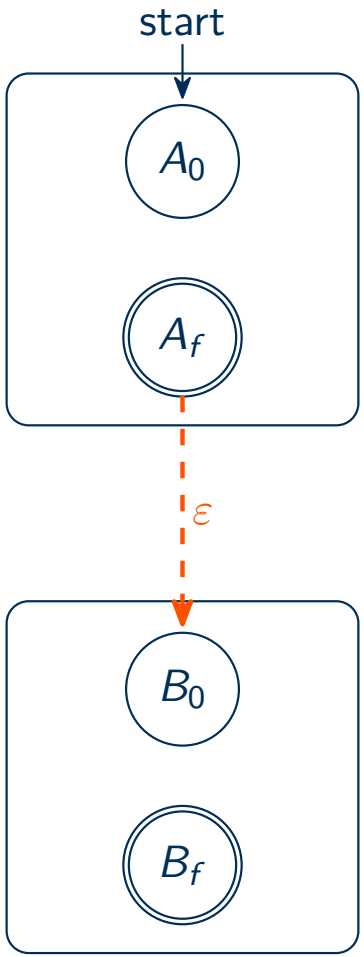
Table of Contents

Review Combining NFAs

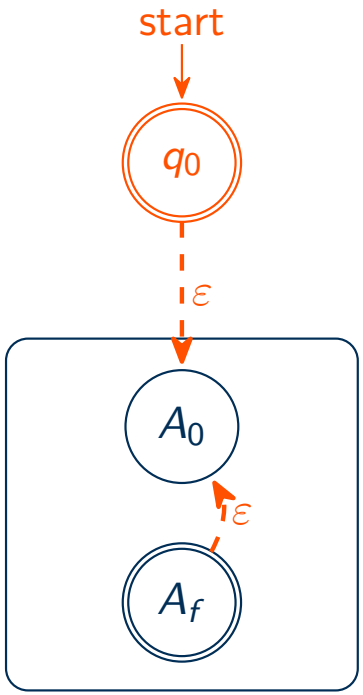
$A \cup B$



$A \circ B$



A^*



Basics of Regular Expressions

Definition (Regular Expression)

Given an alphabet Σ and regular languages A and B a *regular expression* is any combination of the following:

- $a \in \Sigma$

Basics of Regular Expressions

Definition (Regular Expression)

Given an alphabet Σ and regular languages A and B a *regular expression* is any combination of the following:

- $a \in \Sigma$
- ε the empty string

Basics of Regular Expressions

Definition (Regular Expression)

Given an alphabet Σ and regular languages A and B a *regular expression* is any combination of the following:

- $a \in \Sigma$
- ε the empty string
- $\emptyset$ the language containing no strings at all

Basics of Regular Expressions

Definition (Regular Expression)

Given an alphabet Σ and regular languages A and B a *regular expression* is any combination of the following:

- $a \in \Sigma$
- ε the empty string
- $\emptyset$ the language containing no strings at all
- Unions: $A \cup B = A|B$

Basics of Regular Expressions

Definition (Regular Expression)

Given an alphabet Σ and regular languages A and B a *regular expression* is any combination of the following:

- $a \in \Sigma$
- ε the empty string
- $\emptyset$ the language containing no strings at all
- Unions: $A \cup B = A|B$
- Concatenation: $A \circ B = AB$

Basics of Regular Expressions

Definition (Regular Expression)

Given an alphabet Σ and regular languages A and B a *regular expression* is any combination of the following:

- $a \in \Sigma$
- ε the empty string
- $\emptyset$ the language containing no strings at all
- Unions: $A \cup B = A|B$
- Concatenation: $A \circ B = AB$
- Star: A^* (includes ε)

Basics of Regular Expressions

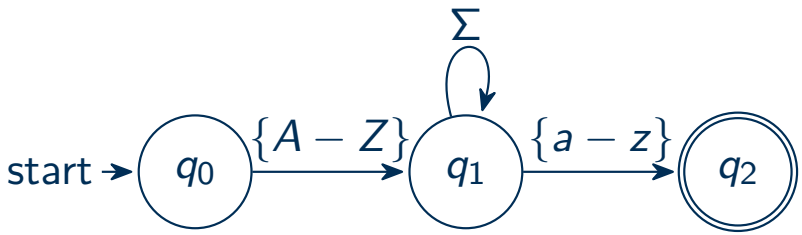
Definition (Regular Expression)

Given an alphabet Σ and regular languages A and B a *regular expression* is any combination of the following:

- $a \in \Sigma$
- ε the empty string
- $\emptyset$ the language containing no strings at all
- Unions: $A \cup B = A|B$
- Concatenation: $A \circ B = AB$
- Star: A^* (includes ε)
- Plus: $A^+ = A \circ A^* = AA^*$

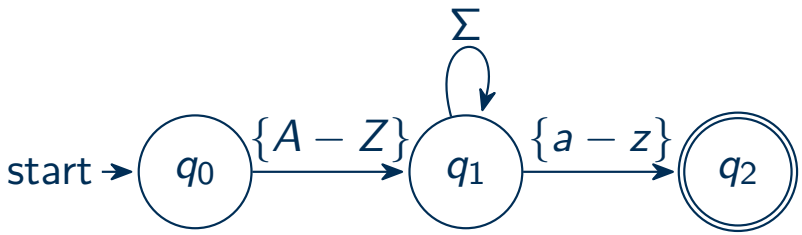
Automaton to Regex

With the alphabet $\Sigma = \{a - z, A - Z, 0 - 9, \$\}$ here are equivalent machines and regular expressions:



Automaton to Regex

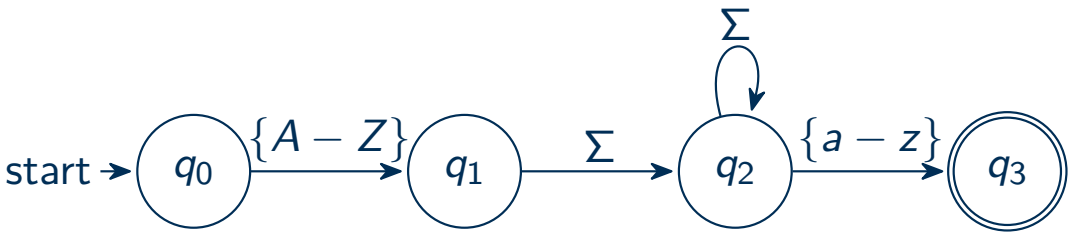
With the alphabet $\Sigma = \{a-z, A-Z, 0-9, \$\}$ here are equivalent machines and regular expressions:



- $\{A-Z\} \circ \Sigma^* \circ \{a-z\}$
- *Python* : $[A-Z][a-zA-Z0-9\backslash\$] * [a-z]$

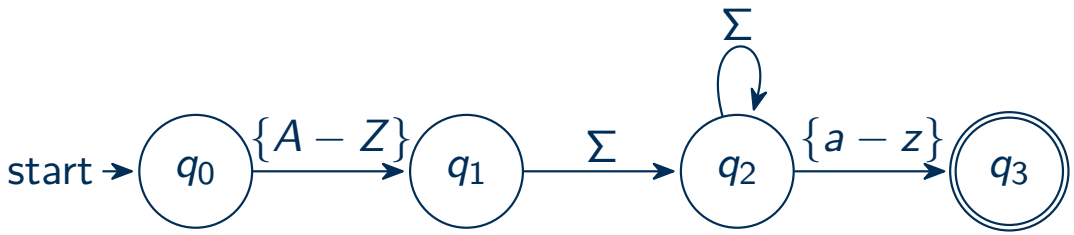
Automaton to Regex

With the alphabet $\Sigma = \{a - z, A - Z, 0 - 9, \$\}$ here are equivalent machines and regular expressions:



Automaton to Regex

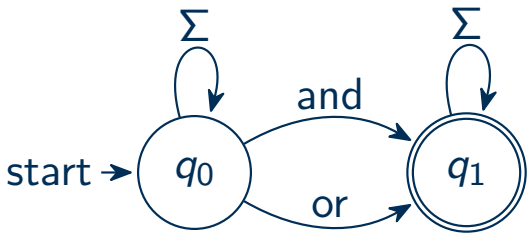
With the alphabet $\Sigma = \{a-z, A-Z, 0-9, \$\}$ here are equivalent machines and regular expressions:



- $\{A-Z\} \circ (\Sigma \circ \Sigma^*) \circ \{a-z\}$
- $\{A-Z\} \circ \Sigma^+ \circ \{a-z\}$
- *Python* : $[A-Z][a-zA-Z0-9\backslash\$]+[a-z]$

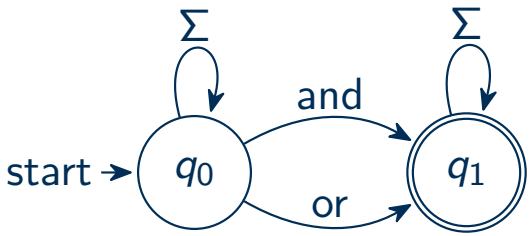
Automaton to Regex

With the alphabet $\Sigma = \{a - z, A - Z, 0 - 9, \$\}$ here are equivalent machines and regular expressions:



Automaton to Regex

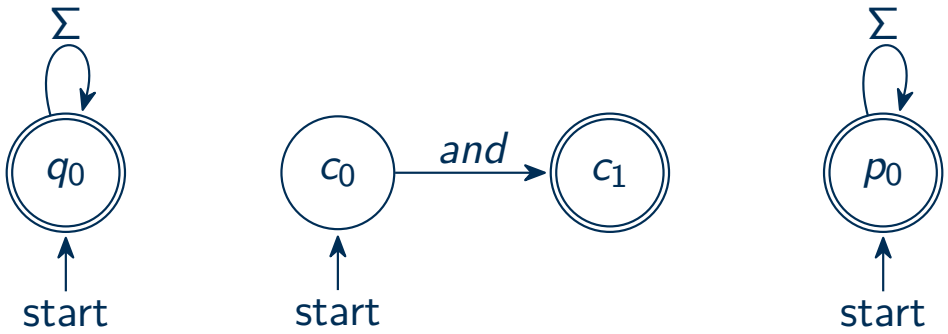
With the alphabet $\Sigma = \{a-z, A-Z, 0-9, \$\}$ here are equivalent machines and regular expressions:



- $\Sigma^* \circ (and \cup or) \circ \Sigma^*$
- $(\Sigma^* \circ (and) \circ \Sigma^*) \cup (\Sigma^* \circ (or) \circ \Sigma^*)$
- *Python* : $[a-zA-Z0-9\backslash\$] * (and|or)[a-zA-Z0-9\backslash\$]*$

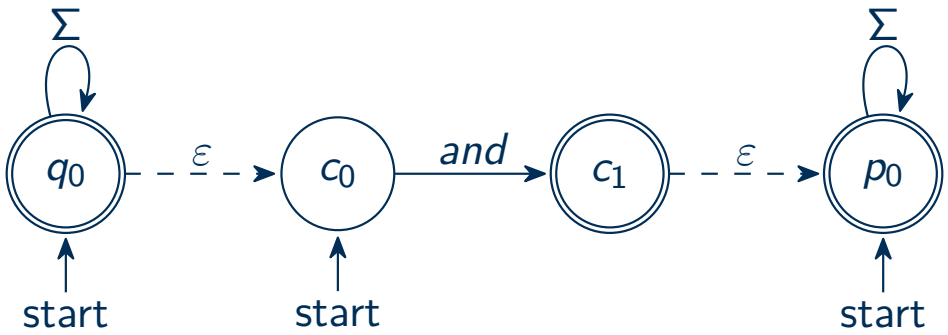
Regular Expressions to NFAs

$$(\Sigma^* \circ (and) \circ \Sigma^*) \cup (\Sigma^* \circ (or) \circ \Sigma^*)$$



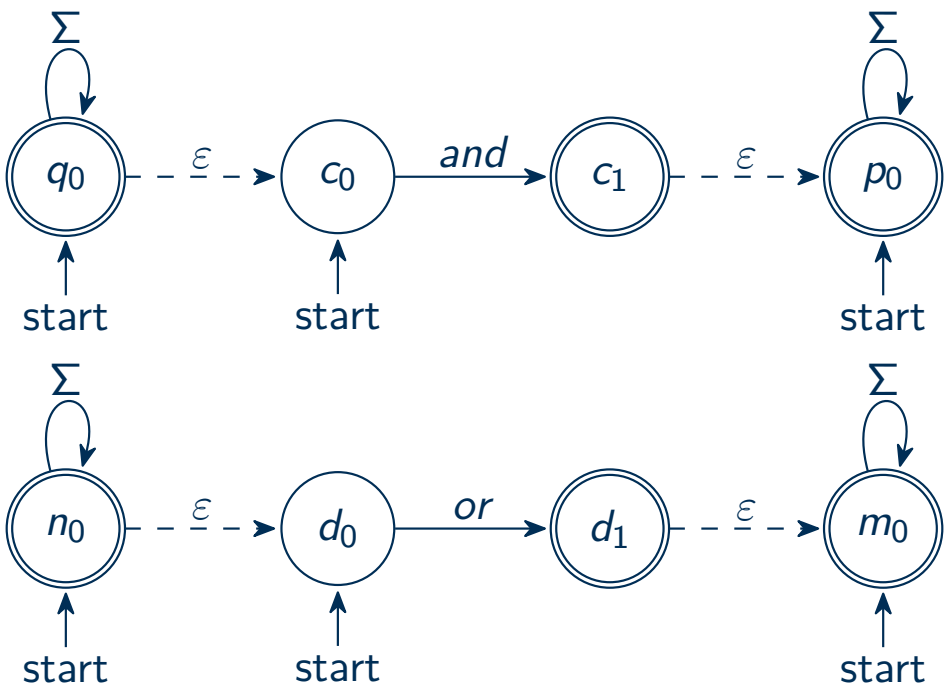
Regular Expressions to NFAs

$$(\Sigma^* \circ (and) \circ \Sigma^*) \cup (\Sigma^* \circ (or) \circ \Sigma^*)$$



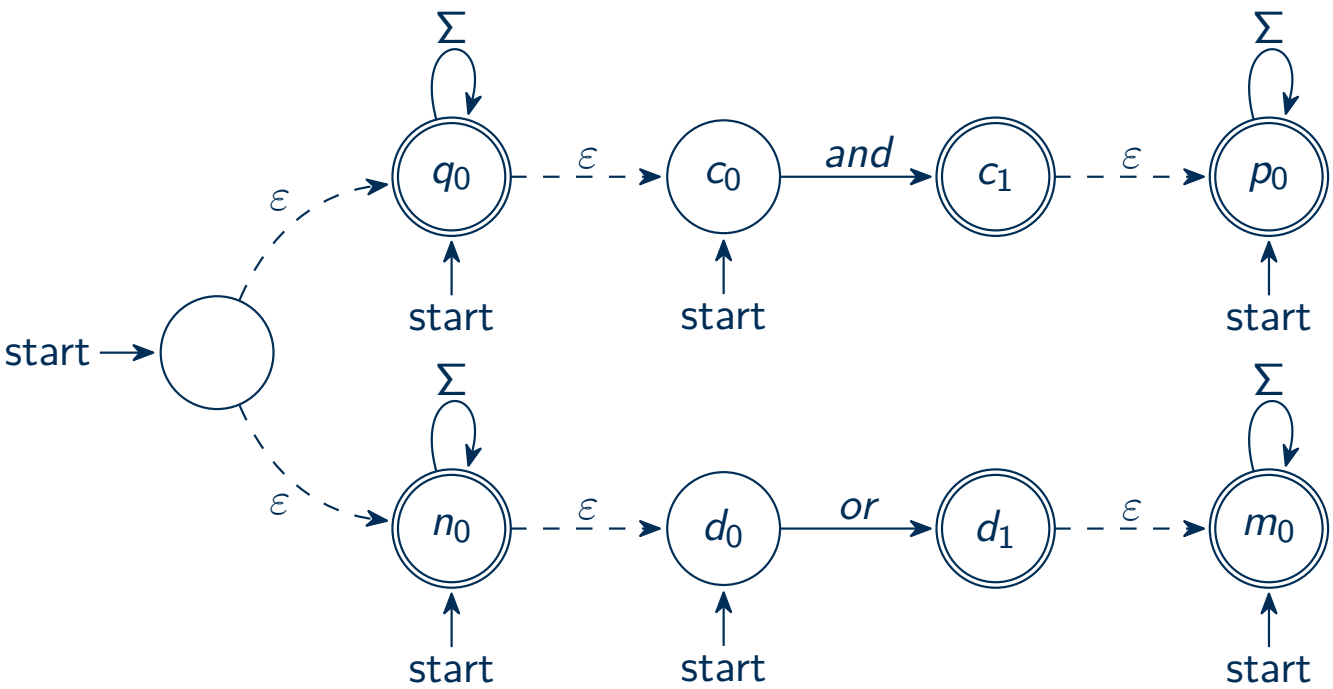
Regular Expressions to NFAs

$$(\Sigma^* \circ (and) \circ \Sigma^*) \cup (\Sigma^* \circ (or) \circ \Sigma^*)$$



Regular Expressions to NFAs

$$(\Sigma^* \circ (and) \circ \Sigma^*) \cup (\Sigma^* \circ (or) \circ \Sigma^*)$$



Identities

- Order of Operations: (), $*$, $\circ$, $\cup$
- Distribution: $(A \cup B) \circ C = A \circ C \cup B \circ C$
- Commutative: $A \cup B = B \cup A$
- "Additive" Identity: $A \cup \emptyset = A$ and $A \circ \emptyset = \emptyset$
- "Multiplicative" Identity: $A \circ \varepsilon = A$

Table of Contents

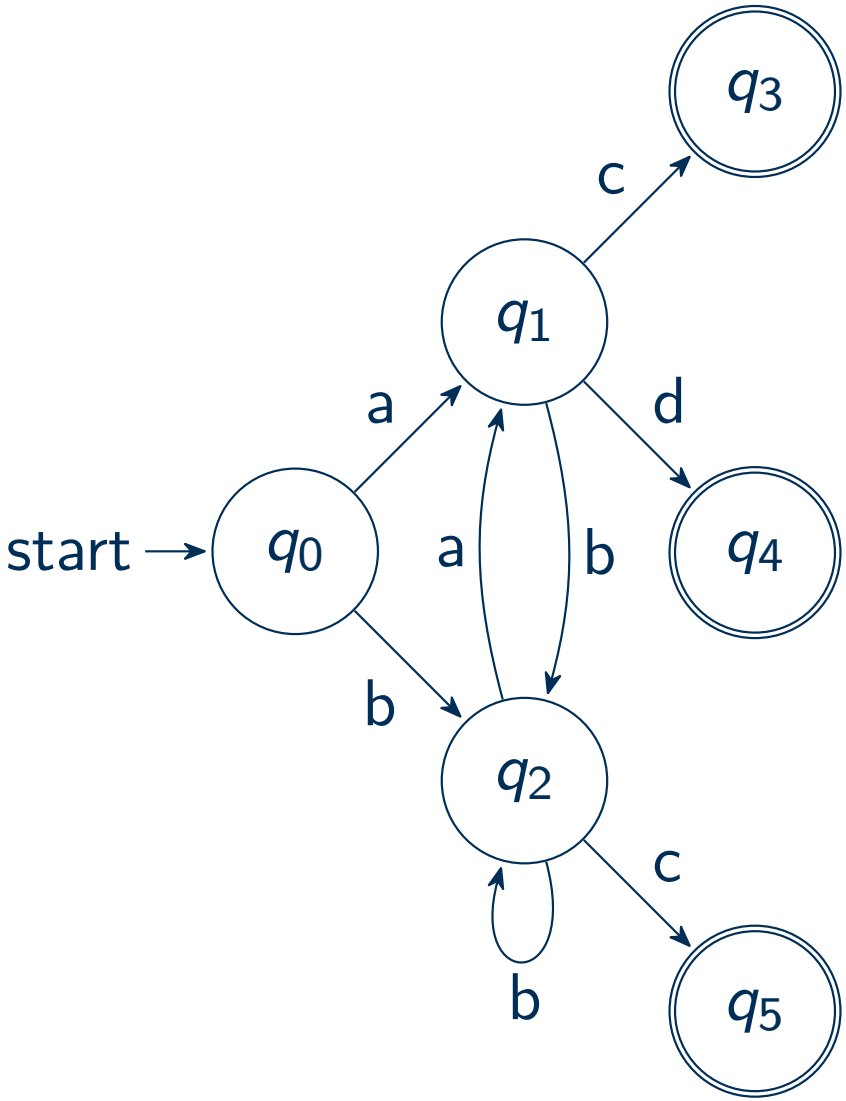
Generalized NFA

Definition (Generalized NFA)

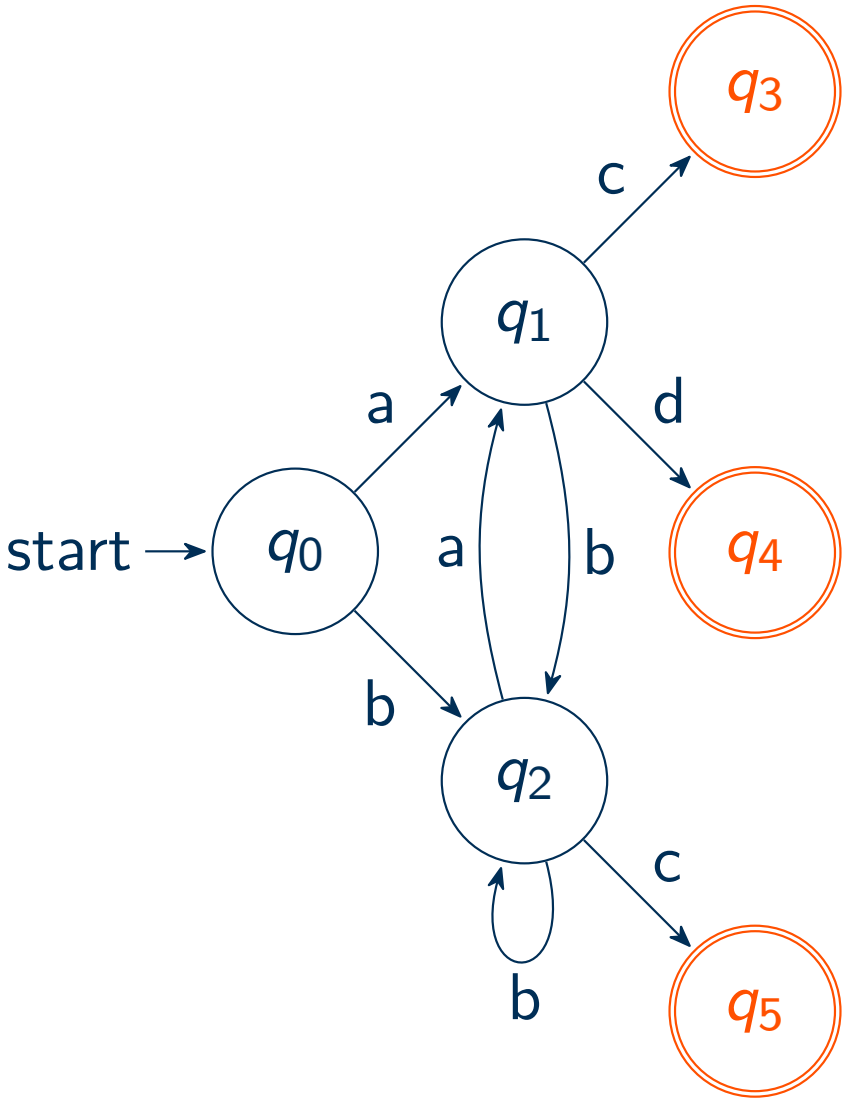
A *Generalized nondeterministic finite automaton* is a 5-tuple $(Q, \Sigma, \delta, q_{start}, q_{accept})$, where

- Q is a finite set of states
- Σ is the input alphabet
- $\mathcal{R}$ is the set of all regular expressions using Σ
- $\delta : (Q \setminus \{q_{accept}\}) \times (Q \setminus \{q_{start}\}) \longrightarrow \mathcal{R}$ is the transition function
- q_{start} is the start state
- q_{accept} is the unique accept state

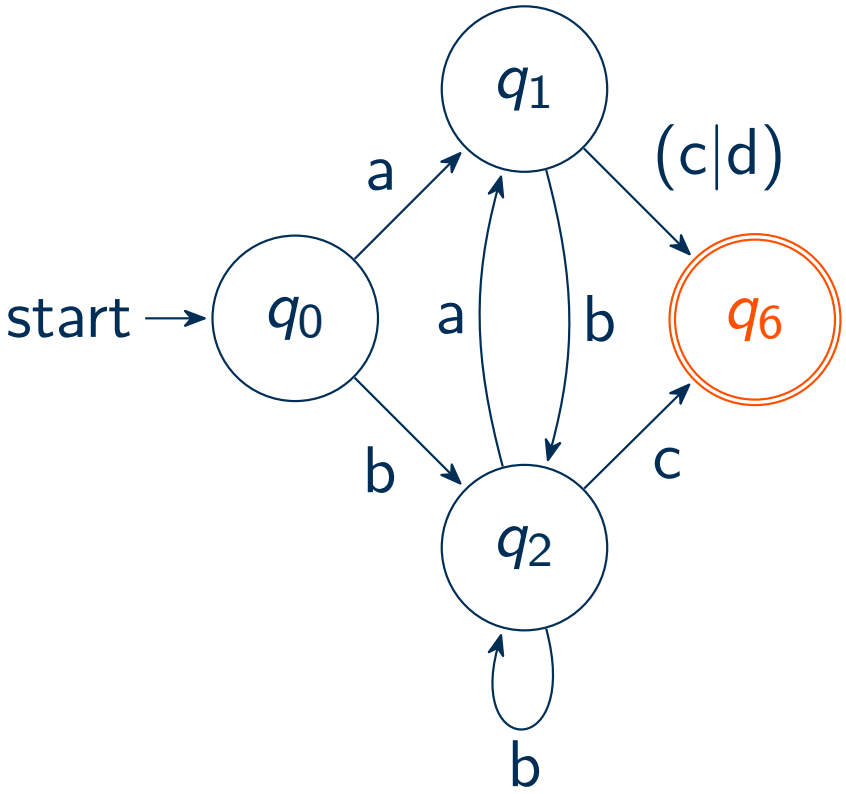
Converting to a GNFA and Reducing (NFA to Regex)



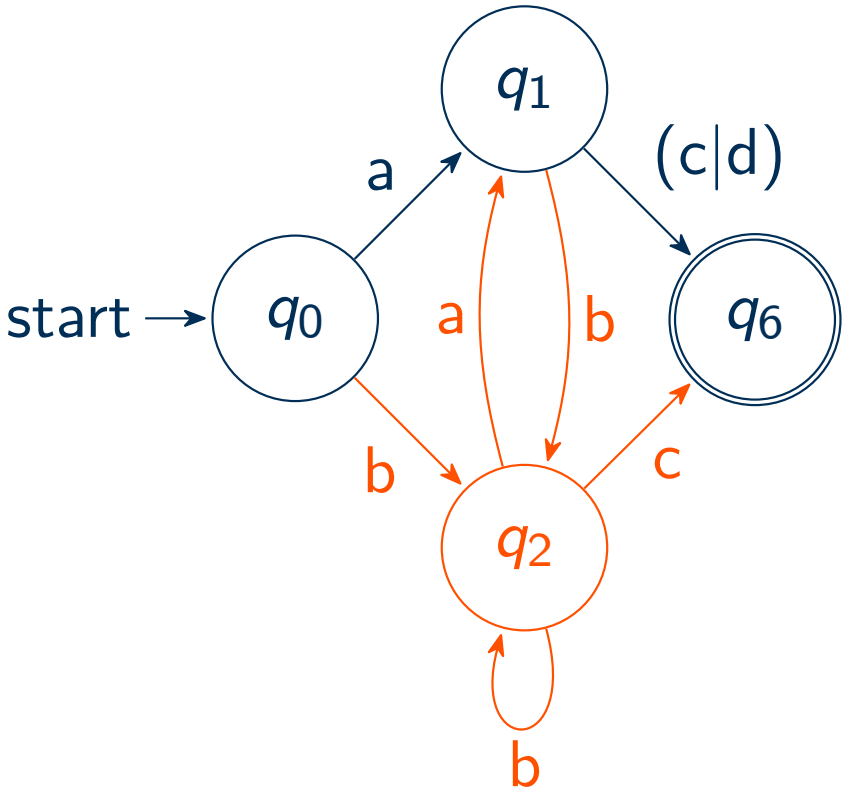
Converting to a GNFA and Reducing (NFA to Regex)



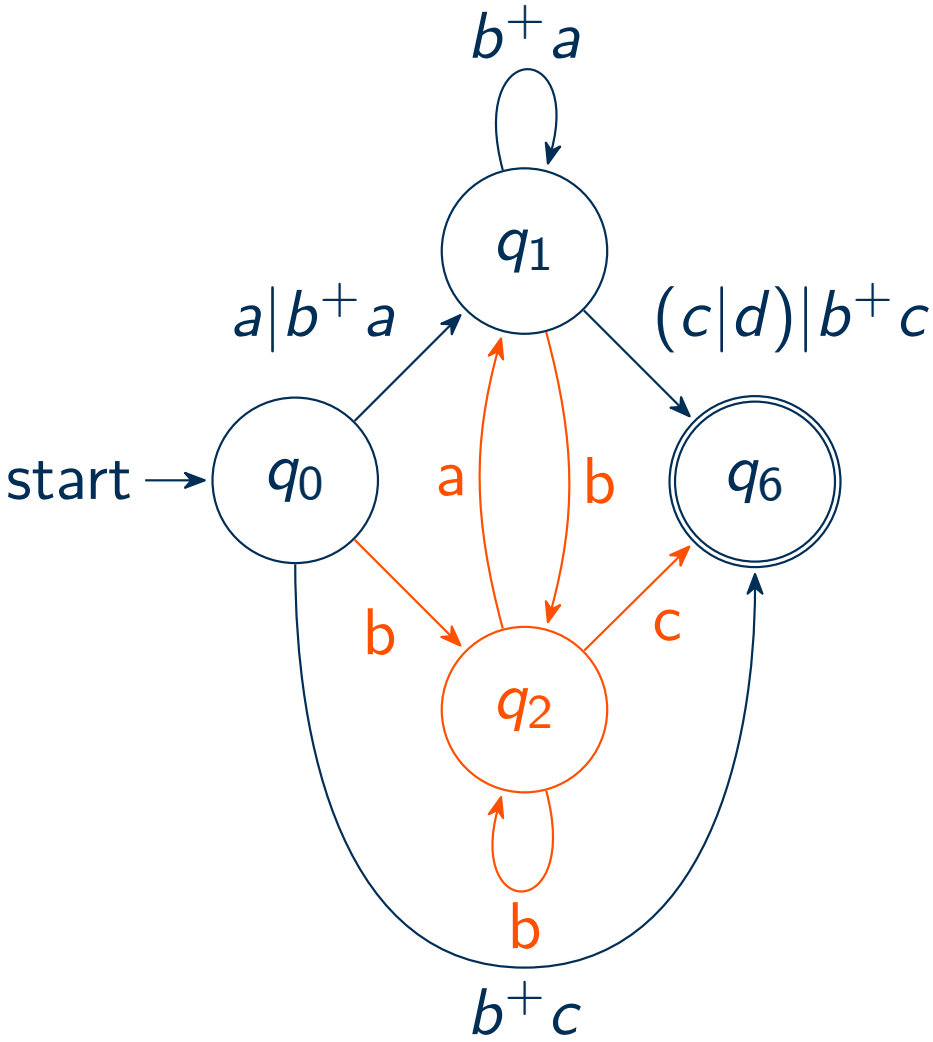
Converting to a GNFA and Reducing (NFA to Regex)



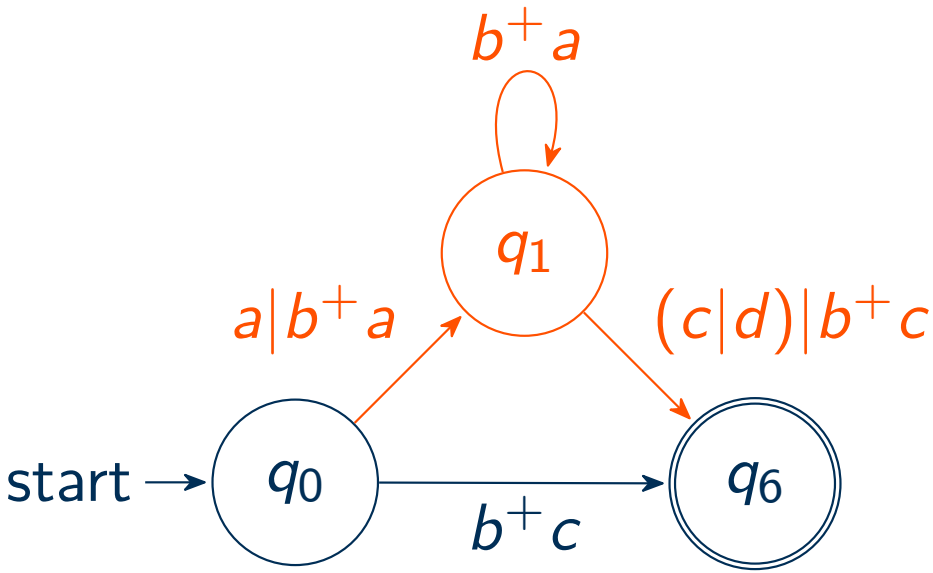
Converting to a GNFA and Reducing (NFA to Regex)



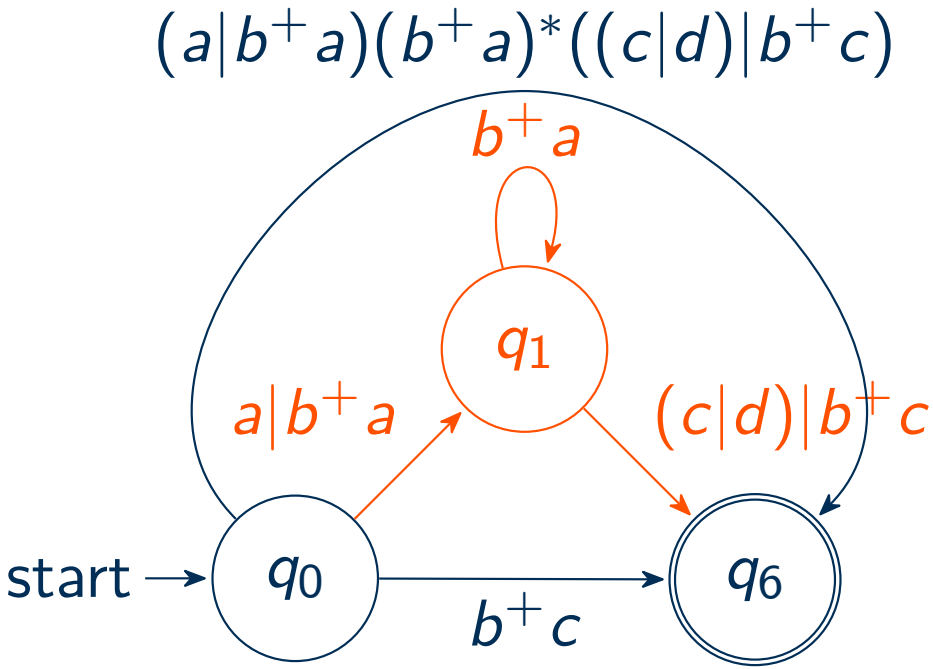
Converting to a GNFA and Reducing (NFA to Regex)



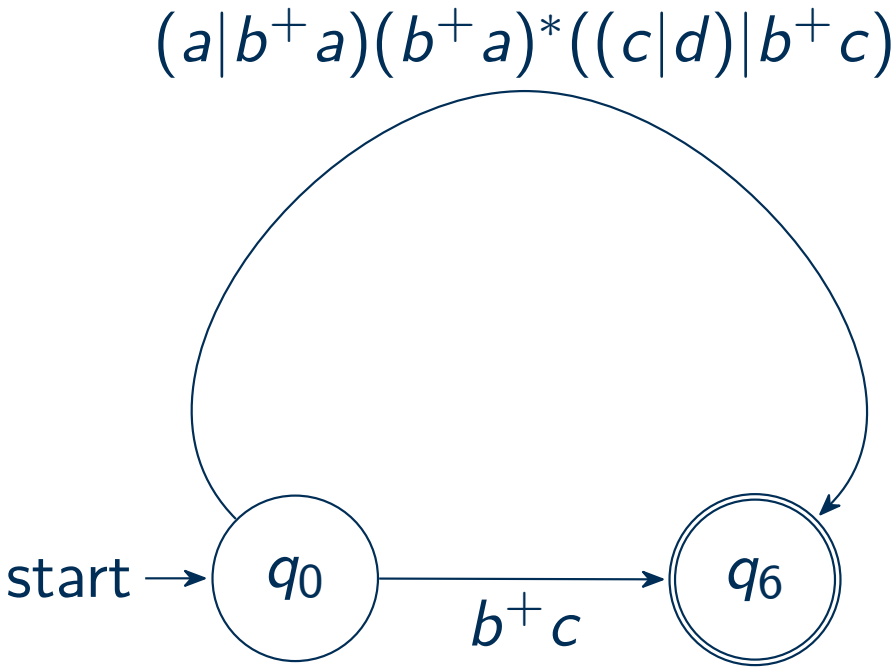
Converting to a GNFA and Reducing (NFA to Regex)



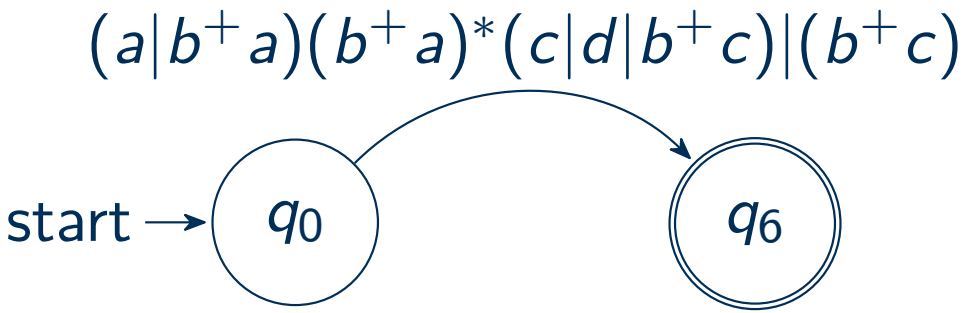
Converting to a GNFA and Reducing (NFA to Regex)



Converting to a GNFA and Reducing (NFA to Regex)



Converting to a GNFA and Reducing (NFA to Regex)



Regular Languages and Regular Expressions

Theorem

A language is regular if and only if some regular expression describes it.

Table of Contents

Statement

Theorem (Pumping Lemma)

If A is a regular language, then there is a number p (the pumping length) such that, if s is any string in A of length at least p, then s may be written in three pieces, $s = xyz$, satisfying the following conditions:

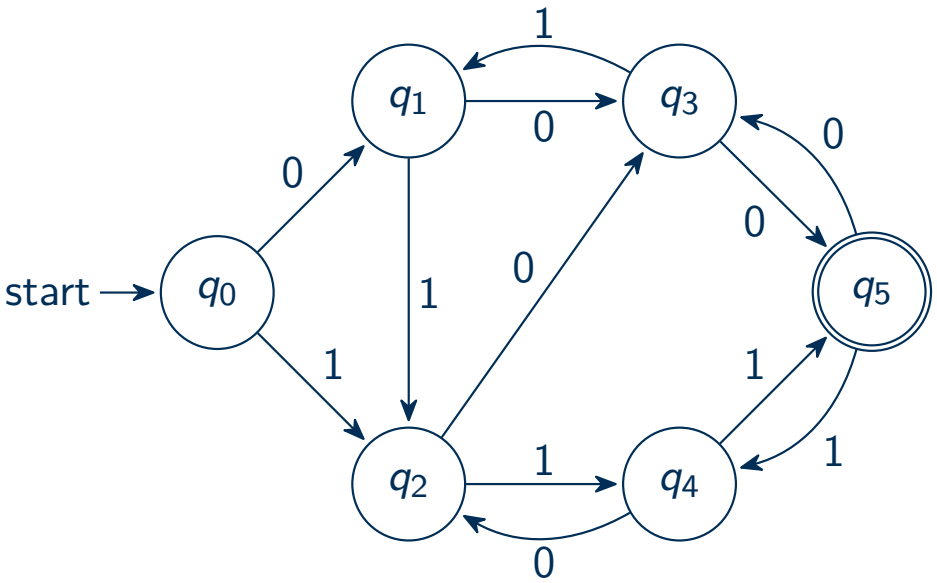
1. *for each $i \geq 0$, $xy^iz \in A$,*
2. *$|y| > 0$, and*
3. *$|xy| \leq p$.*

Pumping Example

Let $\Sigma = \{0, 1\}$ and consider

$$s = 01001001111$$

with $|s| = 11 > 6 = |Q|$.

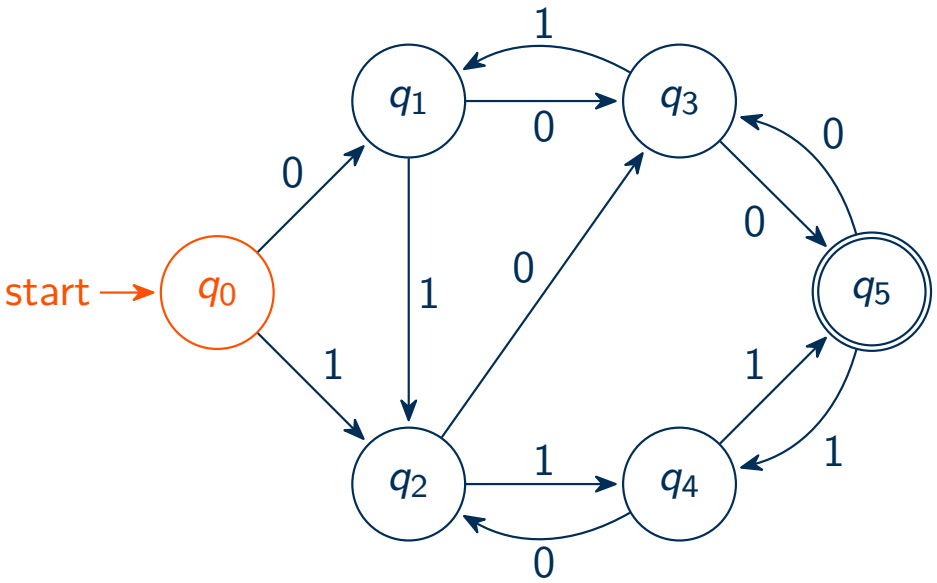


Pumping Example

Let $\Sigma = \{0, 1\}$ and consider

$$s = 01001001111$$

with $|s| = 11 > 6 = |Q|$.

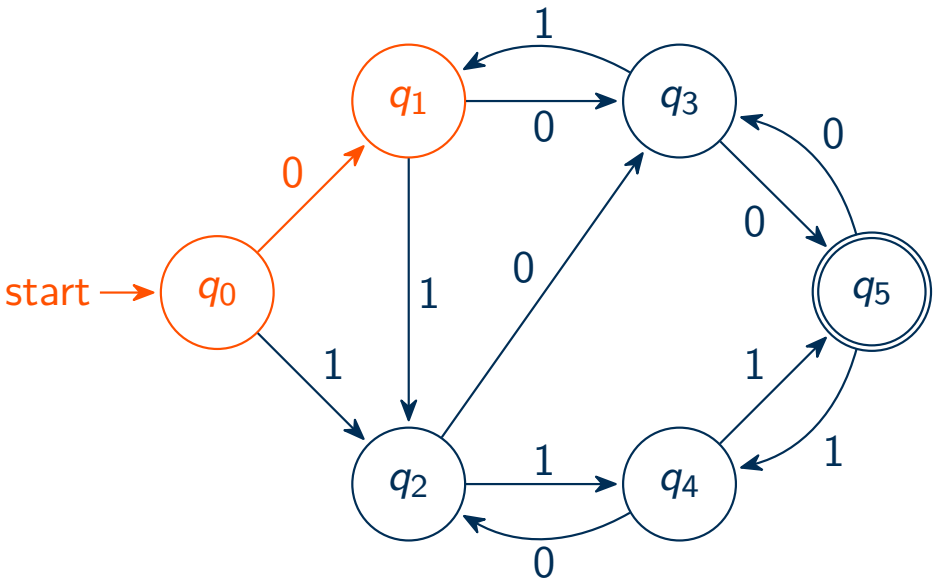


Pumping Example

Let $\Sigma = \{0, 1\}$ and consider

$$s = 01001001111$$

with $|s| = 11 > 6 = |Q|$.

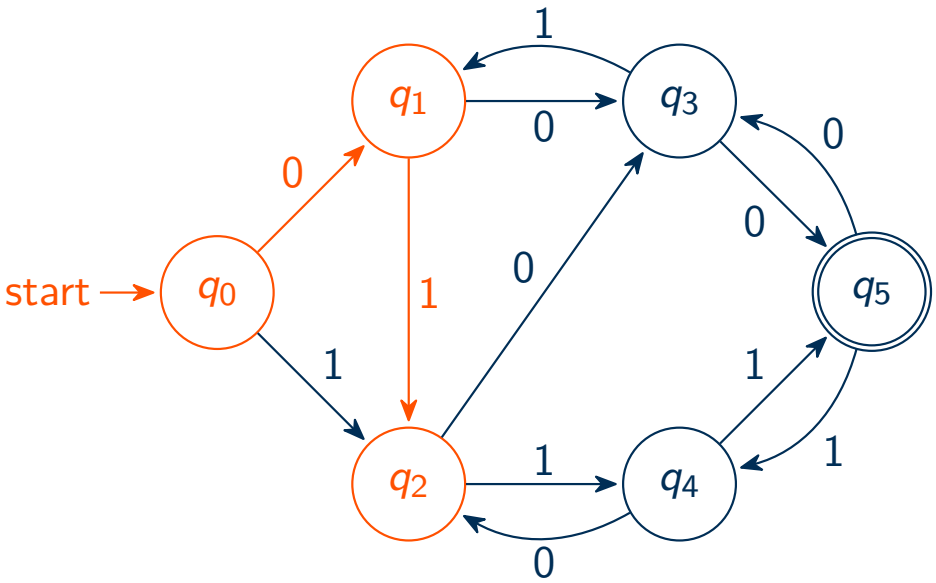


Pumping Example

Let $\Sigma = \{0, 1\}$ and consider

$$s = 01001001111$$

with $|s| = 11 > 6 = |Q|$.

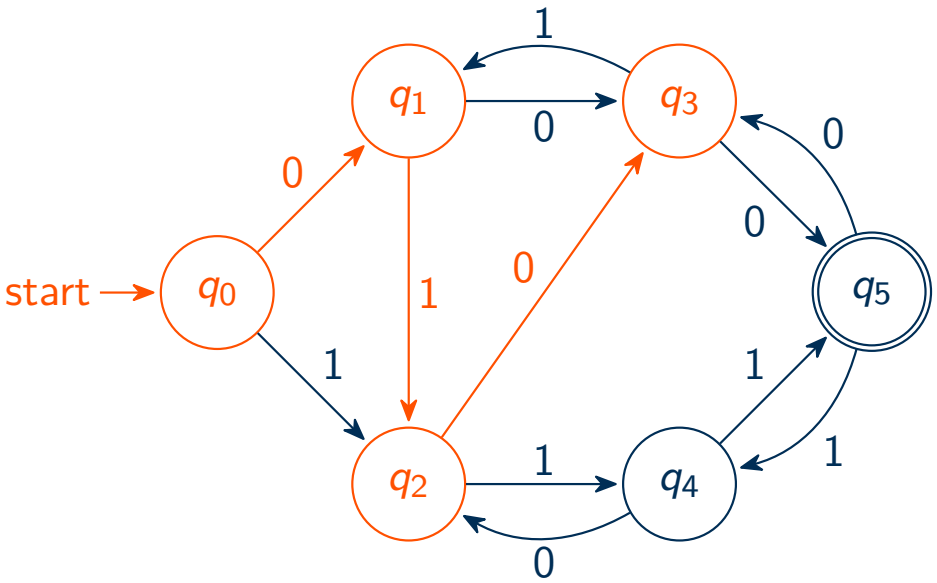


Pumping Example

Let $\Sigma = \{0, 1\}$ and consider

$$s = 01001001111$$

with $|s| = 11 > 6 = |Q|$.

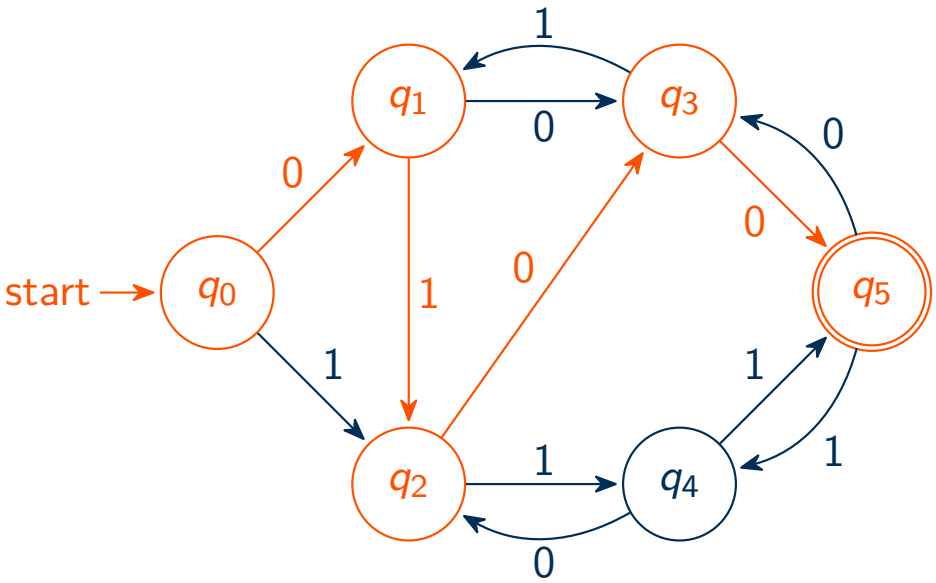


Pumping Example

Let $\Sigma = \{0, 1\}$ and consider

$$s = 01001001111$$

with $|s| = 11 > 6 = |Q|$.

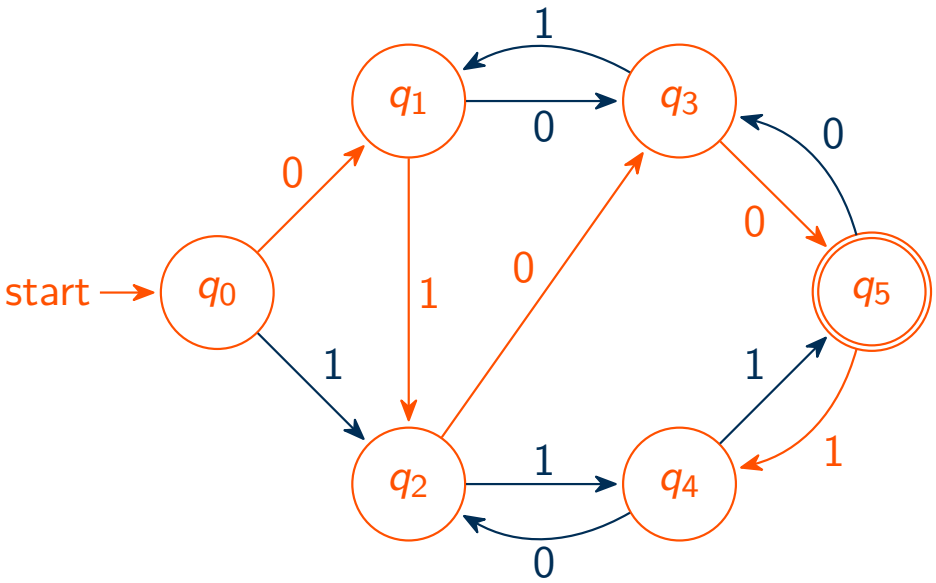


Pumping Example

Let $\Sigma = \{0, 1\}$ and consider

$$s = 01001001111$$

with $|s| = 11 > 6 = |Q|$.

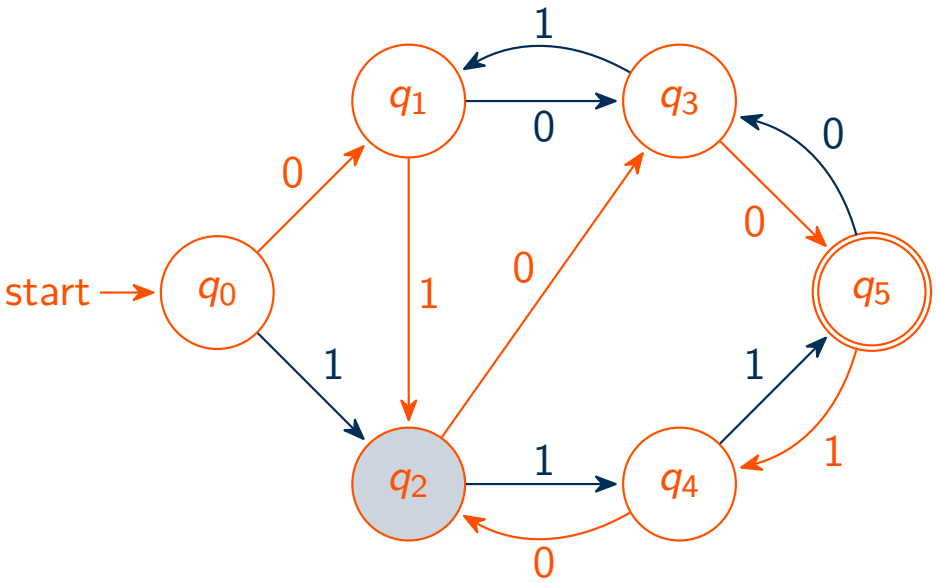


Pumping Example

Let $\Sigma = \{0, 1\}$ and consider

$$s = 01001001111$$

with $|s| = 11 > 6 = |Q|$.

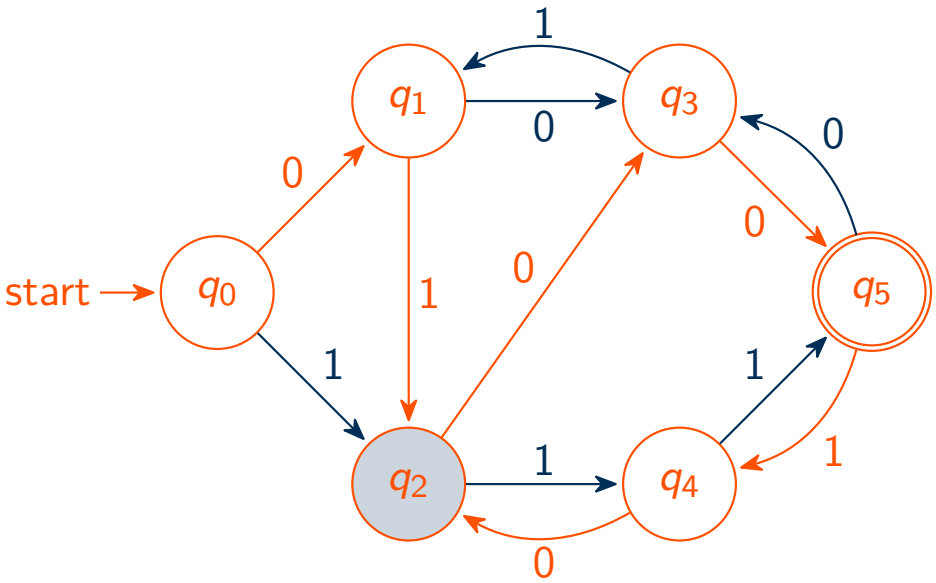


Pumping Example

Let $\Sigma = \{0, 1\}$ and consider

$$s = 010010\mathbf{01111}$$

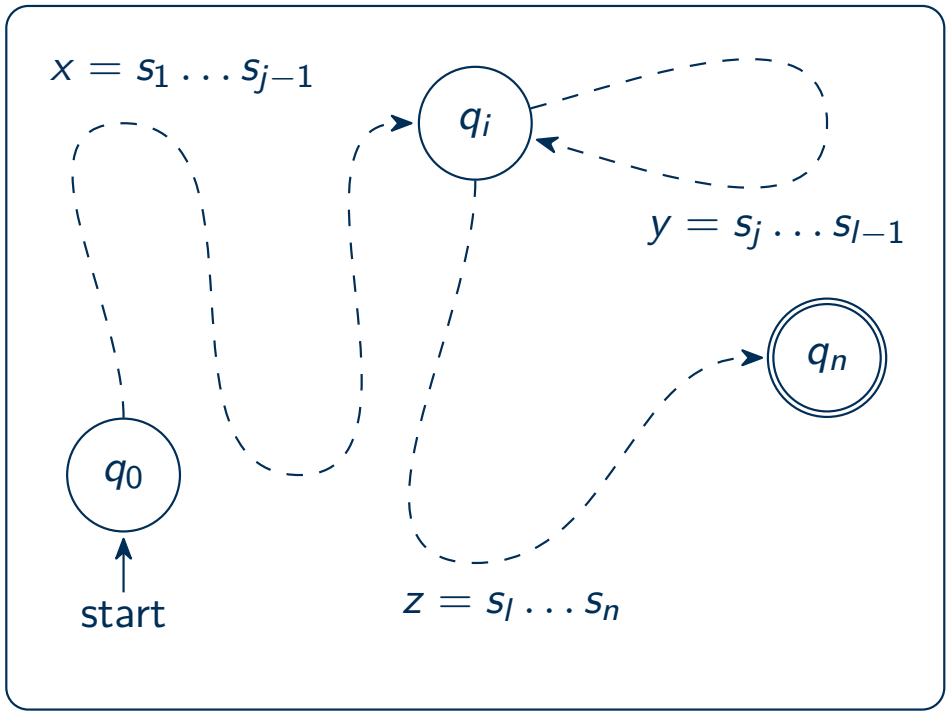
with $|s| = 11 > 6 = |Q|$.



Therefore,

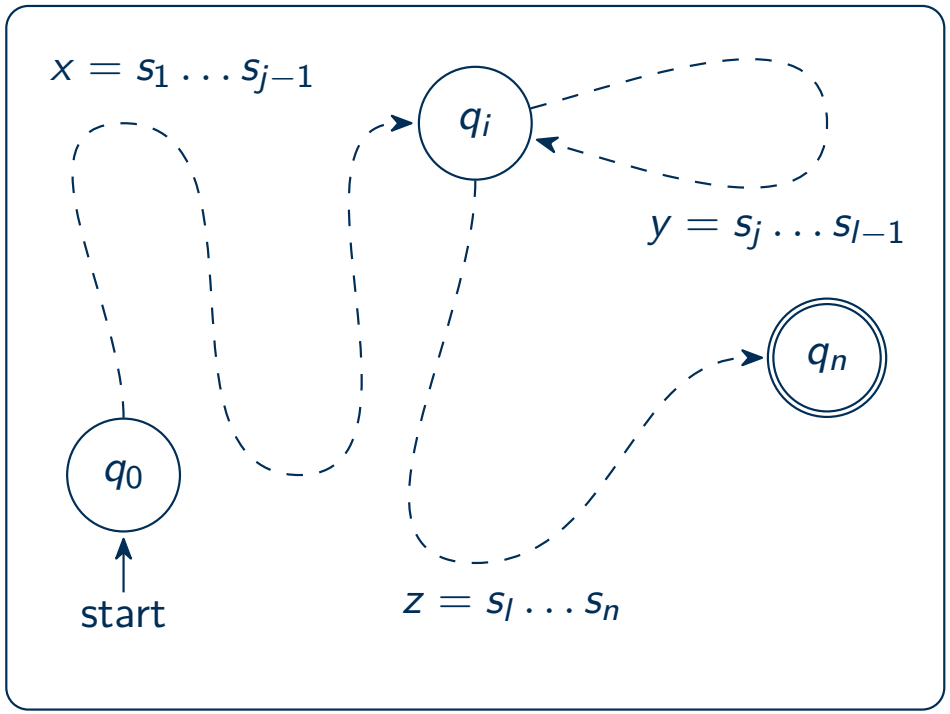
$$x = 01, \ y = 0010, \ \&, \ z = 01111$$

General Idea



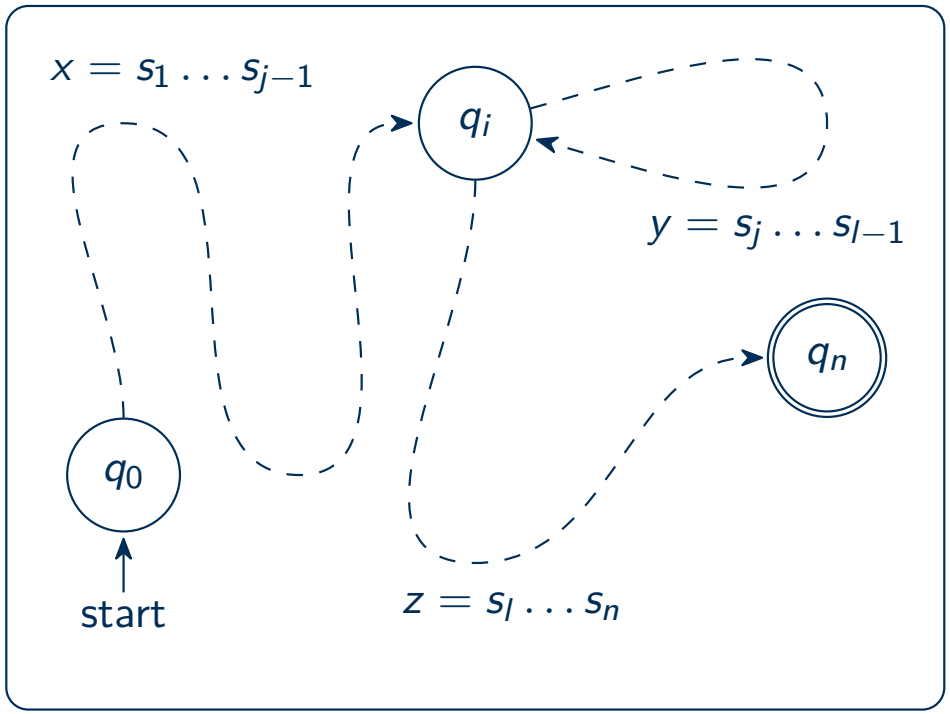
- $p = |Q|$

General Idea



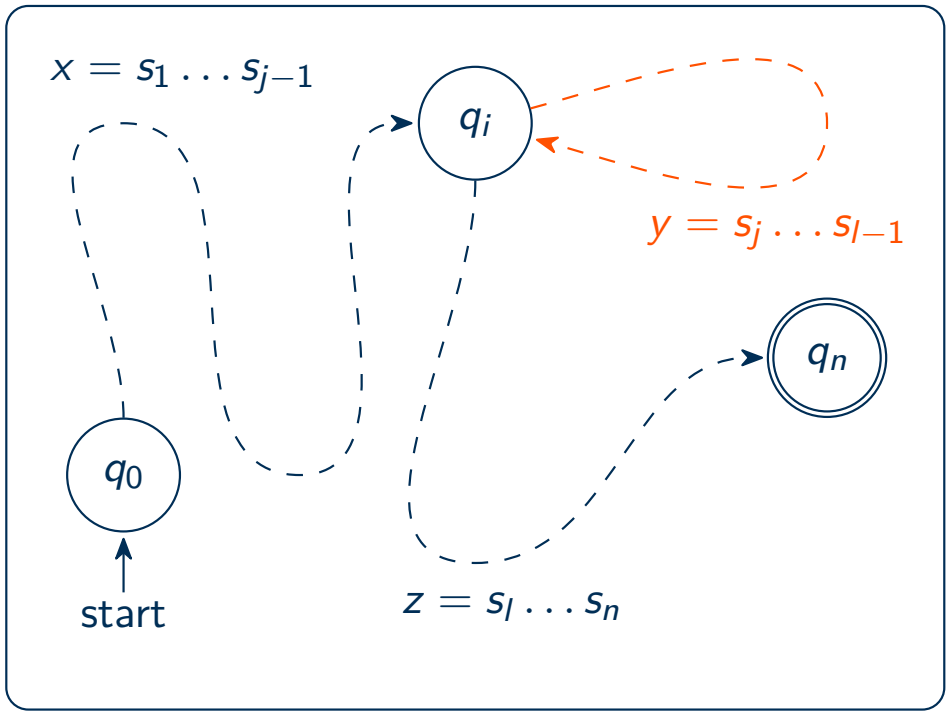
- $p = |Q|$
- $s = s_1 s_2 s_3 \cdots s_n \in A, \ n > p$

General Idea



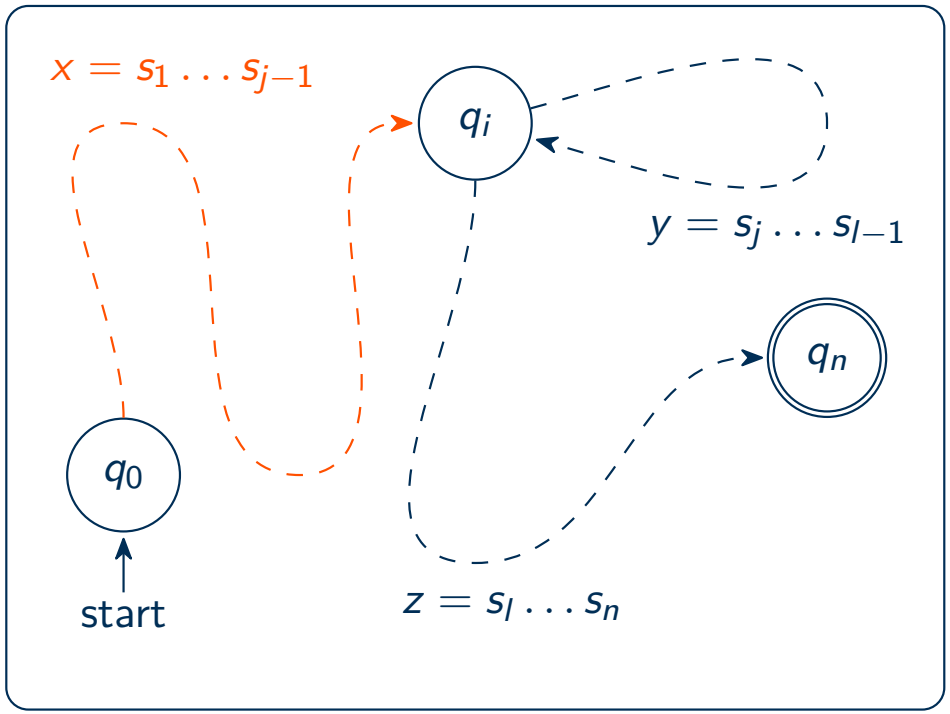
- $p = |Q|$
- $s = s_1 s_2 s_3 \cdots s_n \in A, \ n > p$
- Pigeon holes imply a loop

General Idea



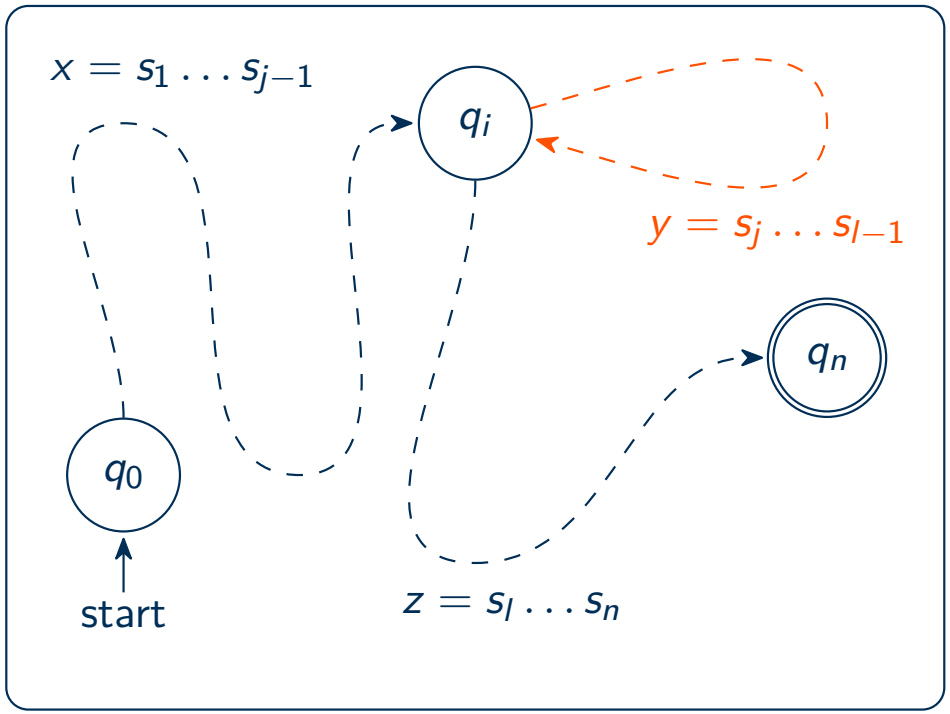
- $p = |Q|$
- $s = s_1 s_2 s_3 \cdots s_n \in A, \ n > p$
- Pigeon holes imply a loop
- Focus on the first loop

General Idea



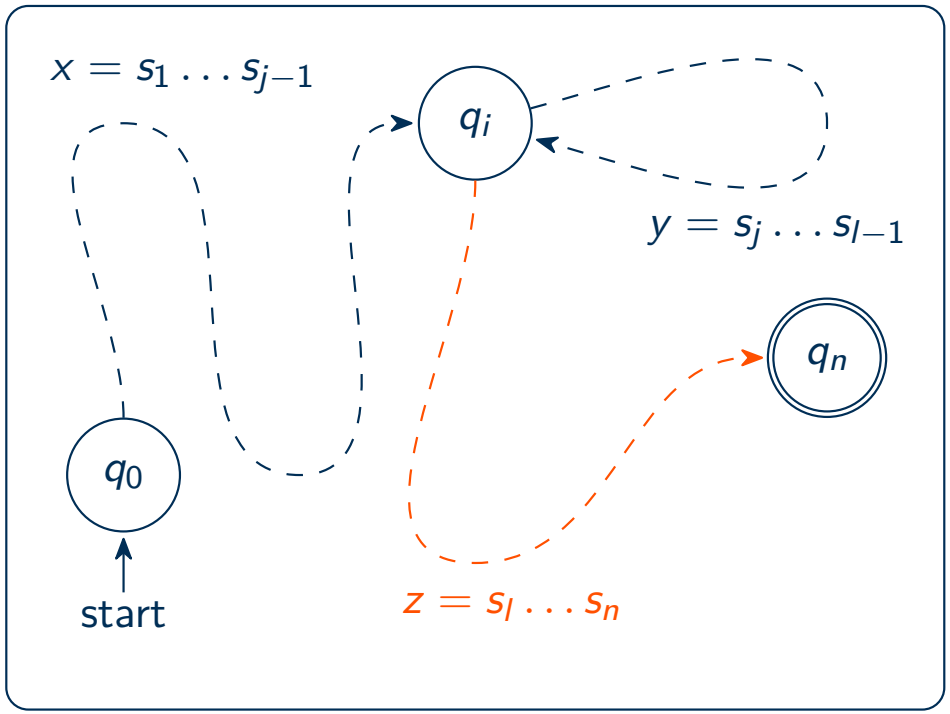
- $p = |Q|$
- $s = s_1 s_2 s_3 \cdots s_n \in A, \ n > p$
- Pigeon holes imply a loop
- Focus on the first loop
- $x = s_1 \dots s_{j-1}$

General Idea



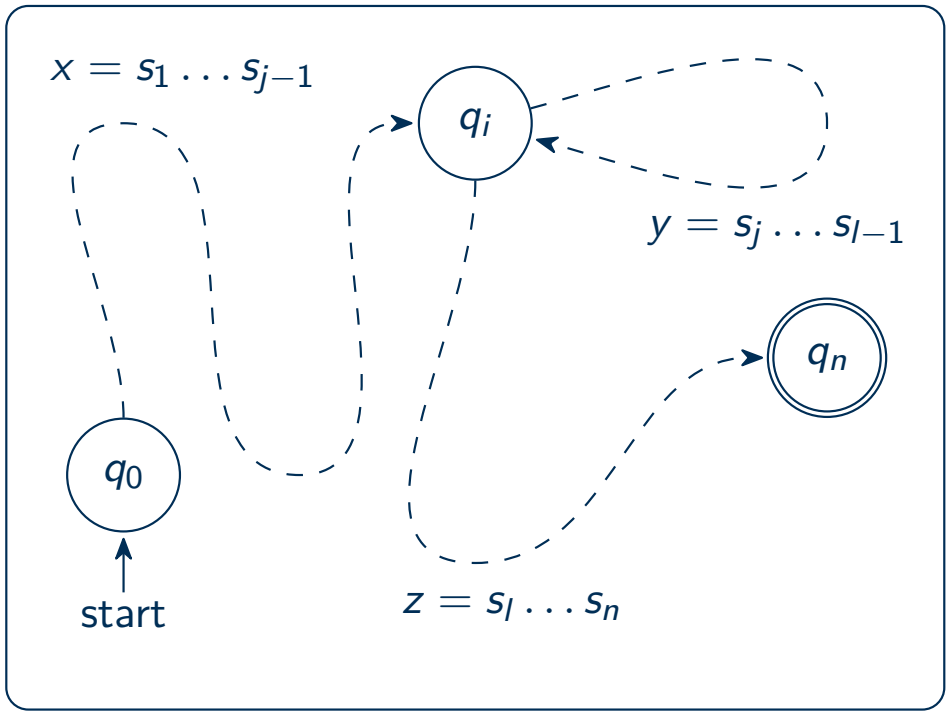
- $p = |Q|$
- $s = s_1 s_2 s_3 \cdots s_n \in A, \ n > p$
- Pigeon holes imply a loop
- Focus on the first loop
- $x = s_1 \dots s_{j-1}$
- $y = s_j \dots s_{l-1}$

General Idea



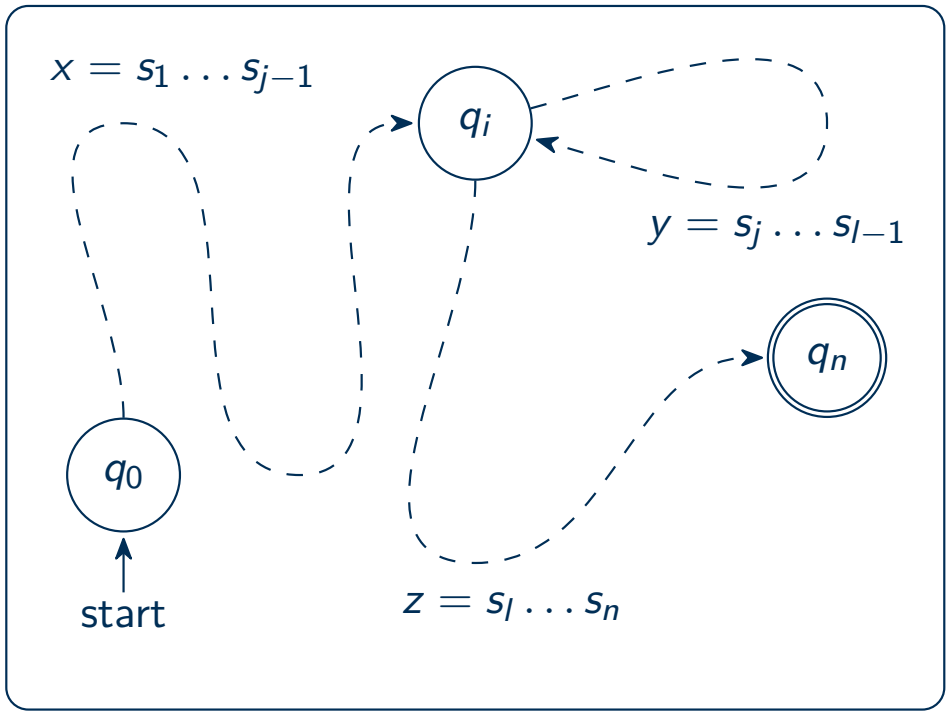
- $p = |Q|$
- $s = s_1 s_2 s_3 \cdots s_n \in A, \ n > p$
- Pigeon holes imply a loop
- Focus on the first loop
- $x = s_1 \dots s_{j-1}$
- $y = s_j \dots s_{l-1}$
- $z = s_l \dots s_n$

General Idea



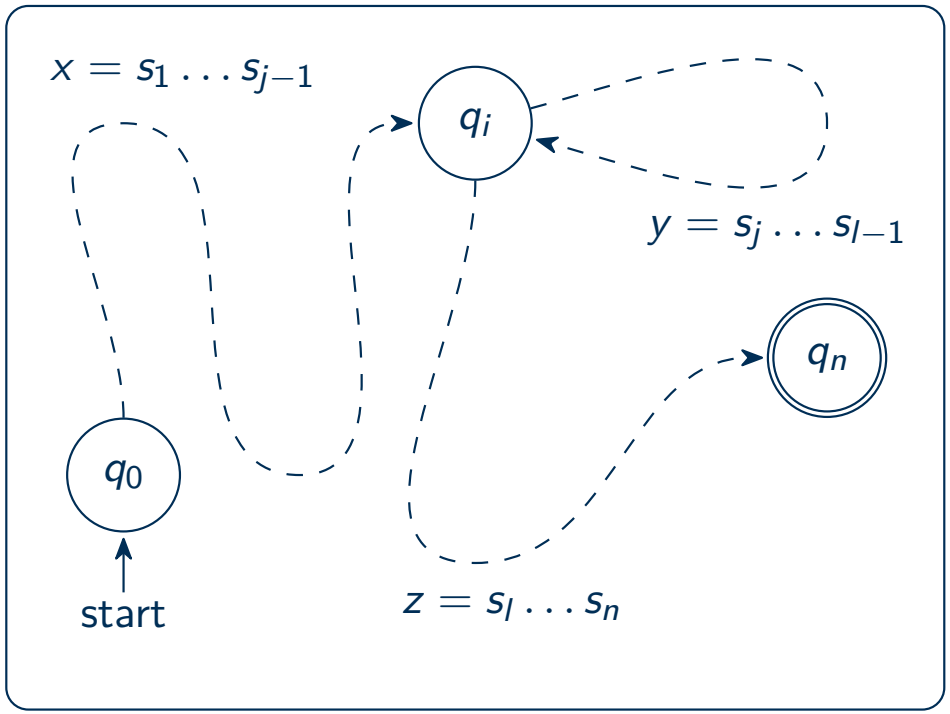
- $p = |Q|$
- $s = s_1 s_2 s_3 \cdots s_n \in A, \ n > p$
- Pigeon holes imply a loop
- Focus on the first loop
- $x = s_1 \dots s_{j-1}$
- $y = s_j \dots s_{l-1}$
- $z = s_l \dots s_n$
- $j \neq l$ implies $|y| > 0$

General Idea



- $p = |Q|$
- $s = s_1 s_2 s_3 \cdots s_n \in A, \ n > p$
- Pigeon holes imply a loop
- Focus on the first loop
- $x = s_1 \ldots s_{j-1}$
- $y = s_j \ldots s_{l-1}$
- $z = s_l \ldots s_n$
- $j \neq l$ implies $|y| > 0$
- y loops so $xy^i z \in A$

General Idea



- $p = |Q|$
- $s = s_1 s_2 s_3 \cdots s_n \in A, \ n > p$
- Pigeon holes imply a loop
- Focus on the first loop
- $x = s_1 \dots s_{j-1}$
- $y = s_j \dots s_{l-1}$
- $z = s_l \dots s_n$
- $j \neq l$ implies $|y| > 0$
- y loops so $xy^i z \in A$
- y is the first loop so $|xy| \leq p$

Statement

Theorem (Pumping Lemma)

If A is a regular language, then there is a number p (the pumping length) such that, if s is any string in A of length at least p, then s may be written in three pieces, $s = xyz$, satisfying the following conditions:

1. *for each $i \geq 0$, $xy^iz \in A$,*
2. *$|y| > 0$, and*
3. *$|xy| \leq p$.*

Table of Contents

Example of a Nonregular Language

- $A = \{0^n 1^n | n \geq 0\}$

Example of a Nonregular Language

- $A = \{0^n 1^n | n \geq 0\}$
- let $n > p$ the pumping length

Example of a Nonregular Language

- $A = \{0^n 1^n | n \geq 0\}$
- let $n > p$ the pumping length
- $|s| = |0^n 1^n| > 2p$

Example of a Nonregular Language

- $A = \{0^n 1^n | n \geq 0\}$
- let $n > p$ the pumping length
- $|s| = |0^n 1^n| > 2p$
- $xy = 0^k$ and $y = 0^j$ with $0 < j \leq k \leq p$

Example of a Nonregular Language

- $A = \{0^n 1^n | n \geq 0\}$
- let $n > p$ the pumping length
- $|s| = |0^n 1^n| > 2p$
- $xy = 0^k$ and $y = 0^j$ with $0 < j \leq k \leq p$
- $xy^i z = 0^{n+(i-1)j} 1^n \notin A$ unless i=1

Example of a Nonregular Language

- $A = \{0^n 1^n | n \geq 0\}$
- let $n > p$ the pumping length
- $|s| = |0^n 1^n| > 2p$
- $xy = 0^k$ and $y = 0^j$ with $0 < j \leq k \leq p$
- $xy^i z = 0^{n+(i-1)j} 1^n \notin A$ unless i=1
- $\therefore$ A is nonregular

Second Example

- $R = \{a^r | r \in \mathbb{N} \text{ is prime}\}$

Second Example

- $R = \{a^r | r \in \mathbb{N} \text{ is prime}\}$
- let p be the pumping length and $n > p + 2$ be prime

Second Example

- $R = \{a^r | r \in \mathbb{N} \text{ is prime}\}$
- let p be the pumping length and $n > p + 2$ be prime
- $a^n = xyz$ with $x = a^{n_x}$, $y = a^{n_y}$, and $z = a^{n_z}$

Second Example

- $R = \{a^r | r \in \mathbb{N} \text{ is prime}\}$
- let p be the pumping length and $n > p + 2$ be prime
- $a^n = xyz$ with $x = a^{n_x}$, $y = a^{n_y}$, and $z = a^{n_z}$
- $n = n_x + n_y + n_z$, $n_x + n_y \leq p$, and $n_y > 0$

Second Example

- $R = \{a^r | r \in \mathbb{N} \text{ is prime}\}$
- let p be the pumping length and $n > p + 2$ be prime
- $a^n = xyz$ with $x = a^{n_x}$, $y = a^{n_y}$, and $z = a^{n_z}$
- $n = n_x + n_y + n_z$, $n_x + n_y \leq p$, and $n_y > 0$
- $n_z = n - (n_x + n_y) \geq n - p > p + 2 - p > 2$

Second Example

- $R = \{a^r | r \in \mathbb{N} \text{ is prime}\}$
- let p be the pumping length and $n > p + 2$ be prime
- $a^n = xyz$ with $x = a^{n_x}$, $y = a^{n_y}$, and $z = a^{n_z}$
- $n = n_x + n_y + n_z$, $n_x + n_y \leq p$, and $n_y > 0$
- $n_z = n - (n_x + n_y) \geq n - p > p + 2 - p > 2$
- $k = n_x + n_z > 2$ so that

$$|xy^k z| = n_x + kn_y + n_z = (n_x + n_z) + kn_y = k(1 + n_y)$$

is not of prime length

Second Example

- $R = \{a^r | r \in \mathbb{N} \text{ is prime}\}$
- let p be the pumping length and $n > p + 2$ be prime
- $a^n = xyz$ with $x = a^{n_x}$, $y = a^{n_y}$, and $z = a^{n_z}$
- $n = n_x + n_y + n_z$, $n_x + n_y \leq p$, and $n_y > 0$
- $n_z = n - (n_x + n_y) \geq n - p > p + 2 - p > 2$
- $k = n_x + n_z > 2$ so that

$$|xy^k z| = n_x + kn_y + n_z = (n_x + n_z) + kn_y = k(1 + n_y)$$

  is not of prime length
- $\therefore$ R is nonregular

Third Example

- $F = \{0^i 1^j | i > j\}$

Third Example

- $F = \{0^i 1^j | i > j\}$
- $s = 0^{p+1}1^p$, where p is the pumping length

Third Example

- $F = \{0^i 1^j | i > j\}$
- $s = 0^{p+1}1^p$, where p is the pumping length
- $xy = 0^k$ and $y = 0^j$ with $0 < j \leq k \leq p$

Third Example

- $F = \{0^i 1^j | i > j\}$
- $s = 0^{p+1}1^p$, where p is the pumping length
- $xy = 0^k$ and $y = 0^j$ with $0 < j \leq k \leq p$
- If $i = 0$, then $xy^i z = xz = 0^n 1^p$, *with* $n \leq p$

Third Example

- $F = \{0^i 1^j | i > j\}$
- $s = 0^{p+1}1^p$, where p is the pumping length
- $xy = 0^k$ and $y = 0^j$ with $0 < j \leq k \leq p$
- If $i = 0$, then $xy^i z = xz = 0^n 1^p$, *with* $n \leq p$
- $\therefore$ F is nonregular

Table of Contents

Next Class

- Context-Free Languages

Next Class

- Context-Free Languages
- Chomsky Normal Form

Next Class

- Context-Free Languages
- Chomsky Normal Form
- Pushdown Automata

Regular Expressions and the Pumping Lemma

Dr. Chuck Rocca
roccac@wcsu.edu
http://sites.wcsu.edu/roccac

WESTERN CONNECTICUT STATE UNIVERSITY
MACRICOSTAS SCHOOL OF ARTS & SCIENCES