PART I: Automata and Languages

CHAPTER I: Regular Languages

Section 1: Finite Automata
Section 2: Regular Languages
Section 3: Nondeterminism
Section 4: Equivalence of NFA and DFA
Section 5: Closure under regular operations
Section 6: Regular expressions
Section 7: Pumping lemma for regular languages

Pumping Lemma for Regular Expressions

Not all the languages are regular.

$B = \{0^n 1^n \in \{0,1\}^* / n \geq 0\}$ – is not a regular language

$C = \{w \in \{0,1\}^* / w$ has an equal no. of 0s and 1s$\}$ – is not a regular language

$D = \{w \in \{0,1\}^* / w$ has an equal no. of occurrences of 01 and 10$\}$ – is a regular language

The pumping lemma (for regular languages) states that any regular language satisfies a certain property – the pumping property.

Consequently, to prove that a language is not regular, it is sufficient to demonstrate that the language does not have this property.

Pumping Lemma for Regular Languages

If L is a regular language, then there exists a number $p \in \mathbb{N}$, called the pumping length such that for any string $w \in L$ with $|w| \geq p$, there exists substrings x, y, z of w that satisfy the following conditions:

$w = xyz$ and
1. $\forall i \geq 0$, $xy^iz \in L$
2. $|y| > 0$
3. $|xy| \leq p$.

In logic: $L\ reg. \Rightarrow [\exists p \geq 1. \forall w \in L, \|w\| \geq p. \exists x, y, z \in \Sigma^*. (w = xyz \wedge 1. \wedge 2. \wedge 3.)]$

- Condition 1. claims that if $w = xyz$, then
  $xz, xyz, xyyz, xyyyz, \dots, x\underbrace{yy\ldots y}_{i}z \in L$, for any i.
- When we split $w = xyz$, we can have $x = \varepsilon$ or $z = \varepsilon$. But condition 2 guarantees that $y \neq \varepsilon$.
- Condition 3 states that the first two substrings have together the length smaller than p.

Pumping Lemma for Regular Languages

If L is a regular language, then there exists a number $p \in \mathbb{N}$, called the pumping length such that for any string $w \in L$ with $|w| \geq p$, there exists substrings x, y, z of w that satisfy the following conditions:

$w = xyz$ and
1. $\forall i \geq 0, \quad xy^iz \in L$
2. $|y| > 0$
3. $|xy| \leq p$.

Proof: Let $M = (Q, \Sigma, \delta, q_1, F)$ be a DFA that recognizes L.

Let p be the number of states of M.

If all the strings in L have the length smaller than p, then L is finite and our lemma is vacuously true.

Pumping Lemma for Regular Languages

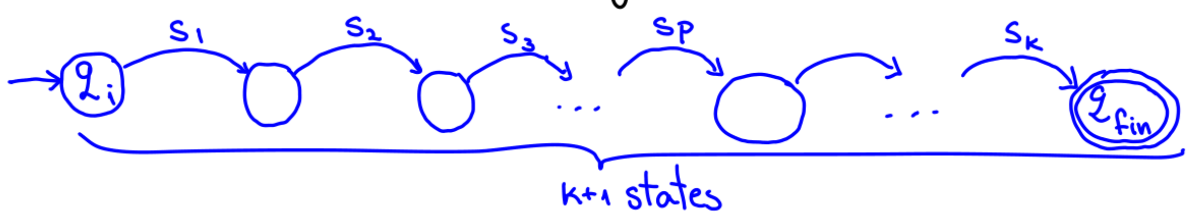
If L is a regular language, then there exists a number $p \in \mathbb{N}$, called the <u>pumping length</u> such that for any string $w \in L$ with $|w| \geq p$, there exists substrings x, y, z of w that satisfy the following conditions:

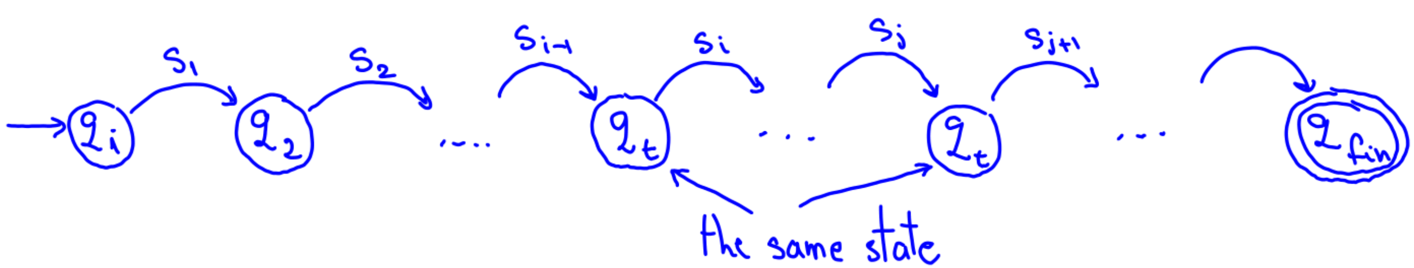
$w = xyz$ and

1. $\forall i \geq 0, \; xy^i z \in L$
2. $|y| > 0$
3. $|xy| \leq p$.

<u>Proof</u>: Let p be the number of states of M.

Let $w \in L$ be a string such that $|w| \geq p$. $w = s_1 \ldots s_p s_{p+1} \ldots s_k$





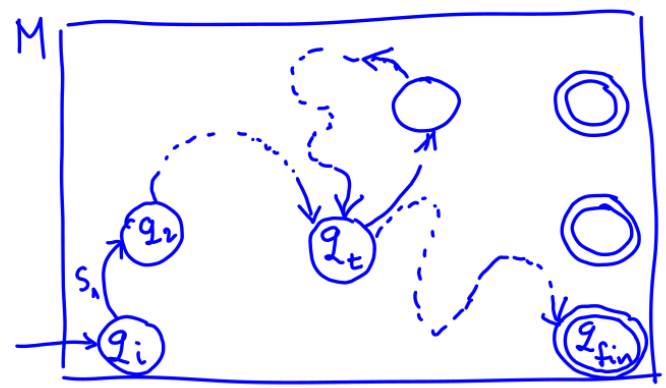
Pumping Lemma for Regular Languages

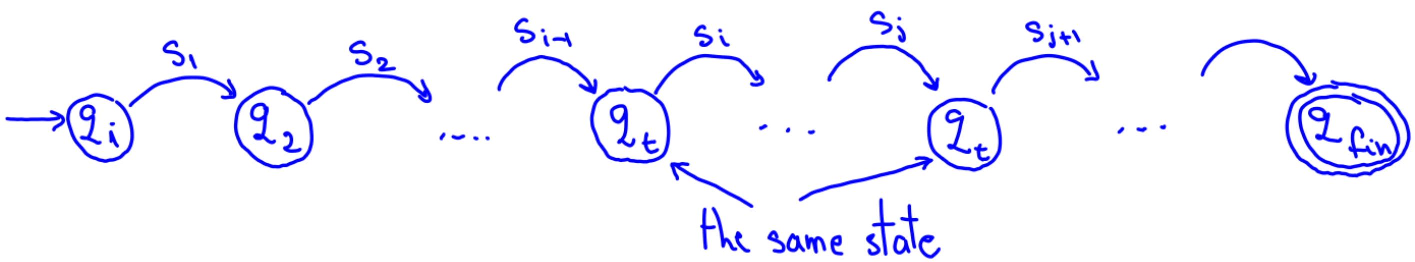
If L is a regular language, then there exists a number $p \in \mathbb{N}$, called the pumping length such that for any string $w \in L$ with $|w| \geq p$, there exists substrings x, y, z of w that satisfy the following conditions:

$w = xyz$ and

1. $\forall i \geq 0, \; xy^iz \in L$
2. $|y| > 0$
3. $|xy| \leq p$.

Proof:





Pumping Lemma for Regular Languages

If L is a regular language, then there exists a number $p \in \mathbb{N}$, called the pumping length such that for any string $w \in L$ with $|w| \geq p$, there exists substrings x, y, z of w that satisfy the following conditions:

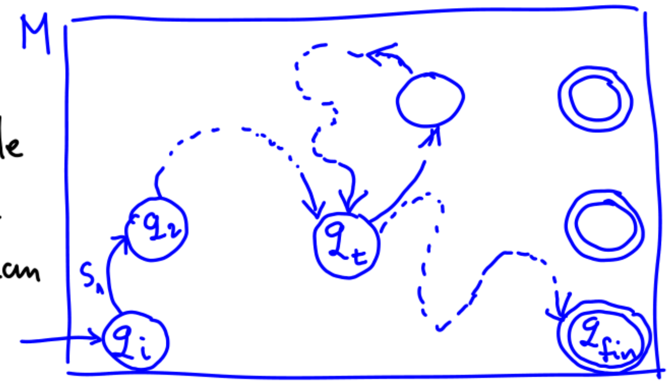
$w = xyz$ and 1. $\forall i \geq 0, \; xy^iz \in L$

2. $|y| > 0$

3. $|xy| \leq p$.

Proof: Let $w = s_1 s_2 \ldots s_k$, $k \geq p$.

Let $q_1, \ldots q_{k+1} \in Q$ be the states visited while computing w in the order they appear – notice that same states might be visited more than once.



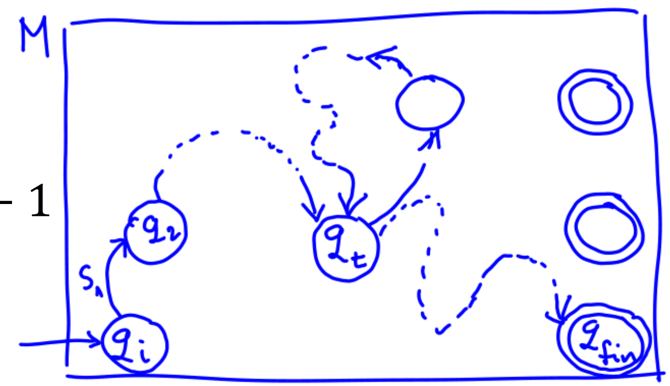
Pumping Lemma for Regular Languages

If L is a regular language, then there exists a number $p \in \mathbb{N}$, called the <u>pumping length</u> such that for any string $w \in L$ with $|w| \geq p$, there exists substrings x, y, z of w that satisfy the following conditions:

$w = xyz$ and
1. $\forall i \geq 0, \; xy^iz \in L$
2. $|y| > 0$
3. $|xy| \leq p$.

<u>Proof</u>: Let $w = s_1 s_2 \ldots s_k$, $k \geqslant p$.

Since $k+1 > p$, there exists a state q_t visited twice. Suppose that $q_t = q_\ell$ for $t < \ell \leq p+1$



Let $x = s_1 s_2 \ldots s_{t-1}$ $\quad y = s_t \ldots s_{\ell-1}$ $\quad z = s_\ell \ldots s_k$

Then, $w = xyz$.

$$\left.\begin{array}{l} t \neq \ell \Rightarrow |y| > 0 \\ \ell \leq p+1 \end{array}\right\} \Rightarrow |xy| \leq p$$

and M accepts xz, xyz, $xyyz$, $xyyyz$, etc.

Pumping Lemma for Regular Languages

If L is a regular language, then there exists a number $p \in \mathbb{N}$, called the pumping length such that for any string $w \in L$ with $|w| \geq p$, there exists substrings x, y, z of w that satisfy the following conditions:

$w = xyz$ and
1. $\forall i \geq 0,\ xy^iz \in L$
2. $|y| > 0$
3. $|xy| \leq p$.

In logic: $L\ reg. \Rightarrow [\exists p \geq 1. \forall w \in L, |w| \geq p. \exists x, y, z \in \Sigma^*. (w = xyz \wedge 1. \wedge 2. \wedge 3.)]$

Contraposition of the Pumping lemma:
$[\forall p \geq 1. \exists\ w \in L, |w| \geq p. \forall x, y, z \in \Sigma^*. ((w = xyz \wedge 2. \wedge 3.) \Rightarrow \neg 1.)] \Rightarrow L\ not. reg.$

Exercise: Using the pumping lemma, show that $\mathrm{L} = \{ a^n b^n \mid n \geq 1\}$ is not regular.
Proof: Assume towards a contradition that the above language is regular. Then, by the pumping lemma, there is some $p > 0$ such that for $a^p b^p \in L$ there exist strings x, y, z satisfying 1.-3. and $a^p b^p = xyz$. With this, we observe:
a) By 2. and 3., we know that $xy = a^k$ for some $k > 0$.
b) With this and $a^p b^p = xyz$, we know that z contains exactly p symbols b (and possibly some symbols a).
c) From $a^p b^p = xyz$, a) and b), we know that xy^2z has more a than b symbols.
d) Since c) contradicts 1., our initial assumption was wrong, i.e., L is not regular.

Recap on Chapter I : Regular Languages

Learning goals:

1. Deterministic Finite Automata: definition, state diagram, computation
2. Construction of DFA for a given language
3. Regular languages: definition, closure operations.
4. Nondeterministic Finite Automata: definition, state diagram, computation
5. Equivalence of NFA and DFA
6. Construction of NFA for a given language.
7. Construction of NFAs that recognize languages defined by regular operations: union, concatenation, star
8. Regular expressions: definition
9. Regular expressions that characterize a given language.
10. Conversions of regular expressions into NFAs.
11. Generalized Nondeterministic Finite Automata: definition
12. Conversion of DFAs into regular expressions
13. Pumping lemma for regular languages.