Pumping lemma for Regular languages

We have seen regular languages and three characterizations: Using DFA's | NFA's | regular expressions.

→ How powerful are regular languages? Are there languages which cannot be captured by DFA | NFA?

Pumping lemma: Necessary Condition for languages to be regular.

A is regular ⟹ A can be "pumped"

⟸ the converse is not true!

Contrapositive

The above implies that if A cannot be pumped, then A is not regular. This gives a way to show that certain languages are not regular.

Caution: This is not a test to show that a language is regular.

$= \{\varepsilon, 01, 0011, 000111, \ldots\}$

Example: $B = \{0^n 1^n \mid n \geq 0\}$.

A DFA that recognizes B, in some sense, has to count the number of 0's and then cross check it with the number of 1's. But n can be arbitrarily large. So this cannot be done with a limited no. of states.

This is still an intuition!

Pumping lemma gives us a way to formalize this.

Pumping lemma was discovered by Michael Rabin and Dana Scott (1959) and later by Bar-Hillel, Perles and Shamir (1961).

Theorem 1.70 (Pumping lemma): If A is a regular language, then there exists a number p

(pumping length) such that, if s is any string in A of length at least p ($\geq p$), then s can be divided into three pieces $s = xyz$, such that

(1) for each $i \geq 0$, $xy^iz \in A$ ($xz, xyz, xyyz, xyyyz, \ldots$)

(2) $|y| > 0$ ($y \neq \varepsilon$)

(3) $|xy| \leq p$.

Proof (Informal/Sketch) Refer to the SIPSER book for the formal proof.

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

We will show that pumping length $p = |Q|$.
(number of states)

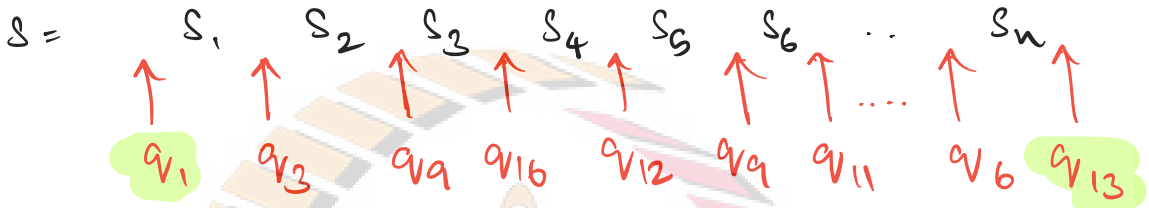
→ What if all strings $s \in A$ are of length $< p$?

Then pumping lemma holds vacuously.
(VACUOUSLY)

→ For strings $s \in A$ with $|s| \geq p$, we will use an argument based on the pigeonhole principle.

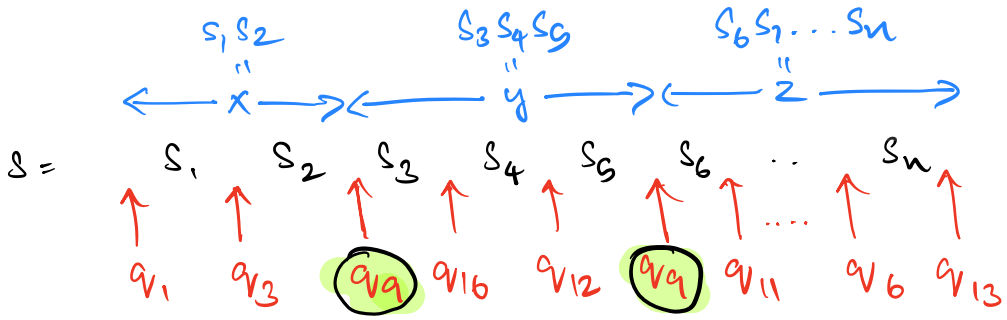
$|s|=n$

Consider the string $s = s_1 s_2 \ldots s_n$



Consider the sequence of the states that M goes through while processing s. This starts with q_1, then say q_3 ... etc. till say q_{13} → Accept State

If $|s|=n$, then this sequence has $n+1$ states, with some possible repetitions. When $|s|=n \geq p$, by pigeonhole principle, there exists at least one repeated state, since $n+1 \geq p+1 > p = |Q|$.

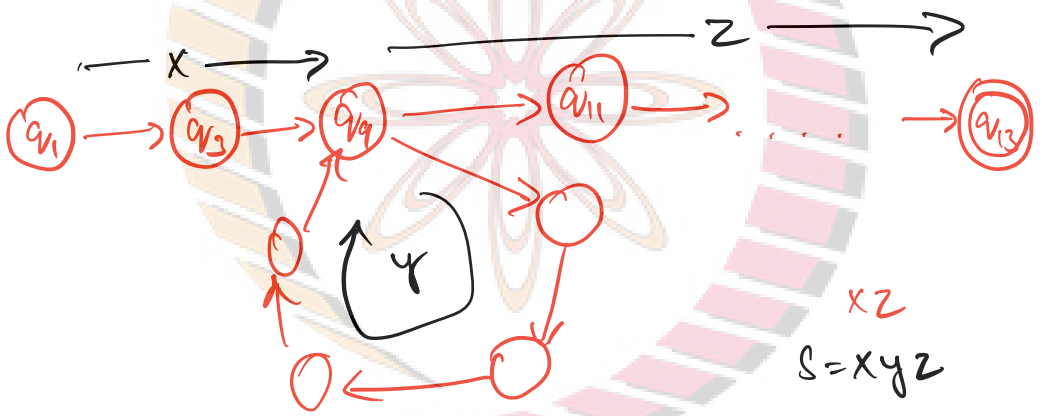


In the above, q_9 is the repeated state. Let

x = String till the first occurrence of q_9.

y = String from the first to the second occurrence of q_9.

z = String from the second occurrence of q_9 till the end of s.



xz

$s = xyz$

xyyz

xyyyz

In the DFA M, x takes M from q_1 to q_9.

y takes M from q_9 to itself.

z takes M from q_9 to q_{13}.

Consider $xyyz$. This will also be accepted by M. The difference is that there are two rounds of y instead of one. Similarly for xy^3z and xz.

Hence xy^iz is accepted by M and hence is in A for all $i \geq 0$.

Since there are two occurrences (or more!) of at least one state (q_9 here in the above example), there exists a non empty string that is processed in between. Thus $|y| > 0$ ($y \neq \varepsilon$).

Pigeonhole principle guarantees that the first repetition occurs on or before s_p (the p^{th} symbol of s) is processed. Hence $|xy| \leq p$.

Exercise: Go through the formal proof in the book.