
Pumping Lemma for Regular Language

For every regular language L ,

there exists a sufficiently large n ,

For all strings $s \in L$, $|s| \geq n$

there exist $u \cdot v \cdot w = s$

$|uv| \leq n$, $|v| \geq 1$

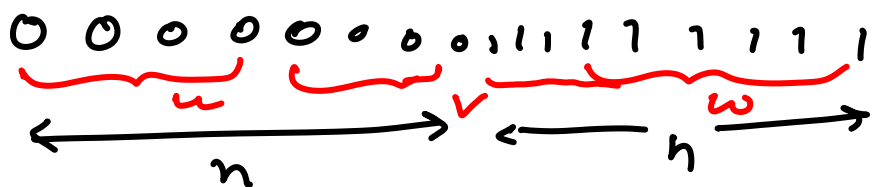
s.t. for all $i \geq 0$ $u \cdot v^i \cdot w \in L$

$$L' = \{ 0^i 1^i \mid i \geq 0 \}$$

Is L' regular?

Suppose L' is regular \Rightarrow we can apply PL to L

Then there exists some n and let us choose $s = 0^n 1^n$ as the string

$s =$ 

By PL $u^i v^i w^i \in L$ for all i and thereby produce strings $\in L$ st $\#0s > \#1s$
 So the assumption that L' is regular is false

$$L_{\text{prime}} = \{ 0^k \mid k \text{ is prime} \}$$

$$\{ 0^2, 0^3, 0^5, 0^7, \dots \}$$

Suppose L_{prime} is regular (P.L. constant n)

Choose a sufficiently long string, say 0^n



$$|u| = k \quad |v| = l \quad |w| = m$$

From P.L. $uv^i w \in L_{\text{prime}}$ for all $i \geq 0$

$$\Rightarrow 0^{k + il + m} \in L_{\text{prime}}$$

Choose $i = k + m$

$$k + (k + m)l + m \in P$$

$$(k + m)(1 + l) \in P$$

$$L_c = \{ 0^k \mid k \text{ is } \boxed{\text{not prime}} / \text{composite} \}$$

Is L_c regular?

Claim If L is regular, then \bar{L} ($\Sigma^* - L$) is regular

Choose a DFA for L and invert the Final and non-Final states

It follows L_c is not regular

Claim: If L_1, L_2 are regular, then $L_1 \cap L_2$ is regular.

DFA: M_1, M_2

Construct a DFA in the following manner

$$M = M_1 \times M_2 : \{ Q_1 \times Q_2, \Sigma, (q_1^0, q_2^0), F_1 \times F_2, \delta \}$$

$$\delta : ((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$$

$p \in Q_1, q \in Q_2$

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}} \quad (\text{De Morgan's law proof})$$

Suppose for each symbol $a \in \Sigma$

we define a substitution function

$f(a) =$ regular expression