More About Regular Languages

CS 712
Dept. of Computer Science
Univ. of New Hampshire
Pumping Lemma for Regular Languages

Let $L$ be a regular language accepted by a DFA $M$ with $K$ states. Let $z$ be any string in $L$ with $|z| \leq K$. Then $z$ can be rewritten $uvw$ with $|uv| \leq K$, $|v| \geq 1$, and $uv^iw \in L$ for all $i \geq 0$.

Proof

Since $|z| \leq K$, some state in $M$ must be visited twice while accepting $z$. So there must be a "cycle" in $M$. This cycle could be traversed an arbitrary number of times.

\[ \Rightarrow u \to o \to w \to o \]

i.e. regular expression labels extraneous nodes eliminated
Using the pumping lemma:

\[ L = \{ a^n b^n \mid n \geq 0 \} \]

Is \( L \) regular?

Assume it is and that it is recognized by a DFA with \( k \) states.

Consider \( s = a^k b^k \) which is in \( L \).

By pumping lemma: \( s = a^k b^k = uvw \), where \( |uv| \leq k \), \( |v| > 0 \). Therefore \( v \) must consist of only \( a \)'s. Call \( lv \) \( v \). Reproduce \( s \) once to produce \( s' = a^{k+\ell} b^k \). But \( s' \) is not in \( L \). Contradiction. So \( L \) must not be regular.

Key: you don't get to choose \( u, v, w \). Your proof must work for all possible \( u, v, w \).
Properties of regular languages:

If \( L_1 \) and \( L_2 \) are regular languages:

1. \( L_1 \cup L_2 \) is regular
2. \( L_1 \cdot L_2 \) is regular
3. \( L_1^* \) is regular

\( L_1^* \) is regular

Take the DFA that recognizes \( L_1 \) and construct a new DFA from it by making all its final states to be non-final and all its non-final states to be final.
5. $L_1 \cap L_2$ is regular

Use DeMorgan's Law:

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$
Using intersection:

$L$ is the language over $\{a, b\}$ consisting of all strings with an equal number of $a$'s and $b$'s.

Is $L$ regular?

Consider $M$, the language described by the regular expression $a^* b^*$

$L \cap M = \{a^n b^n \mid n \geq 0\}$ which we know is not regular. Therefore $L$ cannot be regular.