More About Regular Languages

CS712

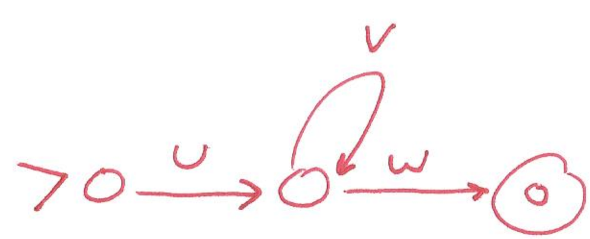
Dept. of Computer Science
Univ. of New Hampshire

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 $|z| \geq K$. Then z can be rewritten uvw with $|uv| \leq K$, $|v| > 0$ and $uv^iw \in L \;\; \forall i \geq \emptyset$.

Proof

Since $|z| \geq K$, 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.



i.e. regular expression labels
extraneous nodes eliminated

Using the pumping lemma:

$L = \{a^n b^n \mid n \geq \emptyset\}$

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 $|uv| \leq K$, $|v| > \emptyset$. Therefore v must consist of only a's. Call $|v|$ n. Pump once to produce $s' = a^{K+n} b^K$. 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 & L_2 are regular languages:

1. $L_1 \cup L_2$ is regular
2. $L_1 L_2$ is regular
3. L_1^* is regular

by definition of regular languages

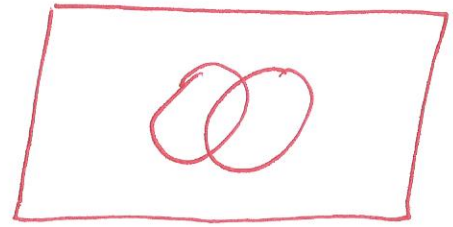
4. $\overline{L_1}$ is regular

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.