**Transition graphs for reg lang**

1. If the graph has no cycle then language is finite and regular
2. All finite lang are regular
3. If the transition graph has cycles with non-empty labels then the language is infinite
4. Every infinite regular language has a DFA with a cycle.
5. If there is a cycle, we can either skip the cycle or repeat it an arbitrary number of times
6. If position of cycle in a DFA is not known, and if the DFA has m states then the cycle must be entered by the time m input symbols have been read.

If for some language L, if atleast 1 string in L does not have the above properties, the lang L is non-regular.

**Pumping Lemma for Regular Languages**

It is a lemma that captures the above properties, using the Pigeon Hole Principle.

Let L be an infinite language. There exists some positive integer m such that w belonging to L has length >= m (i.e. |w| >= m) can be decomposed as

w = xyz

With |xy| <= m and |y| >= 1 such that

wi = x(y^i)z

That is, all the other possible strings in the language can be generated using the form above, by “pumping” together the 3 substrings.

Eg.

**Proof of non-regularity by pumping lemma:**

L = {a^n. b^n}

The string aabb is a part of the above language. Now let us decompose the string.

Let x = a, y = a, z = bb

By repeating the string y, we are generating strings that are not part of the language L. Hence, the language L is not regular.

Proof of Pumping Lemma:

Consider an infinite regular language L. This implies there is a DFA that recognizes it. Let the states of the DFA be

{q0, q1 ... qn}

Take a string w belonging to L such that |w| = n +1.

Now feed w to the DFA, and observe the sequence of states. Let this transition be

{qi, qj ... qf}

Since the string has n+1 symbols, there must exist a repetition in the string w, because of PHP.

String has n+1 symbols, so the sequence must look like

{qi, qj ... qk, qk ... qf} i.e. atleast 1 state is repeated.

Now decompose w into x, y and z, such that

Delta\*(q0, x) = qk

Delta\*(qk, y) = qk

Delta\*(qk, z) = qf

Where |xy| <= n+1 and |y| >= 1

Hence Delta\*(q0, xz) = qf and Delta\*(q0, x.(y^i).z) = qf.