Assignment - I

(1) Show that if L is regular, so $L - \{\lambda\}$.

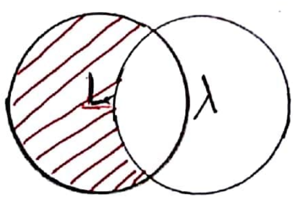
→ A language is regular, by definition, if you can create a DFA for it.

The regular languages are closed under the various operations. means result of the different operations are also regular

Example: If L and λ is regular then

Union $\lambda \cup L$, intersection $\lambda \cap L$, and complement of L and λ are also regular, hence also relative complement $L - \{\lambda\}$ is regular.

Consider the diagram --->



if L and λ is regular then $L - \{\lambda\}$ is shown by red part of this diagram and it also regular

Same if $\lambda \in L$ then $L - \{\lambda\}$ is also regular

Assignment - II

1. Exhibit an algorithm that, given any regular language L, determine whether or not $L = L^*$

→ Regular (L) --> Regular (L^*), but that does not mean that $L == L^*$. Just because two languages are both regular does not mean that they are same regular language. for instance a^* and b^* are both regular languages, but this does not make them the same language.

A Example of $L != L^*$ would be the language $L = a^*b^*$, and thus $L^* = (a^*b^*)^*$. The string abab is part of L^* but not part of L.

As far as an algorithm goes, let me remind you that the concept of a regular language is one that can be parsed by a DFA - and for any given DFA, there is a single optimal reduction of that DFA.