Properties of Regular Languages

A. Every finite language is regular (because we can write a regular expression for it).

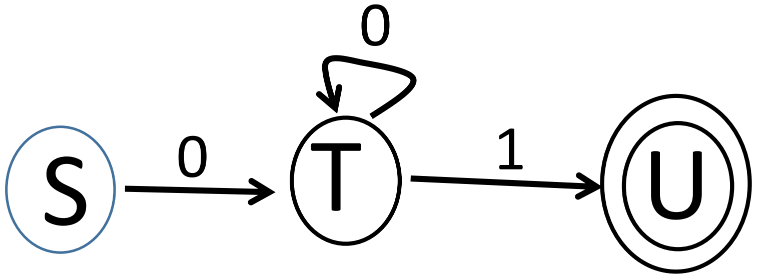
B. The union of two regular languages is regular (if E is a regular expression for one language and F for another, then E+F is a regular expression for the union of the two languages.

C. The number of regular expressions is countable (there is only a finite set of regular expressions of a fixed length n), so the number of regular languages is countable.

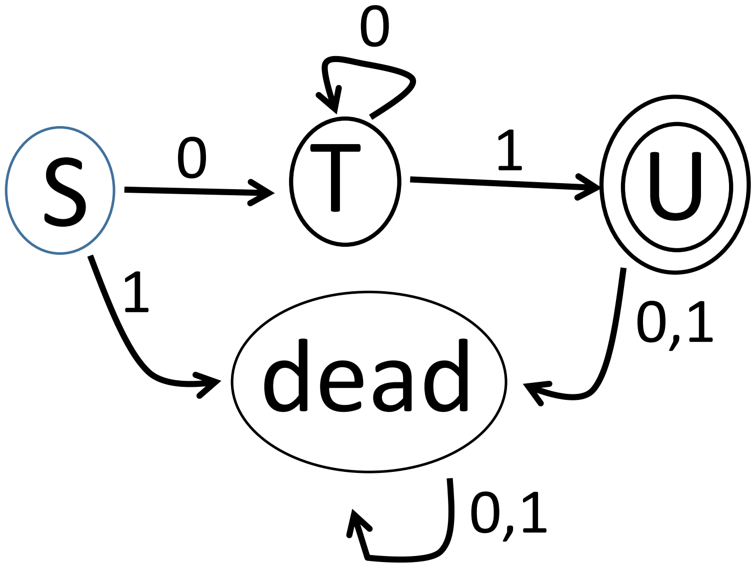
D. The subsets of a regular language are not necessarily regular.

E. Theorem: The complement of a regular language is regular.
Proof: Start with a DFA that accepts the regular language. "Complete" the DFA by adding a "dead" state and fill in any missing transition with a transition to the dead state. Now every string takes the DFA from the start state to some last state, and the string is accepted if that last state is final. Make a new DFA that has the same states and transitions as this, but whose final states are those that aren't final in the original automaton. This automaton accepts the complement of the language accepted by the original automaton.

For example, consider the language 00^*1. This is accepted by

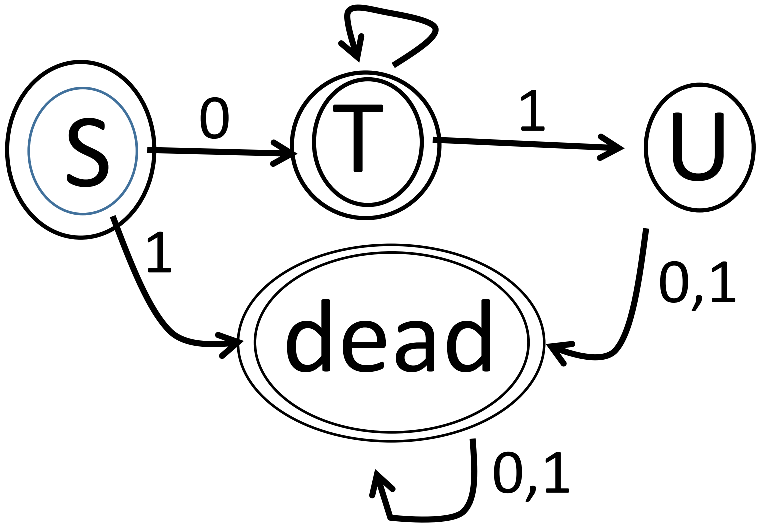


We complete this:



The completed DFA also accepts 00^*1.

If we take the complement of the final states we get a DFA that accepts every string *not* accepted by the original DFA:



Moral: Every *cofinite* language (i.e., every language whose complement is finite) is regular.

F. The intersection of two regular languages is regular.

Proof: $L_1 \cap L_2 = (L_1^c \cup L_2^c)^c$

G. The difference of two regular languages is regular.

Proof: $L_1 - L_2 = L_1 \cap L_2^c$

H. Theorem: If $\mathcal{L}$ is a regular language then $\mathcal{L}^{rev}$, the language consisting of the reversal of every string in $\mathcal{L}$, is also regular.

Proof: We'll do structural induction over the structure of a regular expression to show that the reversal of a regular expression for language $\mathcal{L}$ is a regular expression for the language $\mathcal{L}^{rev}$. This is clearly true for the base cases $\emptyset$, ε, and a. Now suppose that E and F are regular expressions representing $\mathcal{L}_1$and $\mathcal{L}_2$, and that E^{rev} and F^{rev} represent the reversals of $\mathcal{L}_1$and $\mathcal{L}_2$. Then

$(E+F)^{rev} = F^{rev} + E^{rev}$ represent $(\mathcal{L}_1 \cup \mathcal{L}_2)^{rev}$.

$(EF)^{rev} = F^{rev}E^{rev}$ represents $(\mathcal{L}_1\mathcal{L}_2)^{rev}$

$(E^*)^{rev} = (E^{rev})^*$ represents $(\mathcal{L}_1{}^*)^{rev}$

Note that a language can be pumpable but still not regular.
For example consider $\mathcal{L}$={$a^ib^jc^k$|i,j,k >=0 and if i=1 then j=k}.
$\mathcal{L}$ contains strings such as $a^1b^3c^3$, $a^2b^3c^5$, etc.

Note that $\mathcal{M}$ = {$a^1b^jc^k$} is regular because it is represented by ab*c*. If $\mathcal{L}$ was regular then $\mathcal{L}\cap\mathcal{M}$ would be regular; but $\mathcal{L}\cap\mathcal{M}$ is {$a^1b^jc^j$} and that is not regular. So $\mathcal{L}$ is not regular.

We can write $\mathcal{L} = \mathcal{L}_0 \cup \mathcal{L}_1 \cup \mathcal{L}_2$, where $\mathcal{L}_0$={b^jc^k}, $\mathcal{L}_1$={ab^jc^j}, and $\mathcal{L}_2$={$a^2a^ib^jc^k$}. $\mathcal{L}_0$ and $\mathcal{L}_2$ are regular so they are pumpable. If w=ab^jc^j is a string in $\mathcal{L}_1$, let x=ε,y=a, and z=b^jc^j. Then w = xyz and $xy^nz=a^nb^jc^j$ is in $\mathcal{L}$ for every n. So L is pumpable, even though it is not regular.