Avi Hirsch
Design and Analysis of Algorithms
Homework: Reg Langs, Closure, and NFA Transformations

Closure Under "Plus"

1. **For every language A over alphabet Σ, let $A^+ = \{x_1 \dots x_k \mid k \geq 1$ and each $x_i \in \Sigma^*\}$. Show that the class of regular languages is closed under the "plus" operation.**

   a. We know that every regular language A is closed under all regular expressions. Thus, A is closed under the "star" operation, where $A^* = \{x_0\ x_1 \dots x_k \mid k \geq 0$ and each $x_i \in A\}$.

   We can observe that A^+ is equal to A^* with the empty string (x_0 or $\{\varepsilon\}$) removed, or $A^* = \{\varepsilon\} \cup A^+$.

   Because A^+ contains every string in A^* besides for the empty string, it can be written as the concatenation of {anything accepted by A} with $\{A^*\}$, or $A \circ A^*$. Concatenating A with A^* effectively eliminates the possibility of the empty set that A^* includes.

   Both A and A^* are regular languages and are therefore closed under concatenation. Thus, A^+ is also a regular language.

NFA Transformations

1. **Show that a language is regular *if and only if* it can be recognized by some NFA with at most one accepting state.**

   a. To prove an "if and only if," we must prove "if" and "only if" separately.

   "If"

   By definition, every NFA represents a regular language; thus, *if* a language is recognized by some NFA with one accepting state, it is a regular language.

   "Only If"

   Every NFA can be written as an equivalent NFA with at most one accepting state. This can be proven via construction, as follows:

   1. Construct an NFA N_1 with some number of accepting states.
   2. Construct a new NFA, N, that **adds an epsilon transition** from each accepting state of N_1 to a single new accepting state.
   3. Make this new state the only accepting state of N.

   Thus, *only if* a language can be recognized by some NFA with at most one accepting state is it a regular language.