Regular Expressions

If $A$ and $B$ are languages

- $A \cup B$ is the union of $A$ and $B$
- $AB$ is the concatenation of $A$ and $B$: contains all strings that are concatenation of string in $A$ and string in $B$
- $A^*$ is all strings that can be built by arbitrarily concatenating strings in $A$

Example: $A = \{\text{ont, cow}\}$ $B = \{\text{at, hash}\}$

$A \cup B = \{\text{ont, cow, at, hash}\}$

$AB = \{\text{ontat, onthash, cowat, cowhash}\}$

$A^*$ includes $\epsilon, \text{ont, cow, cowcowant, ...}$

A regular expression (RE) generates a language using these 3 operations and recursion.
Example

Give RE for all binary strings:

a. starting with 11
b. ending with 11
c. alternates 0s and 1s
d. has even number of 1's.

\[
\begin{align*}
\text{a) } & 11 (0+1)^* \\
\text{b) } & \Sigma^* 11 \quad \text{(We use } \Sigma \text{ for the alphabet)} \\
\text{c) } & (1+\varepsilon) (01)^* (0+\varepsilon) \\
\text{d) } & (1^*01^*01^*)^* + 1^*
\end{align*}
\]