Language Theory

Definition Alphabet Σ: a finite set of letters s
Σ^*: the set of all finite length strings with letters in Σ, including λ (empty string)
L: language over an alphabet Σ is a subset of Σ^*

Definition Operations on languages and strings: Let $x, y \in \Sigma^*$,
xy or $x \cdot y$ is the string consisting of all letters of x followed by all letters of y
Ex. $aab \cdot ba = aabba$
λ: identity, $x\lambda = \lambda x = x \in \Sigma^*$

If $L, L' \subseteq \Sigma^*$, then $LL' = \{ xy \mid x \in L, y \in L' \}$
Ex. $L = \{a, bb\}, L' = \{\lambda, c\}, LL' = \{a, ac, bb, bbc\}$
$L^0 = \{\lambda\}$
$L^1 = L$
$L^2 = LL, L^k = L \dots L$ (kth L), $L^k = LL^{k-1} = L^{k-1}L$
$L^* = \bigcup_{k\geq 0} L^k, L^+ = \bigcup_{k>0} L^k, L^+ = L^*$ IFF $\lambda \in L$
$[\lambda\} = \{\lambda\}^* = \{\lambda\}^+$
$\emptyset \subseteq \Sigma^*, \emptyset^* = \emptyset^+ = \{\lambda\}$
$L \cup L', L \cap L'$ is the same as the set notation
$\bar{L} = \Sigma^* - L$

Definition Proper prefix/suffix/substring
x is (proper) prefix/suffix/substring is $\exists x'$ s.t. $(y \neq x \text{ AND})\ y = xx' \mid x'x \mid x'xx''$

Definition Regular Expression: Let Σ be a finite alphabet that doesn't include $\lambda, \emptyset, (,), +, *, \cdot$
The set of regular expressions over Σ R is the defined inductively as the set of strings
(notice: all symbols defined below are strings without its actual meaning)
Base case: $\emptyset, \lambda, \Sigma \in R$
Constructor case: $r, r' \in R$ IMPLIES $(r + r'), rr', r^* \in R$

Tha language $\mathcal{L}: R \to \mathcal{P}(\Sigma^*)$ denoted by a regular expression r is defined
$\mathcal{L}(\emptyset) = \emptyset, \mathcal{L}(\lambda) = \lambda, \forall a \in \Sigma, \mathcal{L}(a) = \{a\}$
$\mathcal{L}(r + r') = \mathcal{L}(r) \cup \mathcal{L}(r')$
$\mathcal{L}(rr') = \mathcal{L}(r)\mathcal{L}(r')$
$\mathcal{L}(r^*) = \mathcal{L}(r)^*$

A generalized regular expression includes $(r \cap r'), \bar{r}, (r - r') \in R$, and they are defined
$\mathcal{L}(r \cap r') = \mathcal{L}(r) \cap \mathcal{L}(r'), \mathcal{L}(r - r') = \mathcal{L}(r) - \mathcal{L}(r'), \mathcal{L}(\bar{r}) = \Sigma^* - \mathcal{L}(\bar{r})$

A language is regular IFF $L = \mathcal{L}(r)$ for some regular expression r
For $r, r' \in R, r = r'$ IFF $\mathcal{L}(r) = \mathcal{L}(r')$

Example
Set of strings over {a, b, c} that starts with ab:
$L_1 = \mathcal{L}(ab(a + b + c)^*)$
Set of strings over {0, 1} with even parity (even number of 1's):
$L_2 = \mathcal{L}\big((0^*10^*10^*0)^*\big)$
Set of strings over {0, 1} with first and last letters are different
$L_3 = \mathcal{L}(1(0 + 1)^*0 + 0(0 + 1)^*1)$
Set of strings over {a, b, c} that begin and end with different letters
$L_4 = \mathcal{L}\big(a(a + b + c)^*(b + c) + b(a + b + c)^*(a + c) + c(a + b + c)^*(a + b)\big)$
$\{ 0^m 1^n \mid m + n \text{ is odd} \}$
$L_5 = \mathcal{L}\big((00)^*(0 + 1)(11)^*\big)$