Problem 1 (10 pts). Given an alphabet $\Sigma$, sketch an algorithm to decide whether $(R^* + S^*) \cong \Sigma^*$, for any two regular expressions $R$ and $S$ over $\Sigma$.

Problem 2 (20 pts). Let $\Sigma$ be an alphabet. Recall that a binary relation, $\sim$, on $\Sigma^*$, is left invariant iff $u \sim v$ implies that $wu \sim wv$ for all $w \in \Sigma^*$ and right invariant iff $u \sim v$ implies that $uw \sim vw$ for all $w \in \Sigma^*$. An equivalence relation on $\Sigma^*$ that is both left and right-invariant is called a congruence. Recall that a congruence satisfies the property: If $u \sim u'$ and $v \sim v'$, then $uv \sim u'v'$ (You do not have to prove this). Also recall that there is a version of the Myhill-Nerode theorem that says that a language, $L$, is regular iff it is the union of equivalence classes of a congruence with a finite number of equivalence classes. (You do not have to prove this). Finally, recall that the reversal of a string, $w \in \Sigma^*$, is defined inductively as follows:

\begin{align*}
\epsilon^R &= \epsilon \\
(ua)^R &= au^R,
\end{align*}

for all $u \in \Sigma^*$ and all $a \in \Sigma$.

(i) Let $\sim$ be a congruence (on $\Sigma^*$) and assume that $\sim$ has $n$ equivalence classes. Define $\sim_R$ and $\approx$ by

$$u \sim_R v \iff u^R \sim v^R, \quad \text{for all } u, v \in \Sigma^* \quad \text{and} \quad \approx = \sim \cap \sim_R.$$  

The relation $\approx$ is clearly a congruence (You do not have to prove this). Prove that $\approx$ has at most $n^2$ equivalence classes.

(ii) Given any regular language, $L$, over $\Sigma^*$ let

$$L' = \{w \in \Sigma^* \mid ww^R \in L\}.$$
Prove that $L'$ is also regular, using the relation $\approx$ of part (i).

*Hint.* Use the usual version of the Myhill-Nerode theorem applied to the relation $\approx$.

**Problem 3 (20 pts).** Prove that the following languages are not regular:

$$L_1 = \{a^m b^n c^m \mid m, n \geq 1\},$$
$$L_2 = \{a^n \mid n \text{ is not a prime}\}.$$

**Problem 4 (10 pts).** Give a context-free grammar for the language:

$$L_3 = \{a^m b^m c a^{2n} b^{2n} \mid m, n \geq 1\},$$

where $\Sigma = \{a, b, c\}$.

**Problem 5 (20 pts).** Give an algorithm to decide whether $L \subseteq \{aa, bb\}^*$, where $L \subseteq \{a, b\}^*$ is any context-free language.

**Problem 6 (25 pts).** (i) Prove that the following sets are not recursive ($\varphi_1, \varphi_2, \ldots, \varphi_i, \ldots$ is any acceptable indexing of the partial recursive functions):

$$A = \{i \in \mathbb{N} \mid \varphi_i(0) = \varphi_i(1) \text{ and } \varphi_i(0), \varphi_i(1) \text{ are both defined}\}$$
$$B = \{i \in \mathbb{N} \mid \varphi_i = \varphi_a + \varphi_b\}$$
$$C = \{(i, j, k) \in \mathbb{N} \mid \varphi_i = \varphi_j + \varphi_k\}$$
$$D = \{i \in \mathbb{N} \mid \varphi_i \text{ diverges for exactly one input}\}$$

where $a$ and $b$ are two fixed natural numbers.

(ii) Prove that $A$ is recursively enumerable.

**Problem 7 (15 pts).** Consider the version of the tiling problem where an instance of the problem is of the form $((T, V, H), \hat{s}, \sigma_0)$, where $(T, V, H)$ is a tiling system as before, but $\hat{s}$ is *in tally notation* and $\sigma_0$ is a single tile placed in position $(1, 1)$. Recall that the tally notation of a natural number $n$ is $1 \cdots 1$ with $n + 1$ occurrences of “1”. For example, the tally notation for the decimal number 10 is 1111111111.

Prove that this new tiling problem is also $NP$-complete.