1 Problems

1. Induction: Consider the context-free grammar $G = \langle V, T, P, S \rangle$, where $V = \{S\}$, $T = \{0, 1\}$, and the productions $P$ are defined by:

$$S \rightarrow 0S1 \mid 1S0 \mid S \cdot S \mid \lambda$$

Argue that every string generated by this grammar is balanced, i.e., if $w$ is derived from $S$, then $n_0(w) = n_1(w)$.

Solution:
We use induction on the number of steps used in the shortest, leftmost derivation of $w$ from $S$.

Basis: Let $w$ be derived from $S$ in exactly one step. From the production rules, it is clear that $w$ must be $\lambda$ and hence $w$ is indeed balanced.

Inductive Step: Assume that the theorem is true for all strings $w$, whose shortest leftmost derivations from $S$, take at most $n$ steps.

Now consider the case in which the shortest leftmost derivation of $w$ from $S$ takes $n + 1$ steps, where $n \geq 1$. The first step of the derivation must be one of $S \Rightarrow SS$, $S \Rightarrow 0S1$ or $S \Rightarrow 1S0$.

Assume that the first step of the derivation is $S \Rightarrow 0S1$. It follows that $w = 0x1$, where $x$ is a string in $\Sigma^*$. Since $S \Rightarrow^* w$, we must have $S \Rightarrow^* x$; however, the shortest leftmost derivation of $x$ from $S$ can take at most $n$ steps. By the inductive hypothesis, it follows that $x$ is balanced. Consequently, $w = 0x1$ is also balanced.

An identical argument can be used for the case, in which the first step of the derivation is $S \Rightarrow 1S0$.

Finally, consider the case in which the first step of the derivation is $S \Rightarrow SS$. It follows that $w$ can be broken up into $w_1w_2$, such that $S \Rightarrow w_1$ and $S \Rightarrow w_2$. We cannot immediately apply the inductive hypothesis, since either $w_1$ or $w_2$ could be $\lambda$ and therefore the length of $w$ is not altered. However, observe that we are focussing on the shortest leftmost derivation of $w$ from $S$. If either $w_1$ or $w_2$ is $\lambda$, then we have needlessly used an extra step in the derivation and hence our derivation could not have been the shortest one. It therefore follows that neither $w_1$ nor $w_2$ is $\lambda$. Now, the shortest leftmost derivations of $w_1$ and $w_2$ from $S$ take strictly less than $n + 1$ steps; as per the inductive hypothesis, $w_1$ and $w_2$ are balanced. It therefore follows that $w = w_1 \cdot w_2$ is also balanced.

\[\square\]

2. Closure Properties of Regular Languages: Let $L_1$ and $L_2$ be two regular languages. Is the language $L_3 = L_1 \oplus L_2$ regular? Recall that given sets $A$ and $B$, the set $A \oplus B$ is defined as the set that contains elements which belong to $A$, but not to $B$ and vice versa.

Solution: The key observation is that $L_3$ can be expressed as: $(L_1 \cap L_2^c) \cup (L_1^c \cap L_2)$. Since, $L_1$ and $L_2$ are regular, so are $L_1^c$ and $L_2^c$. By using the fact that regular languages are closed under union, we infer that that $L_1 \cap L_2^c$ and $L_1^c \cap L_2$ are also regular. It immediately follows that $L_3$ is regular, since regular languages are closed under the union operation as well. \[\square\]
3. **Decision Properties of Regular Languages:** Let $L$ denote a regular language. Describe an efficient decision procedure to test whether $L = L^*$, assuming that the DFA for $L$ is provided.

**Solution:** Let $M$ denote the DFA for language $L$. Add $\lambda$-transitions from each final state of $M$ to the start state to get the $\lambda$-NFA $N$ of $L^*$. (Convince yourself that the addition of $\lambda$-transitions in the manner specified does indeed result in the $\lambda$-NFA of $L^*$.) Then, convert $N$ into a DFA $M_1$. It is straightforward to check whether the languages represented by these two DFAs are identical, using the technique discussed in class. To wit,

$$(L_1 = L_2) \leftrightarrow [(L_1 \cap L_2^c) \cup (L_1^c \cap L_2)] = \phi$$

4. **Proving or Disproving Regularity:** Let $\Sigma = \{a\}$ and let $L = \{a^{n^3}, \ n \geq 0\}$. Is $L$ regular?

**Solution:** Assume that $L$ is regular and let $m$ be the number that the Pumping Lemma associates with this language. Consider the string $w = a^{m^3}$; as per the definition of $L$, $w \in L$.

As per the Pumping Lemma, $w$ can be decomposed as $xyz$, where $|xy| \leq m$, $|y| \geq 1$ and $xy^iz \in L, \ \forall i \geq 0$. From the manner in which we have chosen $w$, it must be the case that $y$ must be of the form $a^k$, where $1 \leq k \leq m$. It is important to note that our proof should work regardless of the value of $k$, chosen by the adversary.

If the strings in $L$ were ordered by length, the string preceding $w$ would be $w' = a^{(m-1)^3}$. Let us focus on the string $w_2$ obtained by pumping down $y$, i.e., by setting $i = 0$. Observe that $|w_2| = m^3 - k$, $1 \leq k \leq n$. Regardless of the value assumed by $k$, $m^3 > |w_2| = m^3 - k > (m - 1)^3$. (Requires some nifty algebraic manipulation, but I am sure you can manage it!) But this means that $w_2 \notin L$, contradicting the assertion of the Pumping Lemma. It follows that $L$ is not regular.

5. **General questions on Regularity:**

   Let $L$ be a language over some fixed alphabet $\Sigma$.

   (a) Assume that $L$ is finite. Is it necessarily regular? Justify your answer. (2 points)

   (b) How would you efficiently test whether $L = \Sigma^*$? (2 points)

**Solution:**

(a) Finiteness implies regularity. Assume that $L$ has $n$ strings. Construct a DFA for each of those strings. We then construct a $\lambda$-NFA for $L$, by taking the union of all these individual DFAs; this involves creation of a new start state and a $\lambda$-transitiot to the start states of the each of the DFAs constructed initially. Finally, we convert the $\lambda$-NFA into a DFA using the algorithm discussed in class.

(b) Observe that $L = \Sigma^*$ if and only if $L^c = \phi$. Assume that you are given a DFA $M$ for $L$. Interchanging the final and non-final states of $M$, we get a DFA $M^c$ for $L^c$. If there exists a path from the start state of $M^c$ to a final state then $L^c$ is non-empty implying that $L \neq \Sigma^*$. Likewise, if there is no path in $M^c$ from the start state to a final state, then it must be the case that $L^c = \phi$ and hence, $L = \Sigma^*$.

□