There are **??** questions for a total of **??** points.

Let $x, y, z \in \Sigma^*$. We say that z is a shuffle of x and y if the characters in x and y can be interleaved, while maintaining their relative order within x and y, to get z. Formally, if $|x| = m$ and $|y| = n$, then $|z|$ must be $m + n$, and it should be possible to partition the set $\{1, 2, \ldots, m + n\}$ into two increasing sequences, $i_1 < i_2 < \cdots < i_m$ and $j_1 < j_2 < \cdots < j_n$, such that $z[i_k] = x[k]$ and $z[j_k] = y[k]$ for all k. Given two languages $L_1, L_2 \subseteq \Sigma^*$, define shuffle$(L_1, L_2) = \{z \in \Sigma^* \mid z$ is a shuffle of some $x \in L_1$ and some $y \in L_2\}$. Prove that the class of context-free languages is not closed under the shuffle operation. (Note that in order to do this, you need to produce two context-free languages, L_1 and L_2, such that shuffle(L_1, L_2) is not context-free.)

1. (5 points) **Solution:**

Consider the following two languages L_1, L_2 on the same alphabet a, b, c, d:

$L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ $L_2 = \{c^n d^n \mid n \in \mathbb{N}\}$

These two are context-free languages since the following grammars work:

1. $G_1 = (N, \Sigma, R, S)$ where $N = \{S\}$, $\Sigma = \{a, b, c, d\}$, $R = \{S \to epsilon, S \to aSb\}$.
2. $G_2 = (N, \Sigma, R, S)$ where $N = \{S\}$, $\Sigma = \{a, b, c, d\}$, $R = \{S \to \epsilon, S \to cSd\}$.

Let $L =$ shuffle(L_1, L_2). Note that for any string s in L, $\#a = \#b$ and $\#c = \#d$ (where $\#X$ denotes the frequency of X in s).

Now given any $p \geq 1$, consider the string $w = a^p c^p b^p d^p$ in this language. This is a valid shuffle we interleave them as follows: break $a^p b^p$ into a^p and b^p (call them part 1 and part 2), and break $c^p d^p$ into c^p and d^p (call them part 3 and part 4), and concatenate these parts in this order: part 1, part 3, part 2, part 4.

Consider any u, v, x, y, z such that $w = uvxyz, |vy| > 0, |vxy| \leq p$.

Then vxy is a substring of $a^p c^p$ or $c^p b^p$ or $b^p d^p$.

When we pump the string at least once, we see that the following can happen:

1. If it is a substring of $a^p c^p$, the frequency of at least one of a or c in the string increases, while those of b and d remain the same as before.
2. If it is a substring of $c^p b^p$, the frequency of at least one of c or b in the string increases, while those of a and d remain the same as before.
3. If it is a substring of $b^p d^p$, the frequency of at least one of b or d in the string increases, while those of a and c remain the same as before.

In all these cases, either $\#a$ is not equal to $\#b$ in the new string, or $\#c$ is not equal to $\#d$ in the new string. So the pumped string is not in the language, which implies that L is not context-free, whence we are done.

2. (5 points) Prove that the class of context-free languages is closed under shuffle with regular languages. That is, prove that if L_1 is a context-free language and L_2 is a regular language, then shuffle(L_1,L_2) is a context-free language.

Solution:

Let L_1 be a context free language and L_2 be a regular language. Let P1 be an NPDA recognizing L_1 and D2 be a DFA recognizing L_2.

Suppose P1 = (Q1, Sigma, Gamma, q10, Delta, A1) and D2 = (Q2, Sigma, delta, q20, A2).

Consider the NPDA P = (Q1 x Q2, Sigma, Gamma, (q10, q20), Delta', A1 x A2), where Delta' is defined as follows:

Delta1 = ((q1, q2), a, epsilon, (q1', q2), epsilon) — q1, q1' in Q1, q2 in Q2, a in Sigma, delta(q1, a) = q1' // corresponding to transitions in D2 Delta2 = ((q1, q2), a, B, (q1, q2'), B') — q1 in Q1, q2, q2' in Q2, a in Sigmaepsilon, B, B' in Gammaepsilon such that (q2, a, B, q2', B') in Delta // corresponding to transitions in P1 Delta' = Delta1 union Delta2

The following inductive claim is obvious enough from the description of Delta'.

Claim: Let x, y be any strings in Sigma*. Suppose for some state q1 in Q1 and some S in Gamma*, we have (q1, x, epsilon) —-*P1 (q1', epsilon, S). Suppose for some state q2 in Q2, we have deltahat(q2, y) = q2'. Then for any shuffle z of x and y, we have ((q1, q2), z, epsilon) —-*P ((q1', q2'), epsilon, S).

Now using this claim, we observe the following:

1. For any x in L_1 and y in L_2, we have (q10, x, epsilon) —-*P1 (q1acc, epsilon, S) for some q1acc in A1 and S in Gamma*, and deltahat(q20, y) = q2acc for some q2acc in A2 (by definition). So applying the above claim for any shuffle z of x and y, we get ((q10, q20), z, epsilon) —-*P ((q1acc, q2acc), epsilon, S), from where it follows that z is recognized by P.

2. For any string z accepted by P, we have ((q10, q20), z, epsilon) —-*P ((q1acc, q2acc), epsilon, S) for some S in Gamma*, q1acc in A1, a2acc in A2. Now consider a run of P on z. Any transition changes at most one 'coordinate' in the state of P. The only two kinds of transitions in such a run are as described by Delta', so partition the transitions into two sets: transitions in Delta1 and transitions not in Delta1 (and hence in Delta2 setminus Delta1, which is a subset of Delta2). Corresponding to the transitions in Delta1, there is a subsequence of the string z which is accepted by D2 (a run of the DFA can be constructed simply by reading off the corresponding 'coordinates' of each such transition, and the subsequence in particular can be extracted from reading off the inputs triggering those transitions, and this run is accepting since the last state is in A2). The remaining subsequence has transitions that are present in a subset of Delta2, and hence by a similar analysis as in the previous sentence, this subsequence of z is accepted by P1. Call the condensation of the first subsequence y and that of the second subsequence x. Hence z is a shuffle of two strings x, y, where x is in L_1 and y is in L_2 (since x is accepted by P1 and y by D2), and this shows that each string accepted by P is in shuffle(L_1, L_2).

So using these observations, we note that P recognizes shuffle(L_1, L_2), as needed.