Problem 1.

(a) Discuss how you would prove that the intersection of a context-free language and a regular language is a context-free language. Afterwards, give a written explanation using fewer than three sentences.

(b) Explain how to prove that the set of binary strings with equal numbers of 0s and 1s but containing no substring of the form 0100 or 1001 is context free. Detailed descriptions of machines are not required.

Problem 2. Review the precise statement of the pumping lemma for context-free languages.

Let $\Sigma = \{1, 2, 3, 4\}$ and L be the set of words $w \in \Sigma^*$ such that the number of 1s in w equals the number of 2s in w, and the number of 3s in w equals the number of 4s. Prove that L is not a CFL.

Problem 3. Let $\Sigma = \{0, 1\}$, and let P be the set of words that are palindromes with an equal number of 0s and 1s: $P = \{ww^R : w \in \Sigma^* \text{and the number of } 0s \text{ equals the number of 1s in } w\}$. Prove that P is not a CFL.

Solution. We prove this by contradiction using the pumping lemma for CFLs. Suppose P is context-free, and let p be a pumping length. Consider the word $w = 0^p1^p$ so that $ww^R = 0^p1^{2p}0^p$. By the pumping lemma, we can write $ww^R = uvxyz$ with $|vy| > 0$, $|vxy| \leq p$, and such that $uv^ixy^iz \in L$ for all $i \geq 0$. Label parts of the word ww^R as follows:

$$ww^R = \underbrace{0 \cdots 0}_{a}\underbrace{1 \cdots 1}_{b}\underbrace{1 \cdots 1}_{c}\underbrace{0 \cdots 0}_{d}$$

where each labeled segment has length p. Since $|vxy| \leq p$, we must have one of the following:

» vxy is contained in segment a.
» vxy is overlaps in segments a and b.
» vxy is contained in segment b.
» vxy is overlaps in segments b and c.
» vxy is contained in segment c.
» vxy is overlaps in segments c and d.
» vxy is contained in segment d.

Checking case-by-case, we see that $uv^2xy^2z \notin L$, in contradiction to the pumping lemma.

Problem 4.

(a) What is the formal definition of a Turing machine?

(b) Can a Turing machine ever write the blank symbol ␣on its tape?

(c) Can the tape alphabet Γ ever be the same as the input alphabet Σ?

(d) After executing a transition, is it ever possible that the head of a Turing machine is over the same cell it started in?

(e) Can a Turing machine contain just a single state?

PROBLEM 5. Give a high-level description with numbered steps of a Turing machine that recognizes the palindrome language of Problem 3.

Solution. Let the input be w.

1. If w is empty, accept w. In other words, if the first tape cell is blank, **accept** w.
2. Check to see if the length of the input is odd. If it is, **reject**. This rules out words like 00100 which do not have the form ww^R.
3. Otherwise, let a be the symbol under the tape head.
   - (i) Replace a with x and scan right until a blank or an x is read.
   - (ii) Move one step to the left. If the symbol under the tape head is x, **accept**.
   - (iii) If the symbol under the tape head is not a, then **reject**.
   - (iv) Otherwise, write an x and scan to the left until an x is read. Then move to the right one step, and let a be the symbol under the tape head. If $a = x$, accept. Otherwise, go to step (i).

PROBLEM 6. (Bonus) Consider a Turing machine that cannot write over its input. That is, whatever length of string that the input is on cannot be changed, but can be read as normal (and the rest of the tape can be changed as normal). Show that Turing machines of this type can recognize only regular languages.