1. Prove that the following languages are not context-free using the pumping lemma.

   (a) $L = \{a^i b^j c^m d^n \mid i = m \text{ and } j = n\}$.

   Solution: Consider the following strategy for Spoiler:

   - **Prover:** Chooses $k \geq 0$.
   - **Spoiler:** Chooses $z = a^k b^k c^k d^k$.
   - **Prover:** Chooses u, v, w, x, y, s.t $z = uvwxy$, $vx \neq \epsilon$ and $|vwx| \leq k$
   - **Spoiler:** Chooses $i = 2$, we win irrespective of choice of u, v, w, x, y.

   Proof. We divide string z into four blocks. Now v, x can be at most in two consecutive blocks because $|vwx| \leq k$. By pumping, we increase the length of these blocks making either the first and third or second and fourth block lengths unequal. □

   (b) $L = \{0^i 1^j \mid j \text{ divides } i\}$.

   Solution: Consider the following strategy :

   - **Prover:** Chooses $k \geq 0$.
   - **Spoiler:** Chooses $z = 0^{m^4} 1^{m^3}$ such that $m \gg k$.
   - **Prover:** Chooses u, v, w, x, y, s.t $z = uvwxy$, $vx \neq \epsilon$ and $|vwx| \leq k$

     We have the following cases to consider.

     - $v = 0^\ell$ and $x = 1^j$: In this case, since $\ell + j \leq k$, by choosing $i = 0$, we get the string $0^{m^4-\ell} 1^{m^3-j}$. If $m^4 - \ell = r(m^3 - j)$, then $m^3(m - r) = \ell - rj$, but this is not possible since $r < m$.
     - $v = 0^\ell$ and $x = 0^j$: In this case since $\ell + j \leq k$, we can choose $i = 0$, and we get a string $0^{m^4-\ell-j} 1^{m^3}$. Since $m^3 | m^4$ and $\ell + j \leq k \ll m$, this string is not in L.
     - $v = 1^\ell$ and $x = 1^j$: Choose i such that $m^3 + (i - 1)(\ell + j) > m^4$.

   (c) $L = \{w_1 \# w_2 \# \ldots w_k \mid k \geq 2, w_i \in \{0, 1\}^*, \text{ and } w_i = w_j \text{ for some } i \neq j\}$.

   Solution:

   - **Prover:** Chooses $k \geq 0$.
   - **Spoiler:** Chooses $z = 0^k 1^k \# 0^k 1^k$.
   - **Prover:** Chooses u, v, w, x, y, s.t $z = uvwxy$, $vx \neq \epsilon$ and $|vwx| \leq k$

- **Spoiler:** If w is such that it contains # then we are done with $i = 0$. If not, then by similar argument as first part, the new string is not in language.

2. Consider the following two languages, exactly one of which is context-free.

$$L_1 = \{w\#x \mid w, x \in \{0,1\}^*, w \text{ is a substring of } x\},$$
$$L_2 = \{w\#x \mid w, x \in \{0,1\}^*, w^R \text{ is a substring of } x\}.$$

Explain which one is context-free and which one is not.

Solution: The first is not context free, each string is of form $w\#xwy$. Now, we apply the similar pumping strategy as before with slight modifications.

- Spoiler chooses $z = 0^k1^k\#0^k1^k$.
- If v or x contains #, choose $i = 0$ then the new string has no # and hence not in Language.
- If vwx is entirely in left part or v and x are on either side and x is empty, pump such that length of left side becomes more than right.
- If otherwise, vwx entirely must be on right or x must be non-empty and on right, now choosing $i = 0$ will suffice in both cases.

The Second part is context-free by the following grammar:
$S \longrightarrow AB$
$A \longrightarrow 0A0 \mid 1A1 \mid \#B$
$B \longrightarrow 0B \mid 1B \mid \epsilon$
The strings are of form $w\#xw^Ry$.

3. Suppose that G is a grammar in Chomsky Normal Form with n non-terminals. Show that if there is a string w that has a derivation in G of length more than 2^n, then $L(G)$ is infinite.

Is the statement also true if G is not in CNF? Explain your answer.

Solution:

Claim: If we have a derivation of length $> 2^n$, then there is a path of at least $n+1$ nodes in the corresponding parse tree.

Proof. Suppose not. So, every path in the parse tree is atmost n .This means that the number of nodes can only be atmost 2^n (Complete Binary Tree). Every Derivation corresponds to atleast one node and therefore we have a contradiction.

Now by PHP, there is a non-terminal in the path which is repeated and therefore can be replaced by the bigger tree to get a longer string in the language and hence $L(G)$ is infinite. □

For the second part of the question, consider the following grammar G.

$$S \rightarrow SS \mid \varepsilon$$

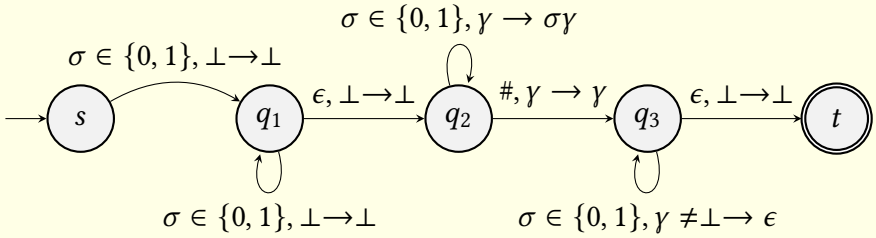
$L(G) = \{\varepsilon\}$, but there are arbitrarily long derivations for ε.

4. Prove that the following languages are context-free by describing a PDA for the same. Don't use the conversion from the grammar directly.

   (a) $L = \{w\#x \mid w \neq x, \text{ where } w, x \in \{0,1\}^*\}$.

   Solution: There are two possibilities by which $w \neq x$. One is that they are not even of the same length, i.e., $|w| \neq |x|$. The other is that they differ in some position, i.e., there is an index i where one w has a 0 and x has a 1 or vice versa. The two error cases could overlap - for example in the string 00#1, but they are together exhaustive and it suffices to solve the two cases separately and take their union.

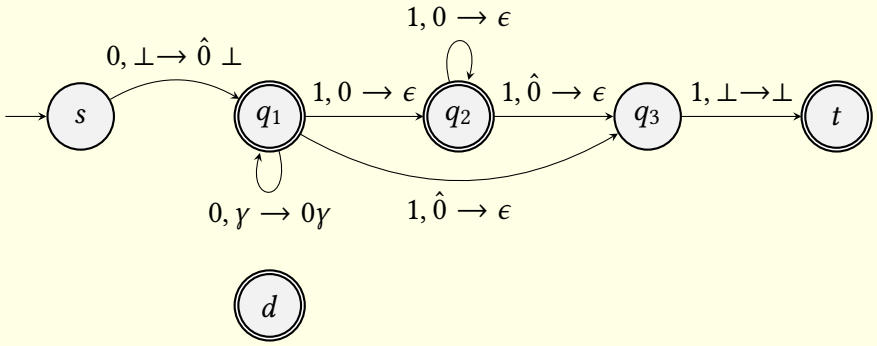
   The first case can be further broken down into two subcases - $|w| > |x|$, or $|w| < |x|$. In the first subcase, the string is of the form $\sigma w' x' \# x$, where $|x'| = |x|$, $\sigma \in \{0,1\}$, $w', x', x \in \{0,1\}^*$. The following PDA works for this subcase, and the PDA for the second subcase can also be built similarly.



   The second case can be described as non-deterministically guessing the index where w and x differ (we push symbols onto the stack until some point within w to count or keep note of the index), after which we branch into two states depending on whether the next symbol is a 0 or a 1 and ignore all symbols until the #. After the #, we use the stack to count down how many symbols to read, popping off the top each time while adding nothing. If there are any symbols left to read, the symbol is read, and if it differs from the symbol remembered by the state, we jump to the accept state and ignore every subsequent symbol. If not, we move into a dead state.

   (b) $L = \{0,1\}^* - \{0^n 1^n \mid n \geq 0\}$.

   Solution:



In order to distinguish the situation when there are equal number of 0s and 1s from the case where they differ by a count of one, we mark the first 0 we see in the string (the first symbol, if the string is a correct instance) in the stack specially, as $\hat{0}$. Almost all transitions have been marked in the PDA, besides some transitions to the dead state (if a 1 is seen at s, or if the stack becomes empty when at q_2, or any more symbols seen after reaching t) but the dead state has been drawn to show that it is an accepting state! Since all transitions are deterministic, flipping the accepting and rejecting states suffices to negate the language. Verifying that the flipped PDA accepts $\{0^n 1^n \mid n \geq 0\}$ suffices.

(c) $L = \{a^i b^j c^k \mid i \neq j \text{ or } j \neq k\}$.

Solution: L can be broken into two languages, $L_1 = \{a^i b^j c^k \mid i \neq j\}$ and $L_2 = \{a^i b^j c^k \mid j \neq k\}$. These can be solved by PDAs using an idea very similar to that of part (b), only that we also want to ensure that the string is of the form a's followed by b's followed by c's.

5. We will prove a couple of closure properties of CFLs.

(a) Show that if L is regular, and L' is context-free, then $L \cap L'$ is context-free.

Solution: Let L be regular via a DFA $M = (Q, \Sigma, \delta, s, F)$, and L' be context-free via a PDA $M' = (Q', \Sigma, \Gamma, \delta', s', \bot, F')$ that accepts via final state.

We define the PDA $M^* = (Q \times Q', \Sigma, \Gamma, \delta^*, (s, s'), F \times F')$ that simulates both automata via a product construction.

$\delta^* \subseteq (Q \times Q' \times (\Sigma \cup \{\epsilon\}) \times \Gamma) \times (Q \times Q' \times \Gamma^*)$

$\forall (q', \sigma, \gamma, q'', \beta) \in \delta', \sigma \in \Sigma, \delta^*(q, q', \sigma, \gamma) = (\delta(q, \sigma), q'', \beta)$

$\forall (q', \epsilon, \gamma, q'', \beta) \in \delta', \delta^*(q, q', \epsilon, \gamma) = (q', q'', \beta)$

Using the correctness of M, M', it can be argued that M^* accepts x iff $x \in L \cap L'$.

(b) For a language $L \subseteq \Sigma^*$, define the language $\mathsf{PALIN}(L)$ as follows:

$$\mathsf{PALIN}(L) = \{w \in L \mid w = w^R\}.$$

Prove using Part (a) that if L is regular, then $\mathsf{PALIN}(L)$ is context-free.

Solution: $\text{PALIN}(L) = \{w \in \{0,1\}^* \mid w = w^R\} \cap L$, and Part (a).

6. (**Challenge question**) This requires an understanding of how the proof of the pumping lemma works. I will not be handing out a solution for this problem. Try this only after you have worked on the other problems.

   Recall that a grammar is said to be *linear* if the right hand side of every production contains at most one non-terminal. A language L is said to be linear if there is a linear grammar G such that $L = L(G)$. We will prove a pumping lemma for linear grammars, and use it to show that prove that certain languages are not linear.

   (a) Show that if L is linear, there exits a $k > 0$ such that for every $z \in L$ such that $|z| > k$, there exists u, v, w, x, y where $z = uvwxy$, $vx \neq \varepsilon$ and $|uvxy| \leq k$ such that $\forall i \geq 0$, $uv^i wx^i y \in L$.

   (b) Use Part (a) to show that $L = \{w \in \{0,1\}^* \mid \#0(w) = \#1(w)\}$ is not linear. We have already seen in class that L is context-free.