Abhinav Jain, 2019CS10322
Arpit Chauhan, 2019CS10332
Himanshu Gaurav Singh, 2019CS10358

Problem Set - 3
COL352 - Introduction to Automata and ToC
Term: Holi 2022
Due Date: 17^{th} March, 2022

Problem 1
We say that a context-free grammar G is self-referential if for some non-terminal symbol X we have $X \rightarrow^* \alpha X \beta$, where $\alpha, \beta \neq \varepsilon$. Show that a CFG that is not self-referential is regular.

Solution: Consider any non-self-referential grammar $G =< N, \Sigma, P, S >$. Assume that the maximum length of the right hand side of any production rule is k. We will prove that no string with length greater than $k^{|N|}$ is derivable.

Lemma 1. *No string with length greater than $k^{|N|}$ is derivable by G.*

Proof. Consider any string s such that $|s| \geq k^{|N|} + 1$. Let a minimal parse tree for s be T. Observe that T has height at least $|N| + 1$ since otherwise, the number of leaves will be less than $k^{|N|}$ but the length of string is greater than this. Consider the path from the root to the deepest node in the parse tree. The path will contain h non terminals if h is the height of the tree. Since $h > |N|$, there must be two positions on the path with the same non terminal(say X). Observe that this would imply that $X \xrightarrow{*} \alpha X \beta$ where the X on the right hand side corresponds to the repeated non-terminal higher up the tree and the production corresponds to the sentential formed contained in its subtree. Observe that both α, β cannot be ϵ, otherwise the lower X could substitute the upper and still generate the same string, contradicting the minimality of T. Thus, the existence of parse tree for s implies there exists non terminal X such that $X \xrightarrow{*} \alpha X \beta$ and atleast one of α, β is non-empty contradicting that G is non-self referential. Hence, contradiction. □

From the above lemma, $L(G) \subseteq \{\Sigma + \epsilon\}^x$ where $x = k^{|N|}$. Thus, $|L(G)| \leq (|\Sigma| + 1)^{k^{|N|}}$. This implies $L(G)$ is finite and hence regular.

Problem 2
Prove that the class of context-free languages is closed under intersection with regular languages. That is, prove that if L_1 is a context-free language and L_2 is a regular language, then $L_1 \cap L_2$ is a context-free language.

Solution: Since L_1 is a context-free language, there exists a PDA $P = \{Q_1, \Sigma_1, \Gamma, \delta_1, q_1, F_1\}$ that recognizes language L_1 and since L_2 is a regular language, there exists a DFA $M = \{Q_2, \Sigma_2, \delta_2, q_2, F_2\}$ which recognizes L_2.
Now consider the PDA $P' = \{Q, \Sigma, \Gamma, \delta, q_0, F\}$ such that -

- $Q = Q_1 X Q_2$, set of states
- $\Sigma = \Sigma_1 \cap \Sigma_2$, input alphabet
- $q_0 = (q_1, q_2)$, start state
- $F = F_1 X F_2$, set of final accept states
- The transition function δ will be very similar to δ_1 with the small change that q_{PDA} will be replaced by (q_{PDA}, q_{DFA}) while reading any alphabet $'a'$ where q_{DFA} is obtained using δ_2, i.e.
  $\delta((q_{PDA}, q_{DFA}), a, A)) = \{((s_{PDA}, s_{DFA}), B) \mid (s_{PDA}, B) \in \delta_1((q_{PDA}, a, A)) \; and \; s_{DFA} = \delta_2(q_{DFA}, a)\}$
  $\delta((q_{PDA}, q_{DFA}), a, A)) = \{((s_{PDA}, q_{DFA}), B) \mid (s_{PDA}, B) \in \delta_1((q_{PDA}, \epsilon, A)) \; if \; a = \epsilon\}$

Claim 1 : For each $s \in L_1 \cap L_2$, the string s is accepted by P'.
Proof : Let the final state of the PDA P' after processing the entire string s be (f_1, f_2). Since $s \in L_1$ and δ is defined as per δ_1, we can be sure that $f_1 \in F_1$ because P accepts any string in L_1. Similarly since $s \in L_2$ also, $f_2 \in F_2$ because the 2nd element in the state tuple is obtained using δ_2 and s must be accepted by M. Therefore, $(f_1, f_2) \in F$, and hence, for each $s \in L_1 \cap L_2$, the string s is accepted by PDA P'.

Claim 2 : For each string s accepted by P', $s \in L_1 \cap L_2$.
Proof : Let the sequence of states obtained while processing s through P' be $\{(q_1, q_2), (p_1, m_1), (p_2, m_2)....(p_k, m_k), (f_1, f_2)\}$. Since P' accepts s, $f_1 \in F_1$ and $f_2 \in F_2$. Since δ is defined as per δ_1, we can get a unique set of states(by taking the 1st element of each state tuple) and the stack strings, corresponding to s being processed through PDA P and since $f_1 \in F_1$, PDA P accepts string s and thus, $s \in L_1$.
We have defined δ such that the 2nd element in the state tuple is obtained using the transition function δ_2 of the DFA M and thus, the sequence of states obtained if s is passed through M will be $\{q_2, m_1, m_2...m_k, f_2\}$. Since $f_2 \in F_2$ and the transitions are as per M, DFA M accepts string s and thus, $s \in L_2$.
Since $s \in L_1$ and $s \in L_2$, $s \in L_1 \cap L_2$.

Thus, using Claim 1 and Claim 2, we can say that $L_1 \cap L_2$ is a context-free language.

Problem 3
Given two languages L, L', denote by

$$L||L' := \{x_1y_1x_2y_2 \dots x_ny_n \mid x_1x_2 \dots x_n \in L, y_1y_2 \dots y_n \in L'\}$$

Show that if L is a CFL and L' is regular, then $L||L'$ is a CFL by constructing a PDA for $L||L'$. Is $L||L'$ a CFL if both L and L' are CFLs? Justify your answer.

Solution:
First Part

Since L is a context-free language, there exists a PDA $P = \{Q_1, \Sigma_1, \Gamma, \delta_1, q_{01}, F_1\}$ that recognizes language L and since L' is a regular language, there exists a DFA $D = \{Q_2, \Sigma_2, \delta_2, q_{12}, F_2\}$ which recognizes L'.

Now consider the PDA $P = \{Q, \Sigma, \Gamma, \delta, q_0, F\}$ such that -

- $Q = Q_1 \times Q_2 \times \{0, 1\}$, set of states
- $\Sigma = \Sigma_1 \cup \Sigma_2$, input alphabet
- $q_0 = (q_{01}, q_{02}, 0)$, start state
- $F = F_1 \times F_2 \times \{0\}$, set of final accept states
- δ_{01}= $\{((q_1, q_2, 0), a, B, (q_3, q_2, 1), B') \mid (q_1, a, B, q_3, B') \in \delta_1\}$
- δ_{02}= $\{((q_1, q_2, 0), \epsilon, B, (q_3, q_2, 0), B') \mid (q_1, \epsilon, B, q_3, B') \in \delta_1\}$
- δ_{03}= $\{((q_1, q_2, 1), a, B, (q_1, q_3, 0), B') \mid \delta_2(q_2, a) = q_3\}$
- $\delta = \delta_{01} \cup \delta_{02} \cup \delta_{03}$

Claim 1 : For each $s \in L||L'$, the string s is accepted by P'.
Proof : We will perform induction on the length of string (say $2n$) to prove the following hypothesis.

Induction Hypothesis $P(n)$: for any $x = x_1x_2 \dots x_n \in {\Sigma_1}^*$ and $y = y_1y_2 \dots y_n \in {\Sigma_2}^*$ and $s = x_1y_1x_2y_2 \dots x_ny_n$, if $\hat{\delta}(q_{02}, y) = q_2'$ and $(q_{01}, x, \epsilon) \rightarrow_P^* (q_1', \epsilon, \alpha)$ then $((q_{01}, q_{02}, 0), s, \epsilon) \rightarrow_{P'}^* ((q_1', q_2', 0), \epsilon, \alpha)$

Base Case $n = 0$, therefore $x = \epsilon$, $y = \epsilon$ and $s = \epsilon$. Therefor we have $q_2' = q_{02}$, and since $(q_{01}, \epsilon, \epsilon) \rightarrow_P (q_1', \epsilon, \alpha)$ all transitions take ϵ as input, hence all transitions are in δ_{02}. Now we follow the same transitions as followed in P, as the second element and third element in the state tuple remains the same throughout the run and thus we can reach the the configuration $((q_1', q_2', 0), \epsilon, \alpha)$ and hence $(q_{01}, q_{02}, 0), s, \epsilon) \rightarrow_{P'}^* ((q_1', q_2', 0), \epsilon, \alpha)$. Hence the base case holds.

Inductive Case For $n > 0$, let $x' = x_1x_2 \dots x_{n-1}$, $y' = y_1y_2 \dots y_{n-1}$ and $s' = x_1y_1x_2y_2 \dots x_{n-1}y_{n-1}$. There exist q_3, q_4 such that $(q_{01}, x', \epsilon) \rightarrow_P^* (q_3, \epsilon, \alpha')$ and $(q_3, x_n, \alpha') \rightarrow_P^* (q_1', \epsilon, \alpha)$ and $\hat{\delta}(q_{02}, y_1') = q_4$ and $\delta(q_4, y_n) = q_2'$. Since length of x' and y' is $n - 1$, and $P(n - 1)$ is true, so we have $((q_{01}, q_{02}, 0), s', \epsilon) \rightarrow_{P'}^* ((q_3, q_4, 0), \epsilon, \alpha')$.
Observe that during $(q_3, x_n, \alpha') \rightarrow_P^* (q_1', \epsilon, \alpha)$ we have certain number of transitions taking ϵ as the input and one transition takes x_n as input. First take all the epsilon transitions till the transition corresponding to x_n. Corresponding to these transitions we get transitions in δ_{02} where only the first element of state tuple changes. Now take the transition where x_n is the input and this results in change of 3rd element of state from 0 to 1. There is a transition corresponding to this in δ_{01}. Now take the corresponding transition in δ_{03} where y_n is read and this results in change of 3rd element of state from 1 to 0. Then again take the ϵ transitions following x_n as done previously. Therefor

we have $(q_{01}, q_{02}, 0), s, \epsilon) \rightarrow^*_{P'} ((q'_1, q'_2), \epsilon, \alpha')$. Hence $P(n)$ holds.

Let $s = x_1y_1x_2y_2 \dots x_ny_n$. Therefor we have $x = x_1x_2 \dots x_n \in L$ and $y = y_1y_2 \dots y_n \in L'$, therefore $\hat{\delta}(q_{02}, x) = q'_2$ where $q'_2 \in F_2$ and there exists a run $(q_{01}, x, \epsilon) \rightarrow^*_P (q'_1, \epsilon, \alpha)$ where $q'_1 \in F_1$. Using the above inductive claim we have$(q_{01}, q_{02}, 0), s, \epsilon) \rightarrow^*_{P'} ((q'_1, q'_2, 0), \epsilon, \alpha)$. Observe that $(q'_1, q'_2, 0) \in F$ as $q'_1 \in F_1$ and $q'_2 \in F_2$. Therefore s is accepted by P', and hence $s \in L(P')$. Hence the claim holds.

Claim 2 : For each $s \in L(P')$, then $s \in L||L'$.
Proof : We will perform induction on number of transitions of P' to prove the following hypothesis.

Induction Hypothesis $P(n)$: for any $x = x_1x_2 \dots x_n \in {\Sigma_1}^*$ and $y = y_1y_2 \dots y_n \in {\Sigma_2}^*$ and $s = x_1y_1x_2y_2 \dots x_ny_n$, if $((q_{01}, q_{02}, 0), s, \epsilon) \rightarrow^*_{P'} ((q'_1, q'_2, 0), \epsilon, \alpha)$ then $\hat{\delta}(q_{02}, y) = q'_2$ and $(q_{01}, x, \epsilon) \rightarrow^*_P (q'_1, \epsilon, \alpha)$.

Base Case $n = 0$, therefore $x = \epsilon$, $y = \epsilon$ and $s = \epsilon$. Therefor we have $q'_2 = q_{02}$ and hence $\hat{\delta}(q_{02}, y) = q'_2$. Since $(q_{01}, \epsilon, \epsilon) \rightarrow_{P'} (q'_1, \epsilon, \alpha)$ all transitions take ϵ as input, hence all transitions are in δ_{02}. Now we follow the same transitions as followed in P', as the second element and third element in the state tuple remains the same throughout the run and thus we can reach the configuration (q'_1, ϵ, α) and hence $(q_{01}, x, \epsilon) \rightarrow^*_P (q'_1, \epsilon, \alpha)$. Hence the base case holds.

Inductive Case For $n > 0$, let $x' = x_1x_2 \dots x_{n-1}$, $y' = y_1y_2 \dots y_{n-1}$ and $s' = x_1y_1x_2y_2 \dots x_{n-1}y_{n-1}$. Since $P(n-1)$ holds there exist q_3, q_4 such that $((q_{01}, q_{02}, 0), s', \epsilon) \rightarrow^*_{P'} ((q_3, q_4, 0), \epsilon, \alpha')$ and $(q_{01}, x', \epsilon) \rightarrow^*_P (q_3, \epsilon, \alpha')$ and $\hat{\delta}(q_{02}, y') = q_4$. Let $((q_{01}, q_{02}, 0), s, \epsilon) \rightarrow^*_{P'} ((q'_1, q'_2, 0), \epsilon, \alpha')$

Observe that during $((q_3, q_4, 0), x_n, \epsilon) \rightarrow^*_{P'} ((q''_1, q_4, 1), \epsilon, \alpha')$, we have certain number of transitions taking ϵ as the input and one transition takes x_n as input. First all the epsilon transitions are taken till the transition corresponding to x_n. Corresponding to these transitions we get transitions in δ_{02} where only the first element of state tuple changes. Then the transition where x_n is the input is taken and this results in change of 3rd element of state from 0 to 1. There is a transition corresponding to this in δ_{01}. Now observe that during $((q''_1, q_4, 1), y_n, \alpha') \rightarrow^*_{P'} ((q''_1, q'_2, 0), \epsilon, \alpha')$, y_n is read corresponding to the transition in δ_{03} and this results in change of 3rd element of state from 1 to 0. Finally we have $((q''_1, q'_2, 0), \epsilon, \alpha') \rightarrow^*_{P'} ((q'_1, q'_2, 0), \epsilon, \alpha')$,here we take the ϵ transitions in δ_{02}, following x_n . Therefor we have $(q_3, x_n, \alpha') \rightarrow^*_P (q'_1, \epsilon, \alpha)$ and $\delta(q_4, y_n) = q'_2$. Since by induction hypothesis we have $(q_{01}, x', \epsilon) \rightarrow^*_P (q_3, \epsilon, \alpha')$ and $\hat{\delta}(q_{02}, y') = q_4$ therefor we have $(q_{01}, x, \epsilon) \rightarrow^*_P (q'_1, \epsilon, \alpha')$ and $\hat{\delta}(q_{02}, y) = q'_2$. Hence $P(n)$ holds.

Let $s = x_1y_1x_2y_2 \dots x_ny_n$. Since s is accepted by P', there exists a run of P' such that $(q_{01}, q_{02}, 0), s, \epsilon) \rightarrow^*_{P'} ((q'_1, q'_2, 0), \epsilon, \alpha)$ where $q'_1 \in F_1$ and $q'_2 \in F_2$. Using the above inductive claim we have $\hat{\delta}(q_{02}, x) = q'_2$ and there exists a run $(q_{01}, x, \epsilon) \rightarrow^*_P (q'_1, \epsilon, \alpha)$. Since $q'_2 \in F_2$ and $q'_1 \in F_1$, therfor $x \in L$ and $y \in L'$. Hence $s \in L||L'$ and the claim holds.

Using the above 2 claims we conclude that $L(P') = L||L'$. Hence $L||L'$ is a CFL.

Second Part

We will give L and L' such that both of them are CFL, but $L||L'$ is not a CFL.
Consider $L = \{0^n1^{2n}\}$ and $L' = \{1^{2n}0^n\}$. Observe that L is CFL, as it is the language of the grammar defined by the production rules $S \rightarrow 0S11 \mid \epsilon$.
Similarly L' is CFL, as it is the language of the grammar defined by the production rules $S \rightarrow 11S0 \mid \epsilon$.

We will use pumping lemma to show that $L||L'$ is not a CFL. Let p be the pumping length. Let $x = 0^p1^{2p}$ and $y = 1^{2p}0^p$. Consider the string $s \in L||L'$ formed by interleaving x and y. Observe

that s is equal to $(01)^p(11)^p(10)^p$.

The pumping lemma for context-free languages states that s can be divided into five pieces $s = uvxyz$ such that $|vy| > 0$, $|vxy| \leq p$ and $uv^ixy^iz \in A$ for each $i \geq 0$. Now the following possibilities exist depending upon the choice of vxy from the string s -

- **Case 1:** vxy lies completely in $w = (11)^p$, i.e. the mid of the string s. Consider i=0 i.e the pumped down string uxz. Observe that since v and y contain only 1's, hence there will be more 0's than 1's in the the pumped down string, hence it can't be of form $(01)^k(11)^k(10)^k$ for some $k > 0$.
- **Case 2:** Observe that since $vxy \leq p$ it can't lie in all the 3 halves of s as $|s| = 3p$. WLOG consider the case when vxy lies in the first two thirds of the string s (other case is symmetric). Now if we consider i=2, uv^2xy^2z will contain more $01's$ than $10's$ as the number of $10's$ has not changed. Hence uv^2xy^2z can't be of form $(01)^k(11)^k(10)^k$ for some $k > 0$.

We have shown that all possible cases contradict the pumping lemma for context-free languages. Hence our assumption of $L||L'$ being context-free is wrong and therefore, $L||L'$ is not a CFL.

Problem 4
For $A \subseteq \Sigma^*$, define

$$cycle(A) = \{yx \mid xy \in A\}$$

For example if $A = \{aaabc\}$, then

$$cycle(A) = \{aaabc, aabca, abcaa, bcaaa, caaab\}$$

Show that if A is a CFL then so is $cycle(A)$

Solution:
We construct a PDA P_C for $cycle(A)$ from a PDA P of A which must exist since A is a CFL.

Assume that P accepts by empty stack. For any xy accepted by P, reading of x by P leaves it in a configuration (q, y, τ) which is then turned to (p, ϵ, ϵ) for some p after y passes through P. We construct P_C such that it non deterministically guesses the stack configuration τ and state q above, reads y and checks if it leads to (p, ϵ, ϵ) and then checks if x indeed could generate that stack configuration and state as guessed, hence accepting/rejecting yx iff xy is accepted by P.

With $P =< Q, \Sigma, T, \delta, q_0, \perp >$. (There is no F since P accepts by empty stack), we construct

$$P_C =< Q_1 \cup Q_2, \Sigma, T \cup \{\$\} \cup T', \delta_C, q_{0_1}, \perp >, where$$
$$Q_1 = \{q' : q \in Q\}$$
$$\mathrm{Q_2} = \{(q'', \tau) : q \in Q, \tau \in T\}$$
$$T' = \{\tau' : \tau \in T\}$$

The transition function will be non-deterministic and is as described below:

1. Initially, pop $\perp$ and push "\$ $\perp$" back into the stack with ϵ being read from the string. The initial state in the automaton is $q'_0 \in Q_1$.
2. Whenever, the topmost character on the stack is \$ and the state on the automata belongs to Q_1, non-deterministically pop \$ and push $X\$X'$ into the stack for all $X \in T$ without reading any character from the string. The state on the automata does not change.
3. If the topmost character is not \$ and the state on the automata belongs to Q_1, then follow the transitions same as P.
4. Also, in the condition same as 2(topmost character is \$), carry out state transition $(q''_0, \perp) \in Q_2$ and push $\perp$ \$ on the stack without consuming any character from the string.
5. If the state on the automata belongs to $Q_2 = (q'', \tau)$ and the topmost character on the stack is \$ and the transition in P is $\delta(q, a, \tau) = (q_1, \tau_1\tau_2..\tau_k)$ where a is the current character, then pop some(possibly none) elements from the top of the stack(after popping \$) that form a prefix of $\tau'_k\tau'_2..\tau'_1$(say $\tau'_k\tau'_2..\tau'_{j+1}$), push \$ back and then push the remaining suffix of $\tau_1\tau_2..\tau_k(\tau'_1\tau'_2..\tau'_j)$ above \$. The length of the prefix to be pushed before \$ is chosen non-deterministically. The state on the automata transitions to (q''_1, τ_1).
6. If the state on the automata belongs to Q_2 and the topmost character is not \$, then perform transitions same as P with the corresponding states in Q_2 such that $\delta_P(q, a, \tau) = (q_1, \tau_1\tau_2..\tau_k) \implies \delta_{PC}((q, \tau), a, \tau) = ((q_1, \tau_1), \tau_1\tau_2..\tau_k)$.
7. If the string is completely consumed and the stack contains just the elements $\perp$ \$ and the automata is in some state belonging to Q_2, then empty the stack and accept the string.

With this description of P_C, now we will prove that P_C accepts $cycle(A)$.

Lemma 2. $cycle(A) \subseteq L(P_C)$

Proof. Consider any yx such that $xy \in L(P)$. Assume that after the consumption of x by P, the configuration is $(q, y, \tau_1\tau_2..\tau_k)$. Consider a run of P_C in which we read \$, push $\tau_i\$\tau_i'$ on the stack, read τ_i from the stack and consume some(or none, as determined by δ_P) of y and then read \$ again as i goes from 1 to k. Since in P, the configuration $(q, y, \tau_1\tau_2..\tau_k)$ reduces to (q', ϵ, ϵ), the run for P_C as mentioned above will consume y completely and the state of the stack will be $\$\tau_k'\tau_{k-1}'..\tau_1'$. Now P_C can transition to the state in Q_2(according to rule 4). Observe that using rule 5,6,7, we backward simulate the runs of the PDA that could have generated the configuration at the transition in rule 4. Since, the stack configuration $\tau_1\tau_{k-1}..\tau_k$ is achieved when starting from the start state and consuming x, when P_C consumes x, the stack will only contain \$ $\perp$ which leads to acceptance of the string from rule 7. Thus, $cycle(A) \subseteq L(P_C)$. □

Lemma 3. $L(P_C) \subseteq cycle(A)$

Proof. Consider any string accepted by P_C. Since we start in a state belonging to Q_1 and the string is accepted in a state belonging to Q_2. We must have transitioned from some state in Q_1 to one in Q_2 through rule 4. Consider the remaining portion of the string not yet consumed at that instant be x and the consumend portion of the string be y. We will prove that $xy \in A$. Consider the state of the stack after y has been consumed by P_C. Since finally the string is accepted, from rule 4,5,6 we can conclude that the stack reaches the same corresponding state in P from the starting state when run on x. Also, since for the application of rule 4, the topmost symbol has to be \$, we conclude that the corresponding state of the stack in P would have been emptied when run on y. Hence, the run on xy would end with an empty stack on P and hence will be accepted. Thus, $L(P_C) \subseteq cycle(A)$. □

From the above two lemmas, we can conclude that $L(P_C) = cycle(A)$. Thus, there exists a PDA accepting $cycle(A)$ and hence $cycle(A)$ is a CFL.

Problem 5
Let

$$A = \{wtw^R \mid w, t, \in \{0,1\}^* \text{ and } |w| = |t|\}$$

Show that A is not a CFL.

Solution: We will show this using Contradiction.
Let A be a context-free language with p as the pumping length. Consider the string $s = 1^{2p}0^p1^p1^{2p}$. Here, $w = 1^{2p}$ and $t = 0^p1^p$. Since |w| = |t|, $s = wtw^R \in A$.
The pumping lemma for context-free languages states that s can be divided into five pieces $s = uvxyz$ such that $|vy| > 0$, $|vxy| \leq p$ and $uv^ixy^iz \in A$ for each $i \geq 0$. Now the following possibilities exist depending upon the choice of vxy from the string s -

- **Case 1:** vxy lies completely in $w = 1^{2p}$, i.e. the first third of the string s. Now if we choose i such that the total length of the string becomes larger than 12p, then the last third would have 0s but not the first third, which violates the defined language. (Entire $0^p1^p1^{2p}$ would go to the last third because each third would have size larger than 4p).

- **Case 2:** vxy lies completely in the last two thirds of the string s. Now if we consider i=2, uv^2xy^2z will contain 0s in the first third but not in the last third because the last 3p characters will still be 1 and the last third will be a subset of that while the additional characters entering the first third would be 0s (Since $|vxy| \leq p$, $|uv^2xy^2z| \leq 6p + p = 7p$ and so the last third would have size less than 3p). Thus, $uv^2xy^2z \notin A$.

- **Case 3:** vxy lies at the intersection of the first third with the last two thirds. Note that since each third is of size 2p, vxy will lie within the first half of s. Hence, $|uv^2xy^2z| \leq 6p + p = 7p$ and thus the last third would consist solely of 1s.

  - If v contains both 0s and 1s, then the $(2p+1)^{th}$ character will still be 0 and thus, uv^2xy^2z will have 0 in the first third but not in the last third. Similarly, if $v = \epsilon$, but y contains both 0s and 1s, then also uv^2xy^2z will have 0 in the first third but not in the last third. Thus, $uv^2xy^2z \notin A$.
  - If v contains only 1s and y contains anything, then I can always pump v such that the first third would consist only of 1s and the last third would contain 0s. We can do similarly for the case when v is empty and y contains only 1. Thus, there exists i for which $uv^ixy^iz \notin A$.(for example i=3p+1, last third will be of size 4p and would contain $0^p1^p1^{2p}$ while the first third will be 1^{4p}, hence cannot be in A)

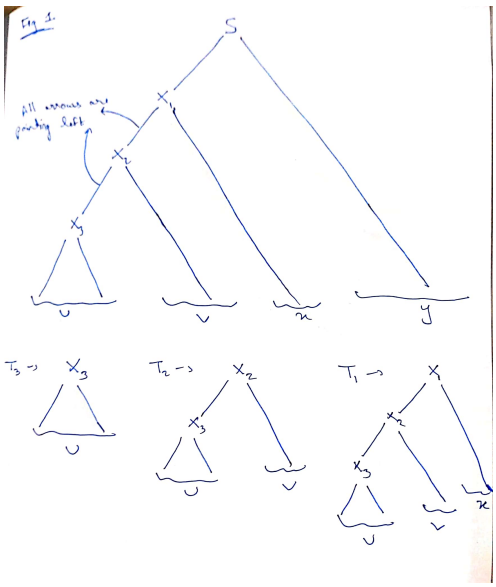
We have shown that all possible cases contradict the pumping lemma for context-free languages. Hence our assumption of A being context-free is wrong and therefore, A is not a CFL.

Problem 6
Prove the following stronger version of pumping lemma for CFLs: If A is a CFL, then there is a number k where if s is any string in A of length at least k then s may be divided into five pieces $s = uvxyz$, satisfying the conditions:

- for each $i \geq 0$, $uv^ixy^iz \in A$
- $v \neq \varepsilon$, and $y \neq \varepsilon$, and
- $|vxy| \leq k$.

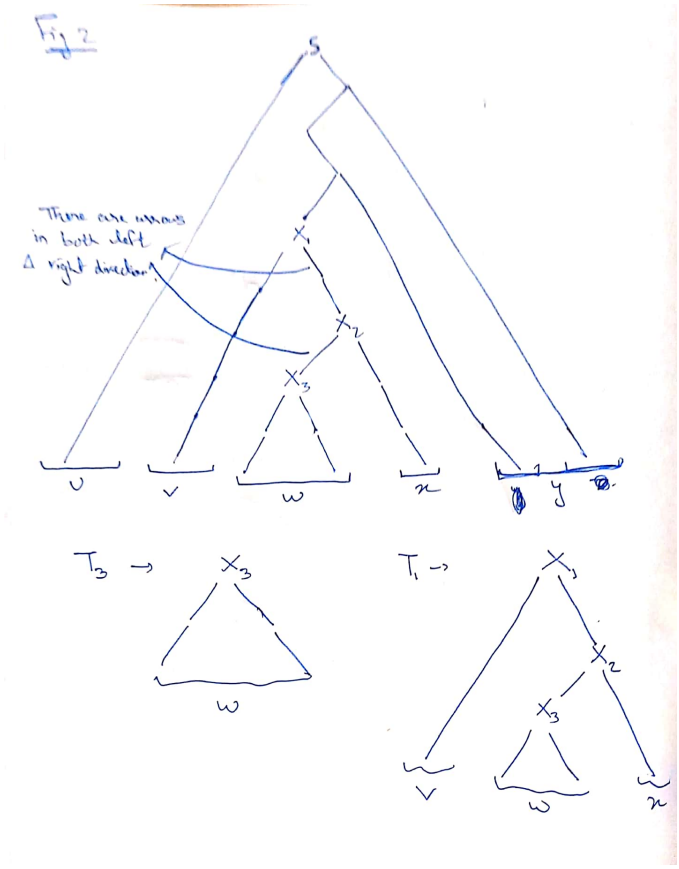
Solution: We will give a proof similar to weaker version of pumping lemma given in the course textbook (D. Kozen). Let G be a grammar for A in Chomsky normal form. Suppose there are n non-terminals in G. Let $k = 2^{3*n+1}$. Consider a string $z \in A$ such that $|z| \geq k$. (If no such string z exists, then pumping lemma is trivially true). Observe that the depth of any parse tree in G for z must be of depth at least $3 * n + 1$. This is because symbols at each level of parse tree is at most double the previous level as G is in Chomsky normal form. Consider a longest path P in the parse tree. P is of length at least $3n + 1$, hence it contains at least $3 * n + 1$ non-terminals. By pigeon hole principle there exists a non-terminal say X that occurs at least 3 times in P. Consider the first 3 occurrences of X along P, reading from bottom to top (say X_1, X_2 and X_3). Now we consider 2 cases :

- **Case 1:** Path from X_1 to X_3 unfolds only in direction i.e all the arrows are either only left pointing or right pointing. WLOG all the arrows are left pointing (see fig 1). In this we break z into substrings $uvwxy$,such that $w = \epsilon$, uvx is the string generated by X_1, uv is generated by X_2 and u is generated by X_1. Let T_1 be the subtree rooted at X_1, T_2 be the subtree rooted at X_2 and T_3 be the subtree rooted at X_3. Observe that we can cut out T_2 and replace it with a copy of T_1 as many times (even 0 times) as we want to get a valid parse for $uvwx^iy$ where $i \geq 0$. Now we can cut out T_3 and replace it i copies of T_1 to get a valid parse for uv^iwx^iy where $i \geq 0$.Observe that both v and y are non-empty as there are no derivations which lead to ϵ as G is in Chomsky normal form. Further it is clear that $|vxy| \leq k$ since we chose the repeated occurrence of non terminals from bottom and such a repetition must occur within height of $3n + 1$ and hence it can't contain more than k non-terminals.



- **Case 2:** Path from X_1 to X_3 contains both left and right pointing arrows.(see fig 2). In this we break z into substrings $uvwxy$ such vwx is the string generated by X_1 and w is generated by X_3. Let T_1 be the subtree rooted at X_1 and T_3 be the subtree rooted at X_3. Observe that we can cut out T_3 and replace it with a copy of T_1 as many times (even 0 times) as we want

to get a valid parse for uv^iwx^iy where $i \geq 0$. Observe that both v and y are non-empty as X_3 lies completely inside T and not on the boundary. Further it is clear that $|vxy| \leq k$ since we chose the repeated occurrence of non terminals from bottom and such a repetition must occur within height of $3n + 1$ and hence it can't contain more than k non-terminals.



Problem 7
Give an example of a language that is not a CFL but nevertheless acts like a CFL in the pumping lemma for CFL (Recall we saw such an example in class while studying pumping lemma for regular languages).

Solution: Consider the language $A = \{a^i b^j c^k d^l \mid if\ i = 1\ then\ j = k = l\ otherwise\ j, k, l \geq 0\}$ and $B = ab^*c^*d^*$. B is a regular language because we can easily construct an NFA with 4 states which recognizes B. Now consider $C = A \cap B = \{ab^n c^n d^n \mid n \geq 0\}$.

Claim 1: C is not context-free.
Proof: Let us assume that C is context-free with pumping length p. Consider the string $s = ab^p c^p d^p$. We break it into five parts as s = uvxyz. Pumping lemma requires that atleast one of v and y must be non-empty. Let us assume that v is non-empty.
If v consists of only a, then the string uv^ixy^iz will not be in C for i=0 or $i \geq 2$.
If v consists of only b or only c or only d, then all the three characters cannot be made equal in frequency for all values of i while also being in C (because then y^i will have to account for the required increase in frequency of both the other characters, but in that case uv^ixy^iz will not be of the form $ab^n c^n d^n$).
If v consists of two characters, then for i>1, uv^ixy^iz will not remain of the form $ab^n c^n d^n$. Similarly we can say for y.
v cannot have more than 2 distinct characters because $|vxy| \leq p$. Therefore, we have a contradiction to the pumping lemma for context-free languages, and thus, C is not context-free.

Claim 2: A is not context-free.
Proof: Using problem 2 of this problem sheet, we can say that if A were to be context-free, it's intersection with regular language B should have been context-free which is not the case as per Claim 1, and thus, by contradiction we can conclude that A is not context-free.

Claim 3: A satisfies pumping lemma for context-free languages
Proof: We will show that A satisfies the pumping lemma for pumping length p = 2. Consider any string $s \in A$ with length atleast p, now depending upon the frequency of a, we have the following three cases(s broken into $uvxyz$) -

- s does not contain a. s will contain atleast one of {b,c,d}, let it be equal to v and let $x = y = \epsilon$. Then, uv^ixy^iz will belong to A for all $i \geq 0$.
- s contains one a. Consider v=a and $x = y = \epsilon$. Then, uv^ixy^iz will belong to A for all $i \geq 0$.
- s contains more than one a. Consider v=aa and $x = y = \epsilon$. Then also, uv^ixy^iz will belong to A for all $i \geq 0$.

We observe that in all the three cases, pumping lemma is satisfied and thus, we can conlcude that A satisfies pumping lemma for context-free languages.

Using Claim 2 and Claim 3, we can say that A is a valid example of a language that is not a CFL but nevertheless acts like a CFL in the pumping lemma for CFL.
