1. **Use the pumping lemma to prove that the language**

$A = \{ 0^{2n} 1^{3n} 0^n \mid n \geq 0 \}$ is not

is not context free.

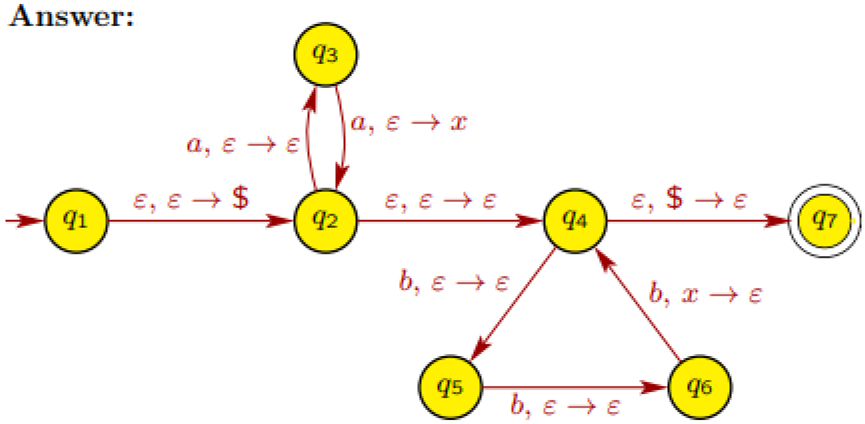
Answer: Assume that A is a CFL. Let p be the pumping length of the pumping lemma for CFLs, and consider string $s = 0^{2p}1^{3p}0^p \in A$. Note that $|s| = 6p > p$, so the pumping lemma will hold. Thus, there exist strings u, v, x, y, z such that $s = uvxyz = 0^{2p}1^{3p}0^p$, $uv^ixy^iz \in A$ for all $i \geq 0$, and $|vy| \geq 1$. We now consider all of the possible choices for v and y:

- Suppose strings v and y are uniform (e.g., $v = 0^j$ for some $j \geq 0$, and $y = 1^k$ for some $k \geq 0$). Then $|vy| \geq 1$ implies that $j \geq 1$ or $k \geq 1$ (or both), so uv^2xy^2z won't have the correct number of 0's at the beginning, 1's in the middle, and 0's at the end. Hence, $uv^2xy^2z \notin A$.
- Now suppose strings v and y are not both uniform. Then uv^2xy^2z will not have the form $0 \cdots 01 \cdots 10 \cdots 0$. Hence, $uv^2xy^2z \notin A$.

Thus, there are no options for v and y such that $uv^ixy^iz \in A$ for all $i \geq 0$. This is a

$F = \{ a^{2n}b^{3n} \mid n > 0 \}$

Answer:



Problem 2: Show that Kruskal's algorithms is in class P.

Solution: These algorithms were developed by Joseph Kruskal. Kruskal algorithms create a minimum spanning tree T by adding the edges one at a time to T. A minimum cost spanning tree is built edge by edge. We start with the edge of minimum cost. However, if there are several edges with the same minimum cost, then we select one of them and add it to the spanning tree T provided its addition does not form a cycle. We then add with the next lower cost, and so on. We repeat this process until we have selected N–1 edges to form the complete minimum cost spanning tree. This algorithm selects the edges for addition in the minimum spanning tree in the increasing order of their cost. We have to remember here that the edges can only be added if it does not form a cycle. Let us take the following graph to find out the MST using Kruskal algorithms. We can organize the whole process in a tabular form with three rows and the number of columns equal to the number of edges. The first row contains the edges in the descending order of their cost, the second row contains the cost and the third contains A if the corresponding edge is added. This can be shown for the fiven graph as shown in Figure 8.7.

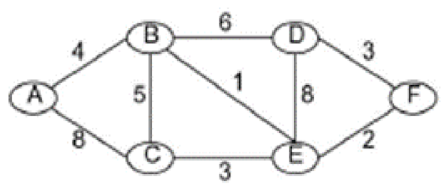


Fig. 8.7 *Simple Weighted Graph*

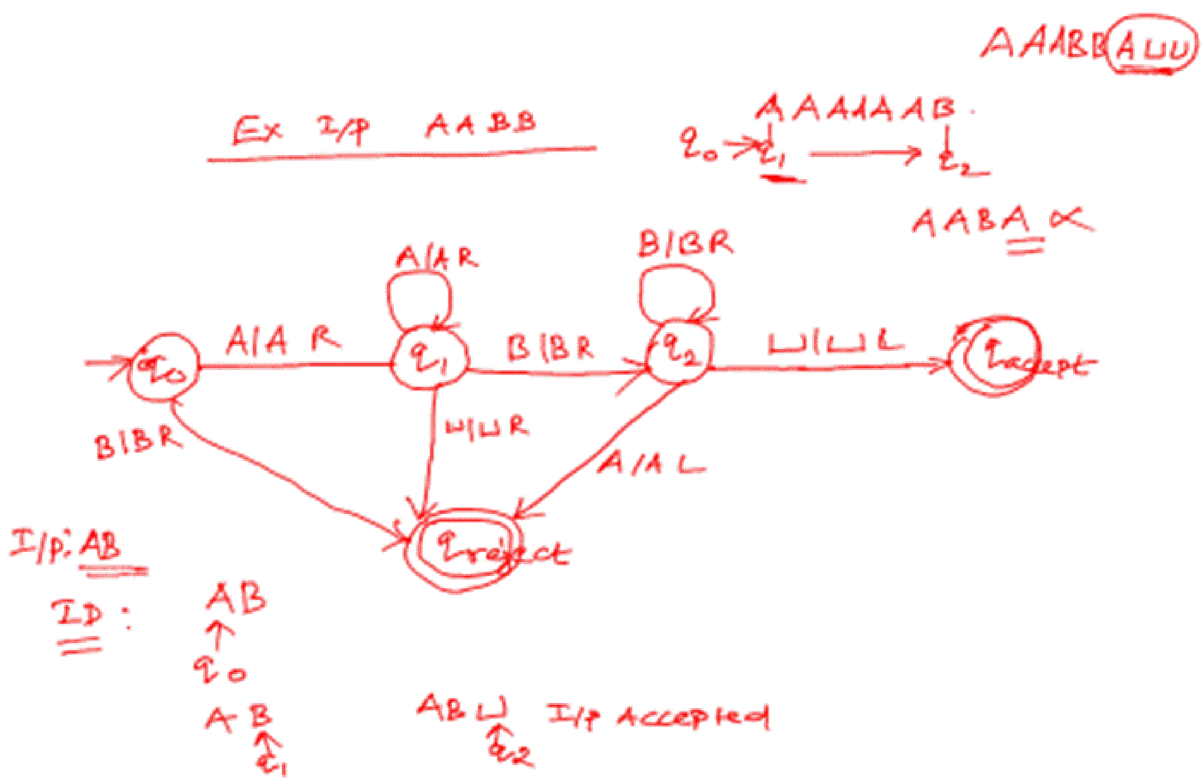
Edges	EB	EF	DF	CE	AB	BC	BD	DE	AC
Cost	1	2	3	3	4	5	6	8	8
Add	A	A	A	A	A				

The complexity of finding the MST using prims's method id $O(n^2)$, where n is number of vertices, for Kruskal method, the complexity is O(e2), where e is the number of edges. So, the problem of finding the MST belongs to class P.

Problem 3: Show that the satisfiability problem is in class NP.

Solution: The Boolean Satisfiability problem (SAT) is a decision problem considered in complexity theory. Suppose we have a Boolean expression which is made up of the variables (x_1, x_2, x_3 … x_n), parentheses and Boolean operators $\vee$, $\wedge$ and $\neg$ where these operators are for logical OR, AND and NOT respectively. A truth assignment for a Boolean expression depends on the values to the variables so that the whole expression is true. Now the question that arises is as follows. Do there exist values of the logical variables (x_1, x_2, x_3… x_n) to make a given Boolean expression true?. Thus SAT is used to determine whether there exists a true or false assignment to the variables such that all clauses are evaluated to be true making entire expression true? The Boolean expression is said to be satisfied if truth values can be assigned to its free variables in such a way that the formula becomes true. SAT clearly belongs to the complexity class NP because we can guess a truth assignment and verify that it satisfies the Boolean expression in polynomial time.

4. Construct a Turing machine that accepts all input in the following format: number of A's (at least 1) followed by number of B's (at least 1). Draw the transition diagram; write the instantaneous description and the transition function. Also give the tuple Notation for the designed TM.



5. Construct a Turing machine to perform function $f(x) = x^2$

The input tape is in the form 1^X. It has to be made in the form $1^X B 1^X$. To make this, the transitional functions are

$\delta(q_0, B) \rightarrow (q_1, X, R)$ // replace the first '1' by X.
$\delta(q_1, 1) \rightarrow (q_1, 1, R)$ // traverse right to find 'B'
$\delta(q_1, B) \rightarrow (q_2, B, R)$
$\delta(q_2, 1) \rightarrow (q_2, 1, R)$ // need from the second traversal
$\delta(q_2, B) \rightarrow (q_3, 1, L)$ // replace one 'B' after the end marker 'B' by '1'
$\delta(q_3, 1) \rightarrow (q_3, 1, L)$ // need from the second traversal
$\delta(q_3, B) \rightarrow (q_4, B, L)$ // traverse left
$\delta(q_4, 1) \rightarrow (q_4, 1, L)$ // traverse left to find the replaced X
$\delta(q_4, X) \rightarrow (q_0, X, R)$
$\delta(q_0, B) \rightarrow (q_5, B, L)$ // if all the 1 are replaced by 'X'
$\delta(q_5, X) \rightarrow (q_5, 1, L)$ // replace all 'X' by '1'
$\delta(q_5, B) \rightarrow$ enter the Turing machine for multiplication

6. Design a Turing Machine to compute 1's complement

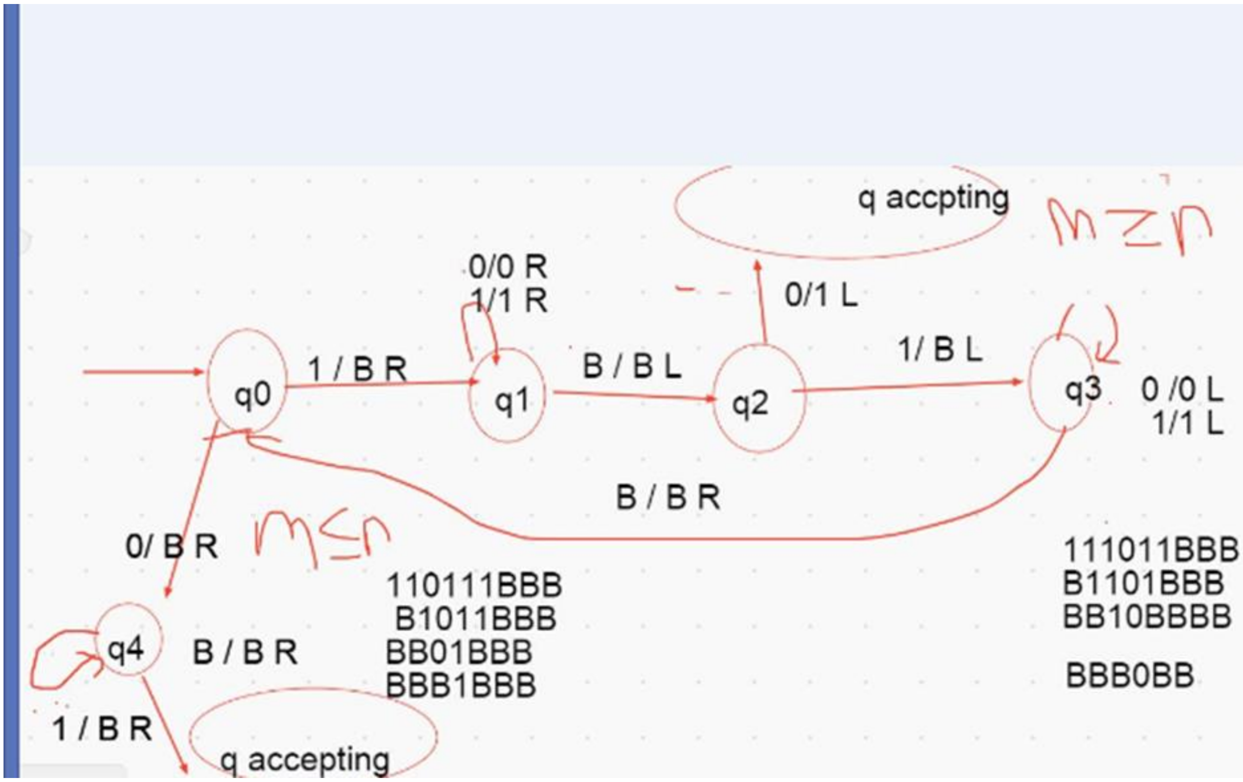
Approach:

1. **Scanning input string from left to right**
2. **Converting 1's into 0's**
3. **Converting 0's into 1's**
4. **Move the head to the start when BLANK is reached.**

Steps:

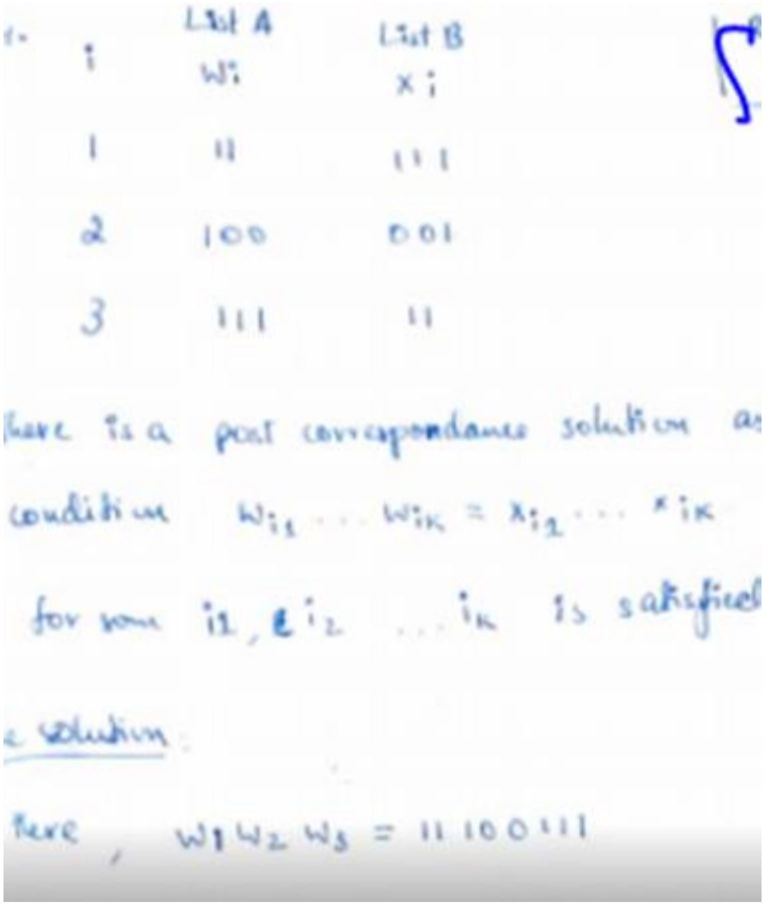
- **Step-1. Convert all 0's into 1's and all 1's into 0's and go right if B found go to left.**
- **Step-2. Then ignore 0's and 1's and go left & if B found go to right**
- **Step-3. Stop the machine.**

7. Smriti brought 8 boxes of sweets. Each box contains 2 sweets. How many sweets would be left with her after giving 10 sweets to friends? Help smriti by designing a Turing machine to find the remaining amount of sweets



7. Show that the following post correspondence problem has a solution and if so give the solution.

i	**List A** **Wi**	**List B** **Xi**
1	**11**	**111**
2	**100**	**001**
3	**111**	**11**



i	List A w_i	List B x_i
1	11	111
2	100	001
3	111	11

here is a post correspondance solution as condition $w_{i_1} \dots w_{i_k} = x_{i_1} \dots x_{i_k}$ for some $i_1, \& \, i_2 \dots i_k$ is satisfied

e solution:

here, $w_1 w_2 w_3 = 11100111$

8. .Do phrase structure analysis of the following sentences

The dog saw a man in the park