1.Give a CFG for:

L = {x ∈ {0, 1}∗ | symbol at position i is same as symbol at position i+2 and | x |≥ 2}

2. Give a CFG for the language of all non-palindromes over {0, 1}∗.

3. Is this grammar ambiguous? If so, prove it and construct a non-ambiguous grammar that derives the same language.

S -> aS |aSbS| c

1.

S 🡪A |B | C | D

A 🡪00A | 00

B 🡪11B | 11

C 🡪10C | 10

D 🡪01D | 01

2.

S🡪 0S0 | 1S1 | D

D 🡪1A0 | 0A1

A🡪 ε | 0A | 1A

3. It is ambiguous! aacbc has two parse trees (not pictured, but you have to show the two parse trees to prove it is ambiguous).

Unambiguous grammar:

S 🡪T | U

T 🡪 aTbT | c

U 🡪aS | aTbU

For the grammar S -> aS | aSbS | c, a parse tree for the string "aab" would be a tree with 'S' at the root, branching down to 'a', then another 'S', then 'a', then 'b', then 'S', and finally 'c'.

Here's a breakdown of how to construct the parse tree:

1. **Start with the input string:** "aab"
2. **Start with the start symbol:** S
3. **Apply the grammar rules:**
   * S -> aS (Apply this to the first 'a')
   * S -> aSbS (Apply this to the second 'a')
   * S -> c (Apply this to the 'c')
4. **Build the tree:**
   * **Root:** S
   * **Left Child:** a
   * **Right Child:** S
   * **Left Child of the right S:** a
   * **Right Child of the right S:** S
   * **Left Child of the right S:** b
   * **Right Child of the right S:** S
   * **Left Child of the right S:** c
   * **Leaves:** a, b, c

