Solⁿ 1: We will need to prove Soundness & Completeness

Soundness: → Every string generated by this grammar has the same number of a's and b's.

Induction over the length N of the derivation.

Base Case: $N = 1$. The only string derivable with a derivation of length $N = 1$ is ϵ $(S \to \epsilon)$: it trivially holds that the number of a's and b's in ϵ is the same.

Inductive Case: Assume that all the strings w derivable from S with derivation length $\leq N$ have an equal # of a's & b's.

Consider a string w' derivable from S with a derivation of length $N+1$. The first production rule used in the derivation that yields w' is $S \to aSbS$ or $S \to bSaS$. The remaining N productions can be split in 2 derivations: one that generate w_1 from first S & another that generates w_2 from second S. By induction, in both w_1 & w_2, the number of a's is equal to the number of b's. Since w is either aw_1bw_2 or bw_1aw_2 ⇒ Number of a's in w is equal to the number of b's.

Completeness: → Any string w with an equal number of a's and b's can be derived from S.

Induction over length ~~of derivation~~ of w:

Base Case: $|w| = 0$, $w = \epsilon$.

w can be desired from S : $S \to \epsilon$

Inductive Case: Assume that all the strings w with an equal number of a's and b's and of length upto $2n$ are desirable from S.

Consider now a string w' with an equal number of a's and b's of length $2(n+1)$. W.l.o.g. assume that the first symbol in w' is an a.

Let $2 \leq j \leq 2n+2$ be the first index such that the substring of w' from position 1 to position j has an equal number of a's and b's. Note that in the worst case $j = 2n+2$.

$j = 2n+2 \Rightarrow$ w' is of the form $a^n b^n$ & can be desired as $S \to aSbS \to aSb \to aaSbSb \to aaSbb \ldots a^n b^n$

Otherwise $j < 2n+2$. In that case, the symbol at position j must be a 'b', w' can be written as $w' = a w_1 b w_2$, where w_1 and w_2 are both strings of length at most $2n$ ($|w_1| = j-2$), each having an equal number of a's and b's.

Thus, by induction, both w_1 and w_2 can be desired from S. $\Rightarrow$ w' can also be desired from S

$S \to aSbS \to aw_1bS \to aw_1bw_2$

Soln 2

1. The variable D is useless because there is no path from S that can take to D

The variable C is useless because it does not give all terminals (C always remains)

Removing these variables & corresponding rules

$S \to aSa \mid A$

$A \to bBb \mid E$

$B \to bBb \mid \epsilon$

$E \to bb \mid bEb$

Step 1. Remove ϵ- production rules

$b \to \epsilon \Rightarrow A \to bb, \; B \to bb$

$\Rightarrow$

$S \to aSa \mid A$

$A \to bBb \mid E \mid bb$

$B \to bBb \mid bb$

$E \to bEb \mid bb$

B & E give rise to the same terminals

$\Rightarrow$

$S \to aSa \mid A$

$A \to bBb \mid B \mid bb$

$B \to bBb \mid bb$

Step 2. Remove unit - production rules

We will apply unit production ~~removal~~ removal

to $A \to B$ before $S \to A$.

$A \to B \overset{new}{\Rightarrow} A \to bBb \mid bb$

now $S \to A \overset{new}{\Rightarrow} S \to bBb \mid bb$ (others repetition)

That gives

$S \to aSa \mid bBb \mid bb$

$A \to bBb \mid bb$

$B \to bBb \mid bb$

At this step, A is useless because no path from start to A

$\Rightarrow$ $S \to aSa \mid bBb \mid bb$

$B \to bBb \mid bb$

Step 3: Let us introduce X & Y as new non-terminals for a & b.

$S \to XSX \mid YBY \mid YY$

$B \to YBY \mid YY$

$X \to a \qquad Y \to b$

Step 4. $S \to XSX \Rightarrow S \to XD$, $D \to SX$

$S \to YBY \Rightarrow S \to YC$, $C \to BY$

$\Rightarrow$ $S \to XD \mid YC \mid YY \qquad X \to a$

$B \to YC \mid YY \qquad Y \to b$

$C \to BY$

$D \to SX$

Soln 3a L_2 is not context-free. Let us try to prove it using pumping lemma.

Game with the Demon

Step 1. The Demon picks k

Step 2. You pick $z = a^k b^k c^k$.

$|z| = 3k \geq k$.

Step 3. Demon segments $z = uvwxy$

$|vwx| \leq k$, $vx \neq \epsilon$

Step 4. Let us find some i such that

$uv^iwx^iy \notin L_2$

Let us consider possible v & x

Case 1: Either v or x runs across the boundary a,b or b,c

take $i=2$ & the resultant string is not of the form $a^* b^* c^*$

Case 2: Both v and x belong to the same block.

→ If they belong to block 'a' or block 'b', then $i=0 \Rightarrow$ number of $c > \#a / \#b$

Not allowed

If they belong to block 'c' $\Rightarrow i=2$ & $\#c > \#a / \#b$

Case 3: v and x belong to adjacent blocks

→ If the blocks are a's and b's

⇒ $i=0$ will give more c's that either a's or b's

→ If the blocks are b's and c's

- Consider sub cases
  - $x = \epsilon$ → $i=0$ will give less b's than c's
  - $x \neq \epsilon$ → $i=2$ will give less a's than c's

In all cases, we can use either $i=0$ or $i=2$ to get a string $\notin L$ ⇒ L_2 is not context-free

Solution 3b: Game with the Demon

Step 1. Demon picks k

Step 2. You pick $z = a^k b^{k^2} \in L$

Step 3. Demon picks u, v, w, x, y

$z = uvwxy$, $|vwx| \le k$, $vx \ne \epsilon$

Step 4: You can choose $i = 2$

Case 1. v and x both contain a's only

$\Rightarrow uv^2wx^2y$ will contain more a's than the square root of b

Case 2. v and x both contain b's only

$\rightarrow uv^2wx^2y$ will contain more b's than the square of a

Case 3. One of v or x contains a's as well as b's

$\Rightarrow uv^2wx^2y$ will not be of the form $a^* b^*$

Case 4. v belongs to a's, x belongs to b's

$\Rightarrow a^i b^j$, $i \ge k$, $k^2 \le j \le k^2 + k < (k+1)^2$ $\left[\begin{array}{l} v = \epsilon \Rightarrow i = k, j > k^2 \Rightarrow \notin L \\ v \ne \epsilon \Rightarrow i = k+1, j < (k+1)^2 \notin L \end{array} \right.$