1)

1. $L = \{c^n b^n a^n \mid n \geq 0\}$ over $\{a, b, c\}$

$L = \{c^n b^n a^n \mid n \geq 0\}$

Assume L is a regular language

let 'p' be the pumping length

consider a string $s = c^n b^n a^n \in L$

$|s| \geq P$

By pumping lemma, take

$S = c^P b^P a^P = xyz$ such that $|xy| \leq P$, $|y| > 0$

Let ccbbaa be the string that belongs to L, i.e pumping lemma, pumping length 'p=2'

To satisfy the conditions of the pumping lemma x='c', y='c', z='bb aa'

$$S = \underset{x}{\underline{c}}\,\underset{y}{\underline{c}}\,\underset{z}{\underline{b\,b\,a\,a}}$$

Pump the middle part such that xy^iz $(i \geq 0)$. For $i=2$ the y becomes 'cc'

The string after pumping is cccbbaa

$$S = (c)(c)^i(bbaa)$$
$$= \underset{x}{\underline{c}}\,\underset{y}{\underline{cc}}\,\underset{z}{\underline{bbaa}} \quad [\text{where } i=2]$$

The string cccbbaa $\notin L$ because the string that is accepted by the language should have equal number of c's, b's and a's.

By proof of contradiction, L is not a regular language.

1)
2. $\{w \mid w \neq w^R\} = L$ over $\{a, b, c\}$

$L = \{w \mid w \neq w^R\}$

Assume that L is regular language
we know that, the compliment of the language L is $\bar{L}$ which is also regular.
As Regular languages are closed under complement.

The compliment of the language L is $\bar{L} = \{w \mid w = w^R\}$ is also regular.
Assume a string $S = 0^P 1 0^P$
Divide the string into three pieces x, y and z
So, $S = 0^P 1 0^P = xyz \in L$ where, P is the pumping length
lets divide S as,

$$0^{P-K} 0^K 1 0^P \quad [\because 0^m 0^n = 0^{m+n}]$$

Assume that $x = 0^{P-K}$, $y = 0^K$ and $z = 10^P$ $[K > 0]$

Now $xy^iz = 0^{P-K} (0^K)^0 10^P$ [when $i = 0$]

$= 0^{P-K} 1 0^P \notin L$ $[\because y^0 = \epsilon]$

The string xy^iz is not same from forward and backward direction because $P-K < P$

So, the string xy^iz does not belong to $\bar{L}$. So, By proof of contradiction $\bar{L}$ is not regular language.
Hence, L is also not regular language as compliment of the language is closed ~~under~~ in regular languages.

1)

3. $L = \{a^n b^m \mid n \neq m\}$

$L = \{a^n b^m \mid n \neq m\}$

Assume L is regular language

let 'p' be the pumping length

Consider a string $S = a^n b^m \in L$

$|S| \geq P$

By pumping lemma, take

$S = a^P b^{P+P!}$ [where $P! = (P) \times (P-1) \times (P-2) \ldots \times 1$]

Divide the string into three pieces x, y and z

so, $S = a^P b^{P+P!} = xyz$

Assume that,

$x = a^u, y = a^v, z = a^w b^{P+P!}$, where $v \geq 1$ and $u+v+w = P$

Now take string $S' = xy^{i+1}z$ where $i = P!$

Then $y^i = a^{P!}$ so $y^{i+1} = a^{v+P!}$, and

so, $xyz = a^{u+v+w+P!} b^{P+P!}$ [$\because u+v+w = P$]

Thus, it gives $xyz = a^{P+P!} b^{P+P!}$

$a^{P+P!} b^{P+P!} \notin L$ [$\because u+v+w = P$]

By this, we get $n = P+P!$ and $m = P+P!$, i.e $m = n$

By proof of contradiction, using pumping lemma it is proved L is not regular.

2)

(a) $\{0^n 1^m \mid n+m \text{ is odd}\}$

The given language is $\{0^n 1^m \mid n+m \text{ is odd}\}$

As, $n+m$ is odd we have 2 possibilities

case (i): 'n' should be odd and 'm' should be even.

case (ii): 'n' should be even and 'm' should be odd.

Our grammer should satisfy both the cases.

The context free grammer that generates the language L.

$L = \{0^n 1^m \mid n+m \text{ is odd}\}$ is given by

$S \rightarrow AB1 \mid A0B$

$A \rightarrow 00A \mid \varepsilon$

$B \rightarrow 11B \mid \varepsilon$

The language contains string $\{0, 1, 011, 001, 00111, 00011, \ldots\}$

let's derive the string 011 (which has odd no. of 0's & even number of 1's)

$S \rightarrow A0B$

$\rightarrow [\varepsilon]0B$

$\rightarrow 0B$

$\rightarrow 0[11B]$

$\rightarrow 011[\varepsilon]$

$\rightarrow 011.$

let's derive the string 001 (which has even no. of 0's & odd number of 1's)

$S \rightarrow AB1$

$\rightarrow [00A]B1$

$\rightarrow 00[\varepsilon]B1$

$\rightarrow 00[\varepsilon]1$

$\rightarrow 001$

let's derive 00111 [even 0's and odd 1's]

$S \to AB1$

$\to 00AB1$

$\to 00[\varepsilon]B1$

$\to 00[11B]1$

$\to 0011[\varepsilon]1$

$\to 00111$

let's derive 0111111[odd 0's & even 1's]

$S \to A0B$

$\to [\varepsilon]0B$

$\to 0B$

$\to 0[11B]$

$\to 011[11B]$

$\to 01111[11B]$

$\to 0111111[\varepsilon]$

$\to 0111111$

∴ we are successful in deriving few strings that are accepted by the language

2)

(b) $\{w \in \{0,1\}^* \mid w \neq w^R\}$

The given language is $\{w \in \{0,1\}^* \mid w \neq w^R\}$

~~The strings of the language are {ab, ba, baba, abaa, aabb, ...}~~

The context-free grammar that generates the language $\{w \in \{0,1\}^* \mid w \neq w^R\}$ is given by

$S \to 0S0 \mid 1S1 \mid 0A1 \mid 1A0$

$A \to 0A \mid 1A \mid \varepsilon$

The strings of the language are {01, 10, 1010, 0100, 0011, ...}

let's derive ab string 01

$S \rightarrow 0A1$
$\rightarrow 0[\varepsilon]1$
$\rightarrow 01$

let's derive 1010

$S \rightarrow 1A0$
$\rightarrow 1[0A]0$
$\rightarrow 10[1A]0$
$\rightarrow 1010$

let's derive 0011

$S \rightarrow 0A1$
$\rightarrow 0[0A]1$
$\rightarrow 00[1A]1.$
$\rightarrow 0011$

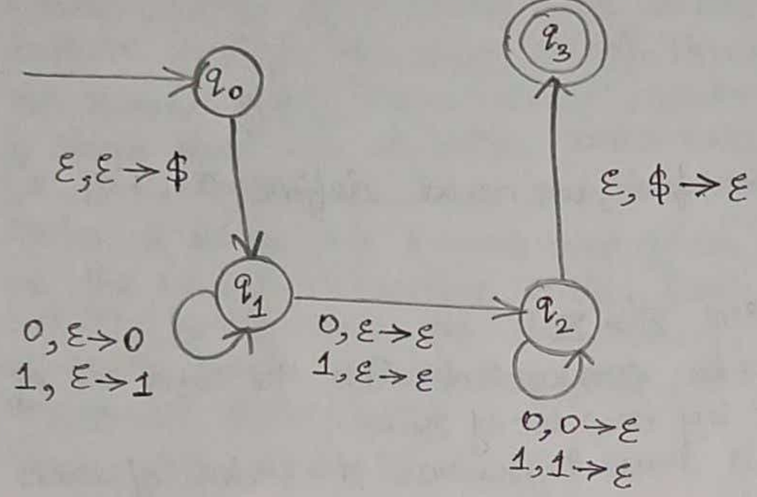
let's derive 0100110

$S \rightarrow 0S0$
$\rightarrow 0[1S1]0$
$\rightarrow 01[0A1]10$
$\rightarrow 010[0A]110$
$\rightarrow 0100110$

We have successfully derived few strings that belong to the given language.

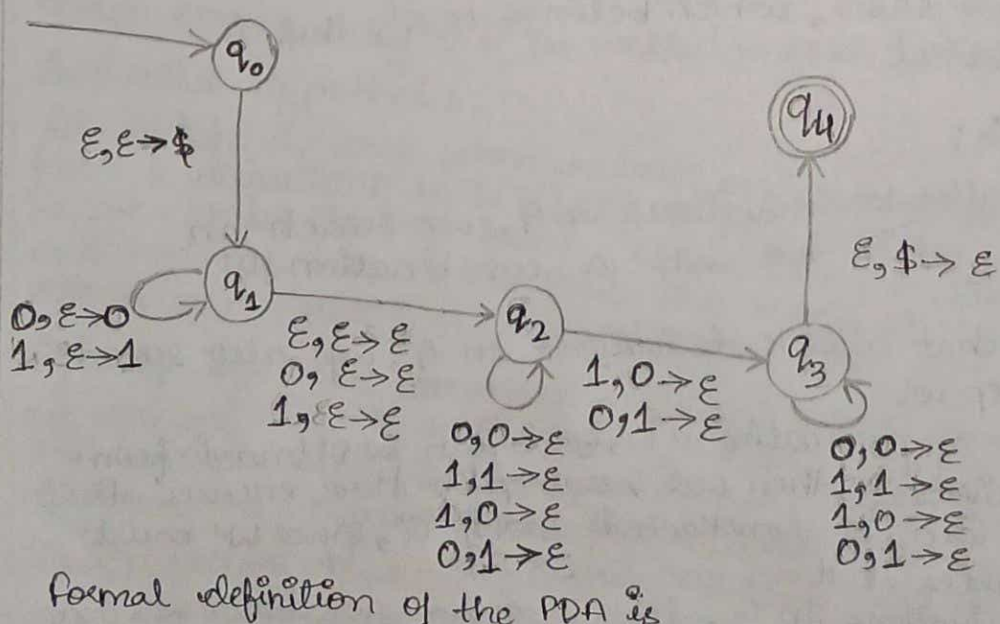
3)

a) $\{0^n 1^m \mid n+m \text{ is odd}\}$



formal definition of ~~above~~ PDA is
$M = (Q, \Sigma, \Gamma, \delta, q_1, F)$

b) $\{w \in \{0,1\}^* \mid w \neq w^R\}$



formal definition of the PDA is
$M = (Q, \Sigma, \Gamma, \delta, q_1, F)$

4) If L is a context-free language, then $L^R = \{w^R | w \in L\}$ is also context-free.

L is context-free language

$G = \{V, \Sigma, R, S\}$

To prove language L^R is context-free, we must define a CFG which constructs L^R

$G' = \{V', \Sigma', R', S'\}$, Here $\Sigma' = \Sigma$

we will show that new CFG can be constucted with the same set of variables as original CFG, but by new set of rules.

As reverse the string in CFG, we will have to reverse the order of each rule we used to generate the CFG.

Rules must be reversed, the order of rules should not be reversed

$\therefore$ start variable should be 'S'.

lets assume that **w** has a derivation in G as $w = uXv$ is a variable in G.

Assume reversal of $w = uXv \to w^R = v^R X u^R$ is derivable using G'.

Base case: when $k=0$

Our derivation must have the derivation $S \to S$, this must be the first derivation rule, ~~to~~ set the initial start variable.

Since reversal of ~~this~~ is $S^R = S$, which belongs to G', this is ~~trivially~~ trivially correct.

Inductive Hypothesis:

let's assume that after k derivations in G, we reach an intermediate step w, which ~~be~~ has a combination of terminals & variables.

Our assumption is that after k derivations in G', we also reach the intermediate step w'.

If w does not have any variables in it, i.e w can be obtained from your CFG using k derivations, then our assumption here ensures that the reversed string can be constucted using G', since w' must also not have variables in it.

However, if after k derivations in G, let us assume we have $w = uXv$ where u and v are a combination of terminals & variables, and X is the next variable on which we apply our derivation. By induction hypothesis, the reversal of $w = w^R = v^R X u^R$, is derivable using k derivations in G', i.e. $w' = w^R$.

Next, the derivation rule exists in G for $X \to a$, where a is some combinations of variables and terminals (can have only one of the two, or just be the empty string). Hence, we have $w_{k+1} = uav$ in G. The reversal of this intermediate construct must be $w^R_{k+1} = v^R a^R u^R$ We need to show that this derivation must exist in G' as well. We know that $w^R = v^R X u^R$ exists in G'. We also know that the reversed derivations rules R' for variable X must now go as $X \to a^R$. Hence, applying this as the next derivation rule from G', on w, we get $w_{k+1}' = v^R a^R u^R$, which is the same as the reversal of the string w_{k+1}. Therefore, at each intermediate step for G, we can attain a corresponding reversed state in G'.

This proves that $L^R \subseteq L(G')$

Now, we must also prove that the CFG G' does not generate strings outside L^R, i.e. $L(G') \subseteq L^R$ For this case, let us again start as before by considering a string w' from L(G'). To prove this, we will show that $w^R \in L$. We will again use a proof similar to previous part, using induction, on the length of the derivation.

Base case: When $k=0$

We have shown that the start state can be the same for both G and G'. In this case, the derivation must have the starting derivation $S \to S$. Since the reversal of this is $S^R = S$, which belongs to G', and hence consequently to L^R as well, this case is trivially correct.

Inductive Hypothesis:

let us take the case when we have an intermediate construct w' obtained from k derivations in L(G'). If w' does not have any variables in it, i.e, w' can be obtained from CFG G' using k derivations, then our assumptions here ensure that the reversed string can be constructed using G, since w must also not have variables in it.

If not, again we can assume without loss of generality, that $w_k' = uXv$. we assume that if $w_k \in L(G') \to w_k = v^R X u^R \in L(G)$. The next derivation rule will be on the variable X here, where $X \to a$. Notice that from from our definition, the equivalent rule from L(G) will go as: $X \to a^R$. Therefore, $w'_{k+1} = uav$. In this case, $w^R_{k+1} = v^R a^R u^R = w'_{k+1} \in L(G)$. Therefore at each intermediate step for G', we can attain a corresponding reversed state in G.

This proves that all strings in $L(G')$ can be represented as a reversal of a string from L, and hence, $L(G') \subseteq L^R$.

Therefore, since $L^R \subseteq L(G')$ and $L(G') \subseteq L^R$, $L(G') = L^R$.

5)

(a) $L_1 = \{a^i b^j c^k \mid i,j,k \geq 0, i=j\}$, $L_2 = \{a^i b^j c^k \mid i,j,k \geq 0, j=k\}$

The context free grammer that generate the language L_1:

$\{a^i b^j c^k \mid i,j,k \geq 0, i=j\}$ is given by

$S \to AB$
$A \to aAb \mid \varepsilon$
$B \to cB \mid \varepsilon$

$G_1 = \{\{S, A, B\}, \{a, b, c\}, \{S \to AB, A \to aAb \mid \varepsilon, B \to cB \mid \varepsilon\}, S\}$

The context free grammer that generate the ~~lange L_2~~ language L_2:

$\{a^i b^j c^k \mid i,j,k \geq 0, j=k\}$ is given by

$S \to AB$
$A \to aA \mid \varepsilon$
$B \to bBc \mid \varepsilon$

$G_2 = \{\{S, A, B\}, \{a, b, c\}, \{S \to AB, A \to aA \mid \varepsilon, B \to bBc \mid \varepsilon\}, S\}$

6)
(c) Given the languages are

$L_1 = \{a^i b^j c^k \mid i,j,k \geq 0,\ i=j\}$

$L_2 = \{a^i b^j c^k \mid i,j,k \geq 0\ j=k\}$

Now we will show that both A and B are context free languages.
In order to show, let us use the grammer that we have generated for L_1

$S \rightarrow AB$
$A \rightarrow aAb \mid \varepsilon$
$B \rightarrow cB \mid \varepsilon$

Observing we say the grammer i.e language L_1 is context-free language.
let us construct grammer that recognizes L_2

$S \rightarrow AB$
$A \rightarrow aA \mid \varepsilon$
$B \rightarrow bBc \mid \varepsilon$

Observing grammer we can say that **language** L_2 is context-free language

Hence, both L_1 and L_2 are context-free language.
consider $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$
Now lets check if $L_1 \cap L_2$ language is contex-free ~~or~~ not using pumping lemma.
let us assume that $L_1 \cap L_2$ is context-free language.
let 'p' be the pumping length for $L_1 \cap L_2$
consider a string $S = a^p b^p c^p$
$S \in L_1 \cap L_2$ and of length 'p'.
Divide 's' into 'uvxyz', 'uv^ixy^iz', condition 2 stipulates that either v or y is non-empty.
consider one of the two cases, depending on whether substring v and y contains more than one type of alphabet symbol.

Case 1:

If both v and y contain only one type of symbol, v doesn't contain both a's and b's or both b's and c's and the same holds for y. Here the string uv^2xy^2z cannot contain equal number of a's, b's and c's. Therefore, it cannot be a member of $L_1 \cap L_2$ which violated the first condition of the pumping lemma and thus is a contradiction to our hypothesis.

Case 2:

If either v or y contains more than one type of symbol uv^2xy^2z may contain equal number of the three alphabets but not in the correct order. Hence it cannot be a member of $L_1 \cap L_2$ and thus is a contradiction to our hypothesis.

However, the both cases raised contradiction. This is because of our assumption $L_1 \cap L_2$ is a context-free language.

Hence, our assumption failed and $L_1 \cap L_2$ is not a context free language.

Hence, we have L_1 and L_2 are context-free languages and $L_1 \cap L_2$ is not a context-free language. So, we can say that the language obtained by intersection of two context-free languages L_1 and L_2 is not a context-free language.

Therefore, the languages L_1 and L_2 are not closed under intersection.

5)

(d) Using DeMorgan's law we will show that the languages L_1 and L_2 is not closed under complement.

DeMorgan's law states that for any two sets L_1 and L_2

$$\overline{L_1 \cup L_2} = \overline{L_1} \cap \overline{L_2}$$

we have L_1 and L_2 are two arbitary context-free languages.

Let these languages are represented in 4-tuples form as $L_1 = (V_1, \Sigma, R_1, S_1)$ and $L_2 = (V_2, \Sigma, R_2, S_2)$ where,

- V_1 and V_2 are finite set of variables of L_1 and L_2 respectively.
- Σ is finite set, disjoint from V_1, V_2 are terminals of L_1 and L_2 respectively.
- R_1 and R_2 are finite set of rules of L_1 and L_2 respectively.
- $S_1 \in V_1$, $S_2 \in V_2$ are the start variables of L_1 and L_2 respectively.

Now construct a grammer G that recognizes $L_1 \cup L_2$.

So $G = (V, \Sigma, R, S)$ where

- $V = V_1 \cup V_2$
- $R = R_1 \cup R_2 \cup \{S \to S_1, S \to S_2\}$ Here, R_1 and R_2 are disjoint.

Now we have to show that L_1 and L_2 are not closed under complement.

lets assume that L_1 and L_2 are closed under complement.

Since, L_1 and L_2 are context-free languages, then $\overline{L_1}$ and $\overline{L_2}$ are also context-free languages. We known that context-free languages are closed under union.
So $\overline{L_1} \cup \overline{L_2}$ is closed. Hence $\overline{L_1} \cup \overline{L_2}$ ~~are~~ is a context-free language.
Since, $\overline{L_1} \cup \overline{L_2}$ is context-free language, we have $\overline{\overline{L_1} \cup \overline{L_2}}$ is a context-free language
Applying DeMorgan's law we get $\overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2$
Hence $L_1 \cap L_2$ is a context-free language which is a contradiction to ~~part~~ (c)
This contradiction occured because our assumption is wrong.
Hence L_1 and L_2 are not closed under complementation.
Therefore, class of context-free ~~gram~~ languages is not closed under complementation.

5)
(b) To show, $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$
From 5) (a) we know the context-free grammer that generates languages L_1, L_2, i.e.
The context free grammer that generates the language L_1:
$\{a^i b^j c^k \mid i, j, k \geq 0, i=j\}$ is given by-

$G_1 = \{\{S, A, B\}, \{a, b\}, P, S\}$
where 'P' represents production rules that are as follows for L_1-

$S \rightarrow AB$
$A \rightarrow aAb \mid \varepsilon$
$B \rightarrow cB \mid \varepsilon$

The context-free grammer that generates the language L_2:

$\{a^i b^j c^k \mid i,j,k \geq 0, j=k\}$ is given by-

$G_2 = \{\{S,A,B\}, \{a,b\}, P, S\}$

where 'P' represents production rules that are as follows for L_2-

$S \rightarrow AB$
$A \rightarrow aA \mid \varepsilon$
$B \rightarrow bBc \mid \varepsilon$

let's take <u>LHS</u>-

$L_1 \cap L_2 \rightarrow$ all strings that are in both the languages ($\therefore$ common strings in L_1 and L_2 is represented as $L_1 \cap L_2$).

let's derive language for L_1 using the language and production rules, we get,

$L_1 = \{\varepsilon, c, ab, abc, aabbc, abcc, a^2b^2c^2, abc^3, abc^2, a^3b^3c, a^3b^3c^3, a^3b^3c^2, \ldots\ldots\}$

let's derive language for L_2 using the language and production rules, we get,

$L_2 = \{\varepsilon, a, bc, ab^2c^2, aabc, abc, a^2b^2c^2, a^3b^2c^2, ab^3c^3, a^2b^3c^3, a^3b^3c^3, ab^4c^4, a^2b^4c^4, a^3b^4c^4, a^4b^4c^4, \ldots\ldots\}$

In L_1, the condition mentioned is $i=j$, but it is not given that $i=j\neq k$, so, we gets strings where $i=j=k$ that belong to language L_1.

Similarly, In L_2, the condition mentioned is $j=k$, but it isn't given that $i\neq j=k$, so we gets strings where $i=j=k$ that belong to language L_2.

So, L_1 and L_2 both have strings where $i=j=k$ and we can observe that in the strings of L_1 and L_2 as mentioned above

we can say that

$L_1 \cap L_2 = \{a^i b^j c^k \mid i,j,k \geq 0, i=j=k\}$

lets substitute, $i=j=k=n$

By substituting 'n' in place of i, j and k we get,

$\{a^n b^n c^n \mid n \geq 0\}$

$\therefore L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$

LHS $\subseteq$ RHS, i.e, $L_1 \cap L_2 \subseteq \{a^n b^n c^n \mid n \geq 0\}$

lets prove the other direction,

let's take RHS -

$\{a^n b^n c^n \mid n \geq 0\} = L'$

The language of the above is as follows:-

$L' = \{abc, \varepsilon, a^2b^2c^2, a^3b^3c^3, \ldots\}$

The language of L_1 is as follows:-

$L_1 = \{\varepsilon, c, ab, abc, a^2b^2, a^2b^2c, a^2b^2c^2, a^3b^3, a^3b^3c, a^3b^3c^2, a^3b^3c^3, \ldots\}$

The language of L_2 is as follows:-

$L_2 = \{\varepsilon, a, bc, abc, b^2c^2, ab^2c^2, a^2b^2c^2, b^3c^3, ab^3c^3, a^2b^3c^3, a^3b^3c^3, \ldots\}$

Now, we can observe that,

$L' \subset L_1$

we also observe that,

$L' \subset L_2$

These can be interpreted as all ~~elements~~ strings of L' are present in L_1 but L' and L_1 are not same (exact same).

Similarly, all ~~elements~~ strings of L' are present in L_2 but L' is not exactly same (i.e not all strings in L_2 are in L') as L_2.

we can say that

$L' = L_1 \cap L_2$ [all elements strings in L' are present in $L_1 \cap L_2$ as $L' \subset L_1$ and $L' \subset L_2$]

= LHS

$\{a^n b^n c^n \mid n \geq 0\} \subseteq L_1 \cap L_2$

we now got,

$L_1 \cap L_2 \subseteq \{a^n b^n c^n \mid n \geq 0\}$ ~~from~~ [by taking LHS.

$\{a^n b^n c^n \mid n \geq 0\} \subseteq L_1 \cap L_2$ by taking RHS.

we can say that,

$L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$

Hence, we have successfully shown that LHS = RHS and also RHS = LHS for,

$$L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$$

6)

1. $\{a^i b^j \mid j = 2i\}$

The language can be written as,

$\{a^i b^{2i}\}$

$L = \{\varepsilon, abb, aabbbb, \ldots\}$

This is a context free language

The production rules of this ~~can be as~~ are as follows -

$$S \rightarrow aSbb \mid \varepsilon$$

let's derive 'aabbbb'

$S \rightarrow aSbb$
$\rightarrow a[aSbb]bb$
$\rightarrow aa[\varepsilon]bbbb$
$\rightarrow aabbbb$

let's derive 'aaabbbbbb'

$S \rightarrow aSbb$
$\rightarrow a[aSbb]bb$
$\rightarrow aa[aSbb]bbbb$
$\rightarrow aaa[\varepsilon]bbbbbb$
$\rightarrow aaabbbbbb$

let's derive 'abb'

$S \rightarrow aSbb$
$\rightarrow a[\varepsilon]bb$
$\rightarrow abb$

The language can be as the following description.

$$G = (V, T, R, S)$$

$$G = \{\{S\}, \{a, b\}, \{S \rightarrow aSbb \mid \varepsilon\}, S\}$$

2. $\{a^i b^j \mid j=i^2\}$

lets assume, that this language is context-free.
According to pumping lemma, there exists a constant 'p' pumping length such that ~~, there exists a constant 'p'~~ any string s is in the language with length atleast p can be split into 5 parts 'uvwxy' satisfying the following conditions:

→ $|vwx| \leq p$
→ $|vx| \geq 1$
→ for all $i \geq 0$, uv^iwx^iy belongs to language.

$S = a^p b^{p^2}$

we can split it into five parts i.e, uvwxy such that the above conditions satisfy.

case 1 - Both v and x contain same symbols:
(i) If both v and x contains only a's then pumping the string will change the number of a's but not the number of b's, which violates $j=i^2$ condition.
(ii) If both v and x contains only b's then pumping the string will change the number of b's but not the number of a's, which violates $j=i^2$ condition.

case 2 - Both v and x don't contain same symbols:
(i) If v contains only a's and x contains only b's then pumping the string will change the number of a's and b's in the same ratio as 'i' will be the same for v and x which violates $j=i^2$.
(ii) This condition is not at all possible i.e v contains only b's and x contains only a's as in our language a's donot follow b's.

Therefore, from both the cases we can say that by proof of contradiction the language is not context-free.
In, all the cases the condition of the language isn't satisfied after pumping string.

∴ By proof of contradiction, The language is not context free.

6)

3. $\{a^i b^j \mid i \neq j\}$

The above given is not regular from 1) (3)
so, it can be context free grammer.
let's try to construct product rules for the language

$L = \{a, b, abb, aab, aaab, aabbb, abbb, \dots\}$

Production Rule -

$S \to aSb \mid X \mid Y$
$X \to aX \mid a$
$Y \to bY \mid b$

let's divide
aabbb
$S \to aSb$
$\to a[aSb]b$
$\to aa[Y]bb$
$\to aa[b]bb$
$\to aabbb$

let's divide
aaaaab
$S \to aSb$
$\to a[X]b$
$\to a[aX]b$
$\to aa[aX]b$
$\to aaa[aX]b$
$\to aaaa[a]b$
$\to aaaaab$

we are successfully able to derive the strings.

$G = (V, T, R, S)$

$G = \{\{S, X, Y\}, \{a, b\}, \{S \to aSb \mid X \mid Y,\ X \to aX \mid a,\ Y \to bY \mid b\}, S\}$