1. 2.30 a)

   To prove that L = $\{0^n1^n0^n1^n | n \geq 0\}$is NOT a context free language.

   We will assume L is a context free language and use the Pumping lemma in order to find a contradiction.

   Now we begin our proof by contradiction

   Assume L is a Context Free language. If L is a Context Free language then there is a integer n for which every word in L of length≥n may be pumped according to the pumping lemma without violating any of the Pumping Lemma's principles.

   According to the Pumping Lemma for some word w in L, it can be broken into uvxyz such that

   ---

   $|vxy| \leq n$

   $|vy| > 0$

   for every integer i ≥ 0 uv^ixy^iz is in L

   Consider the string $0^n1^n0^n1^n$. Clearly it is longer than n and in L. So then, we can divide it into uvxyz such that all of the conditions of the Pumping are met.

   There are 4 ways in which this can be done.

   Case 1: v is the empty string. then y must contain either one ore more zeros, one or more ones, or at least one zero and one one. if y contains only one type of element and we pump it up, Then the word produced will not contain the same number of ones as zeros. if y contains multiple types of elements and we pump up w, then we will produce a word where the zeros and ones are not in an order corresponding to any word in L.

   Case 2: y is the empty string. then for every possible composition of v the exact same violations occur as in case 1 but with regards to v instead of y.

   Case 3: V contains only 0's or only 1's and Y contains only 0's or only 1's. If we pump up i, 2 of the alternating occurrences of 0 or 1 in the string will be pumped up, but it is impossible for the other 2 to change. Therefore, there will be a different amount of the repeating characters in two of the four substrings composed of the same repeated element.

   Case 4: Either V or Y contains a mix of zeros and ones. When we pump up any such string, we will produce a pattern which is not in accordance with the pattern of every word in L. that is the zeros and ones will not be in the correct order.

   We know every possible breakdown of w must belong to one of these cases, and each of these cases shows that the third principle is violated when i is pumped up. Therefore every word in L cannot be pumped, But we know every word in every CFL must be pumpable.

So then our assumption that L is a CFL must be wrong. Therefore we have proved L is not a CFL.

---

d)

To prove that L = $\{t_1\#t_2\#...\#t_k|k \geq 2$, each $t_i \ \in \{a,b\}^*$ $and\ t_i = t_j\ for\ some\ i \neq j\}$is NOT a context free language.

We will assume L is a context free language and use the Pumping lemma in order to find a contradiction.

Now we begin our proof by contradiction

Assume L is a Context Free language. If L is a Context Free language then there is a integer n for which every word in L of length≥n may be pumped according to the pumping lemma without violating any of the Pumping Lemma's principles.

According to the Pumping Lemma for some word w in L, it can be broken into uvxyz such that

---

$|vxy| \leq n$

$|vy| > 0$

for every integer i ≥ 0 uv^ixy^iz is in L

Consider the string w = $\{t_1\#t_2|t_1, t_2 \ \in \{a,b\}^*, and t_1 = t_2, and |t_1| = n, and t_1 = \{aba\}$. Clearly w is in L and w is larger than n. So then, we can divide it into uvxyz such that all of the conditions of the Pumping are met.

There are 4 ways in which this can be done.

Case 1: v is the empty string. then y must contain either one or more elements of some t_i or '#' or both, but not all of the members of any t_i in addition to '#' (because —vxy— cannot be larger than n, and each t_i is the size of n). if y contains only '#' and we pump it up by one, and if none of the t_i's are the empty string, then the word produced will contain one $t_i = \epsilon$ which is not equal to any other t_i . if y contains any part or the whole of some t_i, then pumping it up will fail for the same reason: no other t_i will be equal to the one containing the pumped up segment. The other possibility is that y contains part of some t_i and '#' but then pumping down to i = 0 will remove the '#' and the word produced will not be in L.

Case 2: y is the empty string. then for every possible composition of v the exact same violations occur as in case 1 but with regards to v instead of y.

Case 3: Either v or y contains '#'. pumping down to i = 0 will produce a word without '#' which is not in L.

Case 4: neither v nor y contains '#' and neither v nor y is empty. Since —vxy— $\leq$ n , v cannot be t_1 and y be t_2. therefore any decomposition of the string w will yield either a v and a y which are different, and then pumping them up will yield t_i's which are different. Or if v and y are the same and each comprises some substring of some t_i, then pumping them up will produce two t_i's which have the same number of the same characters, but the characters will be out of order and thus the t_i's will not be equal.

We know every possible breakdown of w must belong to one of these cases, and each of these cases shows that the third principle is violated when i is pumped up. Therefore every word in L cannot be pumped, But we know every word in every CFL must be pumpable. So then our assumption that L is a CFL must be wrong.

Therefore we have proved L is not a CFL.

2.31

To prove that L = $\{w|w$ is a palindrome with equal number of zeros and ones $\}$ is NOT a context free language.

We will assume L is a context free language and use the Pumping lemma in order to find a contradiction.

Now we begin our proof by contradiction

Assume L is a Context Free language. If L is a Context Free language then there is a integer n for which every word in L of length$\geq$n may be pumped according to the pumping lemma without violating any of the Pumping Lemma's principles.

According to the Pumping Lemma for some word w in L, it can be broken into uvxyz such that

---

$|vxy| \leq n$

$|vy| > 0$

for every integer i $\geq$ 0 uv^ixy^iz is in L

Consider the string w = $0^n1^{n*2}0^n$ if. Clearly it is longer than n and in L. So then, we can divide it into uvxyz such that all of the conditions of the Pumping are met.

There are 5 ways in which this can be done.

Case 1: v is the empty string. then y must contain either one ore more zeros, one or more ones, or at least one zero and one one. if y contains only one type of element and we pump it up, Then the word produced will not contain the same number of ones as zeros. if y contains multiple types of elements and we pump up w, then we will produce a word where the zeros and ones are not in an order corresponding to any word in L.

Case 2: y is the empty string. then for every possible composition of v the exact same violations occur as in case 1 but with regards to v instead of y.

Case 3: V contains only 0's and Y contains only 0's =. If we pump up i, we will have more zeros than ones.

Case 4: V contains only 1's and Y contains only 1's =. If we pump up i, we will have more ones than zeros.

Case 5: Either V or Y contains a mix of zeros and ones. Since —vxy— cannot be larger than n, it is impossible for v and y to contain 2 zeros which are an equal distance from the middle of w, (because it would mean —vxy— would transcend ALL of the ones which are 2n in length, which violates the pumping lemma). When we pump up any such string, we will produce a pattern which is not in accordance with the pattern of every word in L. that is the zeros and ones will not create a palindrome.

We know every possible breakdown of w must belong to one of these cases, and each of these cases shows that the third principle is violated when i is pumped up. Therefore every word in L cannot be pumped, But we know every word in every CFL must be pumpable. So then our assumption that L is a CFL must be wrong. Therefore we have proved L is not a CFL.

2.32

To prove that L = $\{w|w = \{1, 2, 3, 4\}^*$ and the number of ones equals the number of twos, and the number of threes equals the number of fours $\}$is NOT a context free language.

We will assume L is a context free language and use the Pumping lemma in order to find a contradiction.

Now we begin our proof by contradiction

Assume L is a Context Free language. If L is a Context Free language then there is a integer n for which every word in L of length$\geq$n may be pumped according to the pumping lemma without violating any of the Pumping Lemma's principles.

According to the Pumping Lemma for some word w in L, it can be broken into uvxyz such that

---

$|vxy| \leq n$

$|vy| > 0$

for every integer i ≥ 0 uv^ixy^iz is in L

Consider the string w = $1^n 3^n 2^n 4^n$ if. Clearly it is longer than n and in L. So then, we can divide it into uvxyz such that all of the conditions of the Pumping are met.

We will find that all of the ways in which we can decompose the string fail to meet the pumping lemma for the same reason. That is: since there are exactly n elements between the ones and the twos (and similarly between the threes and the fours), it is impossible for vxy to transcend across and consist of more than 2 different characters which are not adjacent in w, because the total size of vxy cannot exceed the size of the substring of any one element. Therefore it is not possible in pumping up any such string to increase the number of Ones AND Twos OR the number of Threes AND Fours.

Consider the following 6 cases.

Case 1: if we select v to be a 1, y can only consist of 1's or 3's. If we pump up we will not have as many 2's as 1's.

Case 2: if we select v to be a 3, y can only consist of 3's or 2's. If we pump up we will not have as many 4's as 3's.

Case 3: if we select v to be a 2, y can only consist of 2's or 4's. If we pump up we will not have as many 1's as 2's.

Case 4: if we select v to be a 4, y can only consist of 4's. If we pump up we will not have as many 3's as 4's.

Case 5: if we select v to be empty, y can only consist of either ones and threes, threes and twos, twos and fours, or a string of any one repeating element. In all scenarios we will be unable to pump up and yield a string in L.

Case 6: if we select y to be empty, we will receive the same results as in Case 5, but finding violations of the Pumping Lemma with respect to v instead of y.

In each of the cases as we pump up, either a single element's number of occurrences increases, or ones and threes increase, or threes and twos, or twos and fours, but never ones and twos simultaneously, or threes and fours simultaneously. We know every possible breakdown of w must belong to one of these cases, and each of these cases shows that the third principle is violated when i is pumped up. Therefore every word in L cannot be pumped, But we know every word in every CFL must be pumpable. So then our assumption that L is a CFL must be wrong. Therefore we have proved L is not a CFL.

---

2. 2.33

   To prove that F is NOT a context free language.

   We will assume F is a context free language and use the Pumping lemma in order to find a contradiction.

   Now we begin our proof by contradiction

Assume F is a Context Free language. If F is a Context Free language then there is a integer n for which every word in F of length≥n may be pumped according to the pumping lemma without violating any of the Pumping Lemma's principles.

According to the Pumping Lemma for some word w in F, it can be broken into uvxyz such that

---

$|vxy| \leq n$

$|vy| > 0$

for every integer i ≥ 0 uv^ixy^iz is in F

Consider the string w = a^nb^n. Clearly it is longer than n and in L. So then, we can divide it into uvxyz such that all of the conditions of the Pumping are met.

We will find that all of the ways in which we can decompose the string fail to meet the pumping lemma for the same reason. That is: since there are exactly n elements between the ones and the twos (and similarly between the threes and the fours), it is impossible for vxy to transcend across and consist of more than 2 different characters which are not adjacent in w, because the total size of vxy cannot exceed the size of the substring of any one element. Therefore it is not possible in pumping up any such string to increase the number of Ones AND Twos OR the number of Threes AND Fours.

There are 6 ways in which this can be done.

Case 1: v is the empty string. then y must contain either one or more elements of a or b or both, . if y contains any part or the whole of some substring, then pumping it up will fail because the new word is not F:

Case 2: y is the empty string. then for every possible composition of v the exact same violations occur as in case 1 but with regards to v instead of y.

Case 3: v contains only a's and y contain's only b's. Then pumping them up evenly does not produce a word in F

Case 4: v and y both contain only a's or only b's. Then if we pump down to i = 0 the word produced is not in F

Case 5: v contains only a's and y contains a mix of a's and b's. If we pump up, the word produced is not F.

Case 6: v contains a's and b's and y contains only b's. This fails for the same reason as Case 5.

We know every possible breakdown of w must belong to one of these cases, and each of these cases shows that the third principle is violated when i is pumped up. Therefore every word in L cannot be pumped, But we know every word in every CFL must be pumpable. So then our assumption that L is a CFL must be wrong. Therefore we have proved L is not a CFL.

---

3. problem 2.50

   To show that the class of CFLs is not closed under the operation CUT() we must show that some language produced by the operation on a CFL fails to belong to the class of CFLs.

   Let the language A = $\{a^n b^i c^i|$ n and i are non-negative integers$\}$ so that A contains the same number of b's and c's and is an element of the set of CFL's

   We will utilize a proof by contradiction in which we will assume the class of CFL's is closed under Cut(). So then for every CFL A, CUT(A) produces a CFL.

   Now we begin our proof by contradiction

   Given that A is a CFL. Assume CFL's are closed under CUT(). Since CFL's are closed under CUT(), all the languages produced by said operation will be CFL's .

   Consider the language A where x corresponds to the substring of a's, y corresponds to the substring of b's, and z corresponds to the substring consisting of all the c's. Then CUT(A) produces a language yxz which could be defined as CUT(A) = $\{b^i a^n c^i|$ n and i are non-negative integers$\}$
   where there are the same number of b's as c's and zero or more a's in between

   We will now show via a contradiction proof that CUT(A) is not context free and in doing so find a contradiction.

   Let L = CUT(A)

   Assume L is a Context Free language. If L is a Context Free language then there is a integer n for which every word in L of length≥n may be pumped according to the pumping lemma without violating any of the Pumping Lemma's principles.

   According to the Pumping Lemma for some word w in L, it can be broken into uvxyz such that

   ---

   $|vxy| \leq n$

   $|vy| > 0$

   for every integer i ≥ 0 uv^ixy^iz is in L

   Consider the string w = $a^ib^nc^i$. Clearly it is longer than n and in L. So then, we can divide it into uvxyz such that all of the conditions of the Pumping are met.

   We will find that all of the ways in which we can decompose the string fail to meet the pumping lemma for the same reason. That is: since there are exactly n elements between the b's and the b's, it is impossible for vxy to transcend across and consist of more than 2 different characters which are not adjacent in w, because the total size of vxy cannot exceed the size of the substring of a's. Therefore it is not possible in pumping up any such string to increase the number of b's AND c's simultaneously.

Consider the following 5 cases.

Case 1: if we select v to be a 'b', y can only consist of b's or a's. If we pump up we will not have as many c's as b's.

Case 2: if we select v to be a 'a', y can only consist of a's or c's. If we pump up we will not have as many b's as c's.

Case 3: if we select v to be a c, y can only consist of c's. If we pump up we will not have as many b's as c's.

Case 4: if we select v to be empty, y can only consist of either b's and a's, a's and c's, or a string of any one repeating element. In all scenarios we will be unable to pump up and yield a string in L.

Case 5: if we select y to be empty, we will receive the same results as in Case 5, but finding violations of the Pumping Lemma with respect to v instead of y.

We know every possible breakdown of w must belong to one of these cases, and each of these cases shows that the third principle is violated when i is pumped up. Therefore every word in L cannot be pumped, But we know every word in every CFL must be pumpable. So then our assumption that L is a CFL must be wrong. Therefore we have proved L is not a CFL.

Therefore, L = CUT(A) is not a CFL. But we a said the class of CFL's is closed under the operation CUT(). If A is a CFL then CUT(A) must be a CFL, but CUT(A) is not a CFL so we have a contradiction.

Then our assumption was wrong and we have sufficiently proved that the class of CFL's is not closed under the CUT operation.
