So you want to prove a language is / is not Context Free?

Rebecca J Young

I think it's CF...

Great! Write a grammar. Or draw a PDA. These are your options. If you cannot make a grammar that generates the language nor a PDA that recognises the language, then you may need to reassess. Note that writing a grammar that *almost* generates the language is not good enough.

I think it's NOT CF...

Great! This is what the pumping lemma for CFL is good for.

Do not use the pumping lemma for CFL to prove a language *is* Context Free. Do not do it.

The ingredients you will need for a pumping lemma proof include:

1. a word that is in your language, that is longer than 2^{k-1};
2. a list of cases: i.e., a list of ways in which the word can be subdivided into u, v, x, y, z; and
3. for each case, an i that generates a word not in the language.

How do I find a good word?

1. Ensure the word is in the language, and is sufficiently long.
2. Make the word exactly as complicated as necessary. You are not required to provide a word that 'represents' the language. The more complicated the word, the more cases you will need to deal with.
3. Make each section of the word as long as possible. Usually this will mean that each character occurs $\geq 2^k$ times. However, sometimes this is not possible.
4. Make the word as fragile as possible. For each character block, ensure that EITHER removing a single character OR repeating the character will cause the word to break. For each pair of neighbouring character blocks, ensure that EITHER removing a single character from each block OR repeating characters from both blocks will cause the word to break. As above, sometimes this is not possible.

How do I find all the cases?

It is often wise to list all your cases exhaustively *before* you try to simplify them. That way you can check whether there are any edge cases you are at risk of missing.

How do I break each case?

Remember, you can use a different i per case. You are not required to use the same i for every case.

As to how to find the i for a given case... I mean... you go up or you go down. If deleting the letters in your vy will create a word that is not in the language, then you would set your $i = 0$, thus getting $w' = uv^0xy^0z = uxz$. If repeating the letters in your vy will create a word that is not in the language, then you would set your $i = 2$ (for example), thus getting $w' = uv^2xy^2z = uvvxyyz$.

The things that can make this step difficult include:

- needing to pump up by a specific number (e.g., $i = 27$);
- needing to pump up by a specific number that is related to your N (e.g., $i = 2N$); or
- needing to pump up some arbitrary number of times (e.g., the proof that the square numbers represented in unary is not a CFL).

Examples

Prove that the language $a^x b^x c^x$ where x is a non-negative integer is not context free.

1. A word
Let $N = 2^k$. Let $w = a^N b^N c^N$. Note that $w \in L$ and $|w| > 2^{k-1}$.

2. Cases
Let's start by exhaustively listing each case we can see. We are interested in spans of letters that go up to 2^k. To ensure we capture every possible span of letters up to length 2^k, we will also consider neighbouring spans of letters.

For each case, we will colour the span of letters that vy may appear in. The cases cumulatively should cover all possible locations of vy within the word.

1.	vy comprises one or more a.	$aa..aa\ bb..bbcc..cc$
2.	vy comprises one or more a and one or more b.	$aa..aabb..bb\ cc..cc$
3.	vy comprises one or more b.	$aa..aa\ bb..bb\ cc..cc$
4.	vy comprises one or more b and one or more c.	$aa..aa\ bb..bbcc..cc$
5.	vy comprises one or more c.	$aa..aabb..bb\ cc..cc$

Note that it is not possible for v to contain one or more a and y to contain one or more c, as $|vxy| \leq 2^k$, but $|ab^N c| > 2^k$.

Let's try reducing the number of cases we have by combining some cases. We will colour the span of letters that vy *must* appear in one colour, and the span of letters they *may* appear in another colour. Thus a case such as this: $xx\ yy$ is equivalent to the two cases: $xx\ yy$ and $xxyy$.

1.	vy comprises one or more a and zero or more b.	$aa..aa\ bb..bb\ cc..cc$
2.	vy comprises one or more b and zero or more c.	$aa..aa\ bb..bb\ cc..cc$
3.	vy comprises one or more c.	$aa..aabb..bb\ cc..cc$

Note that it is not possible for vy to contain one or more a and one or more c, as $|vxy| \leq 2^k$, but $|ab^N c| > 2^k$.

This looks good! We have converted our original cases 1 and 2 into the new case 1; and our original cases 3 and 4 into the new case 2.

With this problem, we can actually do it in one case:

1.	vy must contain a minimum of one kind of letter (from a, b, c), and a maximum of two kinds of letters.	$aa..aa\ bb..bb\ cc..cc$	OR
		$aa..aa\ bb..bb\ cc..cc$	OR
		$aa..aa\ bb..bb\ cc..cc$	OR
		$aa..aa\ bb..bb\ cc..cc$	

Note that it is not possible for vy to contain three kinds of letters, as $|vxy| \leq 2^k$, but $|ab^N c| > 2^k$.

3. Break each case
Let's try breaking the version of cases where there are three separate cases:

1. Let p be the number of as in vy, and q be the number of bs in vy; note that $p > 0$. When $i = 2$, the number of as in the word will increase while the number of cs will remain the same. We will create the word: $uv^ixy^iz = uv^2xy^2z = uvvxyyz = a^{N+p}b^{N+q}c^N$. As $p > 0$, the number of as is not equal to the number of cs.
2. Let p be the number of bs in vy, and q be the number of cs in vy; note that $p > 0$. When $i = 2$, the number of bs in the word will increase while the number of as will remain the same. We will create the word: $uv^ixy^iz = uv^2xy^2z = uvvxyyz = a^Nb^{N+p}c^{N+q}$. As $p > 0$, the number of bs is not equal to the number of as.
3. Let p be the number of cs in vy; note that $p > 0$. When $i = 2$, the number of cs in the word will increase while the number of as and bs will remain the same. We will create the word: $uv^ixy^iz = uv^2xy^2z = uvvxyyz = a^Nb^Nc^{N+p}$. As $p > 0$, the number of cs is not equal to the number of as and bs.

Now that we have shown that *every* possible way of subdividing the word into u, v, x, y, z has some method by which pumping the word will lead to creating a word not in the language, we can conclude that there **does not exist a valid way to subdivide the word into** u, v, x, y, z. Thus the Pumping Lemma has not been satisfied.

A common mistake is to state that the Pumping Lemma is not satisfied at the end of each case. Note that we cannot conclude that there is no way to subdivide the word - and thus, that the Pumping Lemma is not satisfied - until we have exhausted *every* possible subdivision.

Example proof write up

Assumption: Let L be the language $a^x b^x c^x$ where x is a non-negative integer. We assume, for the purpose of contradiction, that L is a Context Free language.

Discussion: As L is context free, by assumption, there exists some grammar in Chomsky Normal Form that generates $L - \{\varepsilon\}$. Let k be the number of non-terminals in that grammar.
As L is context free, by assumption, it satisfies the Pumping Lemma for Context Free Languages. That is, for all words $w \in L$ where $|w| > 2^{k-1}$, there exists u, v, x, y, z such that:

1. $w = uvxyz$,
2. $vy \neq \varepsilon$,
3. $|vxy| \leq 2^k$, and
4. for all $i \geq 0$, $uv^i xy^i z \in L$.

Let $N = 2^k$. Let $w = a^N b^N c^N$. Note that $w \in L$ and $|w| > 2^{k-1}$.

As $|vxy| \leq 2^k = N$ (rule 3), we reason that there are three cases we need to consider:
Case 1: vy comprises one or more a and zero or more b.
Case 2: vy comprises one or more b and zero or more c.
Case 3: vy comprises one or more c.
Note that it is not possible for vy to contain one or more a and one or more c, as $|vxy| \leq 2^k$, but $|ab^N c| > 2^k$.

Case 1: Let p be the number of as in vy, and q be the number of bs in vy; note that $p > 0$. When $i = 2$, the number of as in the word will increase while the number of cs will remain the same. We will create the word: $uv^i xy^i z = uv^2 xy^2 z = uvvxyyz = a^{N+p} b^{N+q} c^N$. As $p > 0$, the number of as is not equal to the number of cs. Thus $uv^2 xy^2 z$ cannot be a word in L. This is not a valid u, v, x, y, z.

Case 2: Let p be the number of bs in vy, and q be the number of cs in vy; note that $p > 0$. When $i = 2$, the number of bs in the word will increase while the number of as will remain the same. We will create the word: $uv^i xy^i z = uv^2 xy^2 z = uvvxyyz = a^N b^{N+p} c^{N+q}$. As $p > 0$, the number of bs is not equal to the number of as. Thus $uv^2 xy^2 z$ cannot be a word in L. This is not a valid u, v, x, y, z.

Case 3: Let p be the number of cs in vy; note that $p > 0$. When $i = 2$, the number of cs in the word will increase while the number of as and bs will remain the same. We will create the word: $uv^i xy^i z = uv^2 xy^2 z = uvvxyyz = a^N b^N c^{N+p}$. As $p > 0$, the number of cs is not equal to the number of as and bs. Thus $uv^2 xy^2 z$ cannot be a word in L. This is not a valid u, v, x, y, z.

After considering all possible cases, we conclude that we cannot find a valid u, v, x, y, z.

Contradiction: Thus L does not satisfy the Pumping Lemma for Context Free Languages, but as a CFL it must satisfy the Pumping Lemma for Context Free Languages.
This is a contradiction. We must conclude that it is not true that L is a context free language.

Conclusion: We have proven using contradiction that L is not a context free language.

Prove that the language of strictly increasing sequences of numbers in unary is not context free.

1. A word

Let $N = 2^k$. Let's consider some options:

- $1, 11, 111$: this is a word in L, but it is not long enough
- $1^N, 1^N$: this is a word that is long enough, but it is not in L as it is not a strictly increasing sequence
- 1^N: this is a word in L that is long enough, but it can be pumped up and down and still remain in the language
- $1^N, 1^{2N}$: this is a word in L that is long enough, but when v contains x 1s from the first number and y contains x 1s from the second number it can be pumped up and down and still remain in the language
- $1, 11, 1^N, 1^{2N}, 1^{3N}$: this is a word in L that is long enough, there exists no valid vy that can be pumped, but it is needlessly complicated and has a lot of cases
- $1^N,^N 1^{2N}$: this word minimises the number of cases, but it is not a word in L as we cannot have N ',' in a row
- $1^N, 1^{2N}, 1^{3N}$: this is a word in L that is long enough, there exists no valid vy that can be pumped, and the number of cases is minimised; but it is not fragile, which means it will be more complicated to explain how it is possible to break each case
- $1^N, 1^{N+1}, 1^{N+2}$: this is a word in L that is long enough, there exists no valid vy that can be pumped, the number of cases is minimised, and it is fragile. We can see that each part of this word requires just one character to be added or removed to break the word.

Let $N = 2^k$. Let $w = 1^N, 1^{N+1}, 1^{N+2}$. Note that w represents the sequence $N, N+1, N+2$ which is a strictly increasing sequence, thus $w \in L$. Note also that $|w| > 2^{k-1}$.

2. Cases

1.	vy comprises one or more 1 from the number N.	11...11, 11...111, 11...1111
2.	vy contains the first ',' and zero or more 1.	11...11 , 11...111 , 11...1111
3.	v comprises one or more 1 from the number N and y comprises one or more 1 from the number $N+1$.	11...11 , 11...111 , 11...1111
4.	vy comprises one or more 1 from the number $N+1$.	11...11, 11...111 , 11...1111
5.	vy contains the second ',' and zero or more 1.	11...11, 11...111 , 11...1111
6.	v comprises one or more 1 from the number $N+1$ and y comprises one or more 1 from the number $N+2$.	11...11, 11...111 , 11...1111
7.	vy comprises one or more 1 from the number $N+2$.	11...11, 11...111, 11...1111

Note that it is not possible for v to contain one or more 1 from N and y to contain one or more 1 from $N+2$, as $|vxy| \leq 2^k$, but $|1, 1^{N+1}, 1| > 2^k$. Note that it is not possible for vy to contain both ',' for similar reasons.

1.	vy contains one ',' and zero or more 1.	11...11 , 11...111 , 11...1111 11...11, 11...111 , 11...1111
2.	vy comprises one or more 1 from the number N, and zero or more 1 from the number $N+1$.	11...11 , 11...111 , 11...1111 11...11 , 11...111 , 11...1111
3.	vy comprises one or more 1 from the number $N+1$, and zero or more 1 from the number $N+2$.	11...11, 11...111 , 11...1111 11...11, 11...111 , 11...1111
4.	vy comprises one or more 1 from the number $N+2$.	11...11, 11...111, 11...1111

Note that it is not possible for v to contain one or more 1 from N and y to contain one or more 1 from $N+2$, as $|vxy| \leq 2^k$, but $|1, 1^{N+1}, 1| > 2^k$. Note that it is not possible for vy to contain both ',' for similar reasons.

While we could reduce this down to three cases, it would be unwise to reduce it more than that. One reason we identify cases is so we can use different values of i and provide different justifications for why the case breaks. Trying to overgeneralise will make justifying how we break each case far more difficult than is necessary.

3. Break each case

Let's try breaking the version of cases where there are four separate cases:

1. Let X be the part of vy that the ',' is found (i.e., if the ',' is in v then $X = v$). Let p be the number of 1 before the ',' in X and q be the number of 1 after the ',' in X. When $i = 3$ the created word will contain the substring $XXX = 1^p, 1^{p+q}, 1^{p+q}, 1^q$. Thus the created word contains the same number twice in a row, creating a sequence that is no longer strictly increasing.
2. Let p be the number of 1 from the number N in vy, and q be the number of 1 from the number $N+1$ in vy; note that $p > 0$. When $i = 3$, $uv^ixy^iz = uv^3xy^3z = 1^{N+2p}, 1N+1+2q, 1^{N+2}$. As $p > 0$, $N + 2p \geq N + 2$. Thus the first number in the sequence is greater or equal to the last number in the sequence, creating a sequence that is no longer strictly increasing.
3. Let p be the number of 1 from the number $N+1$ in vy, and q be the number of 1 from the number $N+1$ in vy; note that $p > 0$. When $i = 0$, $uv^ixy^iz = uv^0xy^0z = uxz = 1^N, 1^{N+1-p}, 1^{N+2-q}$. As $p > 0$, $N \geq N + 1 - p$. Thus the second number in the sequence is less than or equal to the first number in the sequence, creating a sequence that is no longer strictly increasing.
4. Let p be the number of 1 from the number $N+2$ in vy; note that $p > 0$. When $i = 0$, $uv^ixy^iz = uv^0xy^0z = uxz = 1^N, 1^{N+1}, 1^{N+2-p}$. As $p > 0$, $N + 1 \geq N + 2 - p$. Thus the third number in the sequence is less than or equal to the second number in the sequence, creating a sequence that is no longer strictly increasing.

We now have all the ingredients to formally write up a proof that this language is not context free!
I leave this as an exercise for the reader.