Context-Free Pumping Lemma Exercise
Martha Kosa

You will play the Context-Free Pumping Lemma game with the language $\{a^k b^n c^n d^j \mid k \geq j\}$. This language is in fact context-free, so you will be able to win the game! For any sufficiently long string in the language, we will be able to decompose it into five concatenated pieces u, v, x, y, and z such that at least one of v and x is nonempty, the substring vxy is not too long, and the string formed by u followed by i copies of v followed by x followed by i copies of y followed by z will belong to the language for all nonnegative integers i.

Try It!

1. Click the **Context-Free Pumping Lemma** button on the main JFLAP screen.
2. Select the **You go first** radio button.
3. Select the language corresponding to our language $\{a^k b^n c^n d^j \mid k \geq j\}$. Your window should look like the following.

1. Click on the **Explain** button to verify that you will be able to win and decompose any sufficiently large string to satisfy the Context-Free Pumping Lemma. It gives you hints about possible decompositions.
2. Enter a positive integer for the middle substring length threshold m in the first text box that you see. If it is too small or too large, you will be prompted to enter another number in a certain range. The number 4 is a good choice. Why? Your window should look similar to the following if 4 is entered. JFLAP has selected a candidate string for exploring the Context-Free Pumping Lemma.

1. The middle substring vxy cannot be longer than m. Move the slider bars to set your substrings in order to satisfy the Context-Free Pumping Lemma. Make sure that you do not see a message at the bottom that starts with **Condition violated**. We need to ensure that the number of b's is always the same as the number of c's. Thus, v and y must be the same size and must only contain one type of character. In the example shown, there is only one possible valid choice for v and y. What is it? When you have determined your choice, click the **Set uvxyz** button.
2. JFLAP has chosen a value of i in an attempt to produce a contradiction for the Context-Free Pumping Lemma. Click on the **Step** button as needed to see the string formed by u followed by i copies of v followed by x followed by i copies of y followed by z. Observe that the string satisfies the definition of the language. Your window should look similar to the following:

1. A case analysis section appears on the right side of the window. Click the **List**

button. Your window should look similar to the following.

1. Notice that there are a lot of cases. This corresponds to all the possible ways our string can be decomposed into five pieces. If we were showing that a language is not context-free, we would be striving for a contradiction to the Context-Free Pumping Lemma. We would have to verify that every case leads to a violation of the Context-Free Pumping Lemma (v and y are both empty, vxy is longer than m symbols, or there exists an i such that the string formed by u followed by i copies of v followed by x followed by i copies of y followed by z is invalid). Since our language is context-free, we only need to find one case that works. Which case was it? Proving impossibility results is harder than proving possibility results. You have not proven that your language is context-free.
2. How do you prove that a language is context-free? You prove that a language is context-free by producing a one-stack PDA that accepts it or by producing a context-free grammar that generates it. Use the PDA feature or the CFG feature of JFLAP to design a PDA that accepts your example language or a CFG that generates it. You cannot build a DFA or a regular grammar because the example language is not regular; that can be proven via the Regular Pumping Lemma When designing your PDA or grammar, it may help to express the example language as the union of two languages. The number of a's in any valid string cannot equal the number of d's. When two numbers are not equal, what are the two possible relationships between them?