Pumping Lemma

Let us now have a look at what we can do with the pumping lemmas. Our main use for them is as aids to show that some given languages are non-context-free. Both of the lemmas state that for any context-free language, any word that is 'long enough' can be broken up in a way which allows us to 'pump' the word. The words we obtain by pumping will also be in the language. So if we are asked to show that a given language is non-context-free, we give a proof by contradiction. First we assume that the language *is* context-free which means that either of the pumping lemmas can be applied. Then we try to derive a contradiction. If we succeed in deriving a contradiction, we are forced to retract our assumption and thus to conclude that the language is non-context-free.

The question is how to derive the contradiction. Well, if the language is context-free, then according to the pumping lemma *every* sufficiently long word can be broken up and then pumped so that all words obtained will also be in the language. We, however, want to find a word that will contradict this statement. We want to find a sufficiently long word in the language which *cannot* be broken up and pumped such that all resultant words are in the language. To put it differently, suppose we take a sufficiently long word in the language, say w, and we consider all permissible ways in which w can be broken up. If we can show that, when we pump the word, we always (i.e. for each permissible way to break up the word) get a new word which is *not* in the language, then we have derived a contradiction.

Let us now consider the application of the two pumping lemmas in more detail. If you are required to prove that a (given) language is non-context-free, the way to go about it is as follows.

1. Assume the given language is context-free. This means there is a CFG in CNF with, say, p live productions that generates the language. Because we assume the language is context-free, the pumping lemma can be applied.

2. Now we have to find a word that consists of more than 2^p letters, a sufficiently long word, that will be pumped and eventually lead us to derive the required contradiction. Will any sufficiently long word do? No. It is quite possible that some words can indeed be broken up and pumped so that all new words are also in the language. We must choose a *suitable* word, one that can be used to derive a contradiction. How does one decide on a suitable word? Unfortunately there is no recipe for this choice. We must take the specific language into consideration. The good news is that it is only necessary to find one such word to derive a contradiction.

3. Suppose we have chosen a suitable word. The next step is the derivation of a contradiction. To do this we have to consider *every* permissible way in which the word can be broken up.

   - What does 'permissible' mean? In the case of the pumping lemma without length, i.e. Theorem 34, it means the word can be decomposed into five parts, $uvxyz$. In the case of the pumping lemma with length, Theorem 35, it means the word can be decomposed into five parts, $uvxyz$, such that

     $$length(vxy) \leq 2^p,$$
     $$length(x) > 0, \text{ and } length(v) + length(y) > 0.$$

     Because the conditions of the pumping lemma with length are more restrictive, it is usually easier to use; there will be fewer cases to consider.

   For *every* permissible way in which the word can be decomposed, we have to show that there is at least one word of the form uv^nxy^nz with $n > 1$, i.e. a word obtained by pumping, which is *not* in the language. Since the pumping lemmas tell us that this word should be in the language, this means that we have derived the required contradiction.

Example

Here is a straightforward example of how to apply one of the pumping lemmas. Suppose we are required to show that the language $L = \{a^n b^n a^n \mid n \geq 1\}$ is not context-free by using a pumping lemma.

The first step is to assume that the language L is context-free. This means there is a CFG in CNF with, say, p live productions that generates L. According to the pumping lemma with length, every word w in L with $length(w) > 2^p$ can be broken up into five parts $uvxyz$, such that

$$length(vxy) \leq 2^p,$$
$$length(x) > 0, \text{ and } length(v) + length(y) > 0$$

and for which all words of the form $uv^n xy^n z$ with $n > 1$ are in L.

The next step is to choose a suitable word. Let us choose $a^{2^p} b^{2^p} a^{2^p}$ which is in L and clearly has more than 2^p letters. Now we have to consider the permissible ways in which this word can be broken up. Since $length(vxy) \leq 2^p$, we know vxy is either contained in exactly one of the three clumps of 2^p letters or vxy straddles two adjacent clumps of 2^p letters. Consider the first case and suppose vxy is contained in the first clump of a's. Then the word $uvvxyyz$ will contain more than 2^p a's in the first clump which is (still) followed by a clump of 2^p b's and a clump of 2^p a's. So the new word $uvvxyyz$ is *not* in the language L. Note that this contradiction is not enough — we must get a contradiction in every possible case, thus we proceed. A similar argument holds if vxy is contained in the clump of 2^p b's or in the second clump of 2^p a's. Next, consider the case where vxy is contained in two adjacent clumps of 2^p letters, say the first clump of a's and the clump of b's. Then the word $uvvxyyz$ will *not* be in L because it will contain more than 2^p a's in the first clump of a's or more than 2^p b's or more than one occurrence of ab, with still only 2^p a's in the last clump of a's. A similar argument holds if vxy is contained in the last two adjacent clumps of letters.

We have thus shown that, for every permissible way in which the word $a^{2^p} b^{2^p} a^{2^p}$ can be decomposed into the five parts $uvxyz$, the word $uvvxyyz$ is not in L, which contradicts the pumping lemma. We are thus forced to retract our assumption that L is context-free and conclude that the language is non-context-free.

Will a proof that uses the pumping lemma without length differ much from the above? The structure of such a proof will not change substantially but we will have to think again about all the permissible ways in which the chosen word can be broken up because we cannot use the fact that $length(vxy) \leq 2^p$. You may take a look at *Cohen*'s proof that the language L given above is non-context-free — he uses the pumping lemma without length.

A Few Warnings

- Do not take a word with a specific (numerical) number of a's, b's and other letters when you are required to show that some given language is non-context-free. This means that in the example above you should *not* choose a word like $a^{101} b^{101} a^{101}$. Such a word can be used to illustrate a technique, et cetera, but is not adequate for a proof.

- Suppose we have to prove that some given language is non-context-free. Now suppose we consider a number of sufficiently long words, and for every one of them we are able to find a permissible way to break them up such that the new words found by pumping *are* also in the language. What happens now? Well, nothing really. It may simply mean that we have not yet found a suitable word. Suppose, on the other hand, that we are able to *prove* that, for every sufficiently long word, it is possible to find a permissible way to break it up so that all pumped words are also in the language. Have we then shown that the language is indeed context-free? NO. We still do *not* know whether the language is context-free or not. The pumping lemmas simply state that context-free languages have certain properties; they do *not* state that non-context-free do *not* have these properties.

- Let us now look at an example in order to show that we have to be very careful when choosing the word to be pumped. Consider the language VERYEQUAL defined over the alphabet $\Sigma = \{a,b,c\}$. This language consists of all words with, in total, an equal number of a's, b's and c's, excluding the empty word Λ.

Suppose we want to prove that this language is not context-free and have proceeded as above and are now at the point of choosing a suitable word. Consider the word $(abc)^{2p}$ which is in VERYEQUAL and clearly has more than 2^p letters. Can we use it to derive a contradiction? No, we cannot. It is easy to see that

$$u = v = \Lambda \text{ and } x = y = abc \text{ and } z = (abc)^{2p-2}$$

is a permissible way to break up the word $(abc)^{2p}$, and also that all words of the form uv^nxy^nz with $n > 1$ are in VERYEQUAL. Thus we have a permissible way to break up the chosen (sufficiently long) word which does not lead to a contradiction. The word $(abc)^{2p}$ is therefore a bad choice. There are, however, other words that do lead to a contradiction, for example the word $a^{2p}b^{2p}c^{2p}$. You may like to try it.

- Directly following Theorem 35, *Cohen* shows that the pumping lemma without length is sometimes not strong enough to prove that a language is non-context-free. Will the pumping lemma with length always be powerful enough? No. Sometimes, however, it is possible to add other bits of information and then use the pumping lemma with length. As an example, consider the language

$$L_1 = \{a^m b^n a^n b^n \mid n \geq 1, m \geq 1\}.$$

Let us try to show that L_1 is non-context-free in the usual way. We assume it is context-free, i.e. is generated by a CFG in CNF with, say, p live productions. By the pumping lemma with length every word w for which $length(w) > 2^p$ can be broken up into five strings $uvxyz$ such that

$$length(vxy) \leq 2^p,$$
$$length(x) > 0, \text{ and } length(v) + length(y) > 0.$$

and for which all words of the form uv^nxy^nz, $n > 1$, are in L_1. Now we have to choose a suitable word. The word $a^{2p}b^{2p}a^{2p}b^{2p}$ is a bad idea, because there is a permissible way to break it up into parts so that all pumped words are in L_1: if we let $u = \Lambda$, $v = a^{2p-1}$, $x = a$, $y = \Lambda$ and $z = b^{2p}a^{2p}b^{2p}$, then pumping the word just means that we increase the number of a's in the first clump of a's, so all words of the form uv^nxy^nz, $n > 1$, are in L_1. What is worse, is that this argument holds for *every* word in L_1: for any word $a^m b^n a^n b^n$, where $m,n \geq 1$, of the language, if we set $u = a^{m-1}$, $v = a$, $x = b$, $y = \Lambda$ and $z = b^{n-1}a^n b^n$, then this is a permissible way to decompose the word and all words of the form uv^nxy^nz, $n > 1$, are in L_1.

From the above it seems as if we cannot use the pumping lemma with length to prove that the language L_1 is non-context-free. But we actually can. Consider the following argument:

– Assume L_1 is context-free. Then the language $L_2 = \{b^n a^n b^n \mid n \geq 1\}$ will also be context-free. Why? Well, since L_1 is context-free, there must be a PDA that accepts it. Such a PDA will start off by reading the first clump of a's and then proceed to read the last three clumps of letters. Moreover, there is no connection between the number of letters in the first clump of a's and the number of letters in the last three clumps of letters. So there is a PDA which accepts L_1 and which reads the first clump of a's *without putting any of them on the stack*. Now take this PDA and simply throw away the part that reads the first clump of a's. We end up with a PDA which accepts the language L_2, thus L_2 must be context-free. So, if we now show that L_2 actually is non-context-free, then we will have derived a contradiction and we will have to retract the assumption that the language L_1 is context-free. It is possible to show that L_2 is non-context-free. Either of the pumping lemmas may be used. We leave the proof to you.