Using the Pumping Lemma for Non-context-free Languages

Proof Structure

The Sipser textbook gives proofs showing that the following languages are not context-free:

- $B = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 0\}$
- $C = \{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid 0 \leq i \leq j \leq k\}$
- $D = \{ww \mid w \in \{\mathtt{0},\mathtt{1}\}^*\}$

Below are some additional examples of proofs showing that a language L is not context-free. The general structure of each proof is as follows (it looks very similar to the structure of proofs showing that a language is not regular):

1. Assume to the contrary that L is a CFL, in which case it would have a pumping length p with all the properties described by the pumping lemma for CFLs (Sipser p. 125).
2. Choose a string $s \in L$ that is at least p symbols long. Depending on your choice of s, the steps below may be easier, harder, or impossible. If it appears that your choice of s results in a case that can always be pumped, try to select a new s that avoids the problem and try again.
3. Identify all ways of selecting a pair of substrings v and y in s (so that $s = uvxyz$), subject to the constraints $|vy| > 0$ and $|vxy| \leq p$. Much of the work of the proof consists of identifying these cases. It is essential that the cases you present in your proof are *exhaustive*, meaning they truly cover all possible ways of selecting substrings v and y in s. The constraint $|vy| > 0$ means that v or y can be empty, but not both at the same time (the pumping lemma would always be vacuously true otherwise). The constraint $|vxy| \leq p$ is very useful for limiting the number of cases you need to consider, because although vxy may occur anywhere within s, the v and y cannot be arbitrarily far apart, but must occur within a sliding window p symbols wide.
   (a) For each case of v, y, prove that there is a way of pumping it (up or down, your choice) that results in a string not in L. For each case of v, y, you can pump in a different way; that is, you do not need to pump up or pump down for all cases. See Sipser's Example 2.37 for an example where pumping differently in each case is necessary.
4. Since all of the cases for v, y in s fail to be pumped in some way, L does not have a pumping length, so it is not context-free.

Examples

1. $L = \{\mathtt{0}^n\mathtt{\#}\mathtt{0}^{2n}\mathtt{\#}\mathtt{0}^{3n} \mid n \geq 0\}$

   We will prove by contradiction that L is not context-free. Suppose that L is a CFL. Then it has a pumping length p with all the properties described by the pumping lemma for CFLs. Let $s = \mathtt{0}^p\mathtt{\#}\mathtt{0}^{2p}\mathtt{\#}\mathtt{0}^{3p}$ (setting $n = p$ seems like a natural first choice, and in fact for this problem it will work out well). The cases for partitioning $s = uvxyz$ are as follows:

(a) At least one of v or y contains a `#`. Pumping down to $uv^0xy^0z = uxz$ would delete at least one `#` and result in a string with fewer than two `#`'s, which would not be in L. (We could also choose to pump up to, say, uv^2xy^2z, in which case the string would have at least one too many `#`'s.)

(b) Both v and y consist only of `0`'s. Note that this is just the opposite of the above, so these two cases combined are exhaustive. There are three main sections in s: the first $\mathtt{0}^p$, the middle $\mathtt{0}^{2p}$, and the last $\mathtt{0}^{3p}$. Because v and y do not contain `#`, each one must lie entirely within one section. Together, v and y are in at most two different sections. If we pump down to uxz, it will change the number of `0`'s in one or two sections but not the third, so the ratio of `0`'s will be violated, and the string will not be in L.

Since the above cases are exhaustive and each case fails to be pumped in some way, L does not have a pumping length p, so it is not context-free.

2. $L = \{\mathtt{a}^n\mathtt{b}^m\mathtt{a}^n \mid n, m \geq 0 \text{ and } n \geq m\}$

Whenever we have a language with inequality constraints, it is often useful to choose an s that is right at the boundary of the constraints. There are languages where only a string at the boundary will work, and there are other languages where the string doesn't need to be at the boundary to show that the pumping lemma fails. For this language, all strings at the boundary fail the pumping lemma. Other strings in L fail the pumping lemma when m is sufficiently large (at least $p - 1$).

Let $s = \mathtt{a}^p\mathtt{b}^p\mathtt{a}^p$. There are three main sections, and because $|vxy| \leq p$, v and y can be in sections 1 and 2, sections 2 and 3, but not sections 1 and 3. The cases are as follows:

(a) v or y straddles the boundary between two sections and contains both `a`'s and `b`'s. Pumping up would result in `a`'s and `b`'s out of order. If is often useful to eliminate cases where v or y straddle boundaries in this way to simplify your thought process for the remaining cases.

(b) v or y contains at least one `a` from section 1. Because $|vxy| \leq p$, neither v nor y can be in section 3, so pumping down causes a mismatch between sections 1 and 3.

(c) v or y contains at least one `a` from section 3. Similarly to the previous case, pumping down causes a mismatch between sections 1 and 3.

(d) v and y are both in section 2 and consist only of zero or more `b`'s (again, this case is just the remainder of all of the cases above, so all cases combined are exhaustive). Pumping up violates $n \geq m$.

Because all the cases above fail, L is not context-free. Note that case (a) could actually be eliminated, and the proof would still be correct.

3. $L = \{wtw^R \mid w, t \in \{\mathtt{a}, \mathtt{b}\}^* \text{ and } |w| = |t|\}$

Intuitively, this language is not context-free because it requires us to match pairs of things twice. With the power of CFLs, we can check that w matches w^R, and we can check that $|w| = |t|$, but we cannot do both. To choose a good s, first observe that making w and t similar and with no pattern is not a good idea. Also, if t is symmetrical, the pumping lemma is unlikely to fail regardless of w. For example, when $s = \mathtt{a}^p\mathtt{b}^p\mathtt{a}^p$, a careful analysis of the cases for v and y shows that it can *always* be pumped when $v = \mathtt{a}$ and $y = \mathtt{aa}$, because the pumped string will always be symmetrical and the total length of the string will always be a multiple of 3. Instead, we can choose $s = \mathtt{a}^p\mathtt{b}^p\mathtt{a}^p\mathtt{b}^p\mathtt{b}^p\mathtt{a}^p$. It is not difficult to see that if we pump up or down within any window of p symbols in s, the result will no longer be in the language.

Acknowledgements

This document was based on an earlier document written by Dr. Marina Blanton.