Spring, 2014 CIT 596

Theory of Computation

Pumping Lemma for CFG notes

The pumping lemma for context free languages is similar to the one for regular languages in that it is used typically to prove something is not context free.

Just like the proofs of a language being not regular, the usage of the pumping lemma for proving something to be non context free involves the following

- Consider any arbitrary p.
- Pick a string **in the language** that is longer than p. Note that this means the string you pick has to be dependent on p in some manner.
- Now split the chosen string in 5 parts. $uvxyz = s$. The constraints are v and y cannot both be empty and $|vxy| \leq p$. Note that you have consider all possible splits and in almost all questions/problems this will involve breaking this down into a few different cases.
- Pump the string either up or down. Pumping up means considering strings uv^2xy^2z then uv^3xy^3z and so on. Pumping down means considering the string uxz. If any of those 'pumped' strings is not in the language we are done.

To make this clear, let us consider the examples in the book and try and reason about them in detail.

1. $P = \{a^i b^i c^i | i \geq 0\}$

   Assume the language is context free. Then there must be some pumping length p. Now we must choose some string that is in the language and longer than p. In this case, the appropriate choice seems to be $a^p b^p c^p$. Let us see how we apply the pumping lemma to this string.

   We have to consider all possible ways of splitting and divide them up into cases. The following are the cases

   (a) v and y both contain only type of symbol. Without loss of generality (because the other cases will be similar), consider $v = b^k$, where $0 < k \leq p$. Then uv^2xy^2z will have p as but more than p bs and therefore cannot be in the language.

   As said before, the other cases of v and y containing the same symbol work the same way.

(b) v and y have a combination of 2 symbols. Again, without loss of generality (the other cases can be reasoned about in a similar manner) let us say v is $a^k b^l$ and y is $b^m c^n$. Then when we consider uv^2xy^2z we get a string of the form $a^{p-k}a^k b^l a^k b^l x b^m c^n b^m c^n c^{p-n}$ which is not in the form of strings that are in the language since it does not have a block of as followed by a block of bs followed by a block of cs.

Therefore regardless of which way we split the string into u, v, x, y, z, there is no way we can ensure that every pumped string continues to lie in the language. This completes the proof that the language P is not context free.

2. $Q = \{a^i b^j c^k | 0 \leq i \leq j \leq k\}$.

Again consider any p and we need to figure out a string that we might be unable to pump. Once again the obvious choice is $a^p b^p c^p$.

Again, we have to consider every possible way in which this string can be split into 5 parts while adhering to the constraints $|vy| > 0$ and $|vxy| \leq p$.

(a) As before if v and y contain a mix of symbols then pumping up will result in a string that does not have the same pattern.

To clarify, consider for instance the case when v has some bs and some c. Or more formally $v = b^k c^l$. Then when pumping up once (uv^2xy^2z), we will get a block of bs followed by a block of cs followed by another block of bs, of the form $a^p b^k c^l b^k c^l xyyz$.

(b) Now consider the cases when v and y contain the same symbols. There are a fair number of subcases here which we will analyze one by one

- v and y contain only as. Then pumping up, $s_1 = uv^2xy^2z$ will increase the number of as while keeping the number of bs and cs the same. Therefore the string s_1 is not in Q.
- v contains only as, y contains only bs. Consider then uv^2xy^2z. Since both v and y are not allowed to the empty string at the same time, either the number of as or the number of bs will be more than p while the number of cs will stay the same. So again, when pumping up we get a string that is not in Q.
- v and y contains only bs. Again, pumping up and considering the string uv^2xy^2z. This string contains more bs than the original string while still having the same number of as and cs. This pumped string therefore cannot be in the language Q.
- v and y contain only cs. This case is interesting since pumping up will increase the number of cs, which does not cause any issues since all strings that have more cs than as and bs is still in the language Q.
  Then we try pumping down. Consider what happens to the string uxz. Since we cannot have both v and y be the empty string, this results in a reduction in

the number of cs which maintaining the same number of as and bs. Therefore the pumped down string cannot be in Q.
So for this case pumping down works.

- v contains only bs and y contains only cs. Again, pumping down is the method to use because this would mean you reduce either the number of bs or cs while maintaining the number of as intact.
- v is the empty string and y contains only cs. Once again pumping down will work since it results in a reduction of the number of cs while maintaining the same number of as and bs.
- v contains only as and y is the empty string. uv^2xy^2z will have more as than bs and hence cannot be in the language Q.
- Can we have v be only as and y be only cs? That would be impossible because $|vxy| \leq p$ can never be satisfied in that case since the block of bs is of length p. So we do not need to consider that case.

That exhausts all possible cases of splitting the string $a^p b^p c^p$. We did not manage to successfully pump the string regardless of how we tried our splitting. Therefore this language is not context free.

Note that we did this example a little differently from the way it is done in Sipser's book. If you can think of the cases by dividing them into three cases based on which letter is not appearing, then read the book's explanation. We just wanted to show you how to approach this problem even more directly.

(c) $D = \{ww | w \in \{0,1\}^*\}$