Pumping lemma for context free grammars

This post is meant to complement the chapter on pumping lemma for context free grammars in the textbook, but not to replace it. You should read this first to keep the relevant example in mind when you read the formal proof from the textbook.

The main idea

As an example, we will consider the following context free grammar and the string $acegfdb$ (that belongs to the language generated by the grammar):

$$A \to aRb$$

$$R \to cBd \mid g$$

$$B \to eRf$$

In the following derivation, R plays a special role: it generates a sub-string, $ceRfd$, which also contains R. This second R is used to generate the final terminals, which in this case is just g.

$$A \Longrightarrow a\mathbf{R}b \Longrightarrow acBdb \Longrightarrow a\mathbf{ceRfd}b \Longrightarrow acegfdb$$

Before leading to the terminals, we could have instead repeated the generation of $ceRfd$ as many times as we wish:

$$A \Longrightarrow aRb \Longrightarrow acBdb \Longrightarrow aceRfdb \Longrightarrow a(ce)^2R(fd)^2b \ldots \Longrightarrow a(ce)^nR(fd)^nb$$

before finally using R to lead to the terminal

$$\Longrightarrow a(ce)^n g(fd)^n b$$

So $a(ce)^2 g(fd)^2 b$, $a(ce)^3 g(fd)^3 b$, etc are also in our language:

In fact, we could have skipped the steps shown in bold

$$A \Longrightarrow aRb \Longrightarrow \mathbf{acBdb} \Longrightarrow \mathbf{aceRfdb} \Longrightarrow \mathbf{acegfdb}$$

and directly proceeded to

$$A \Longrightarrow a\mathbf{R}b \Longrightarrow agb$$

It is now easy to generalize these observations:

If a string s is generated in a series of steps that look like this:

$$S \ldots \Longrightarrow \ldots x_1 R x_2 \ldots \Longrightarrow \ldots x_1 \mathbf{y_1 R y_2} x_2 \ldots \Longrightarrow \ldots x_1 y_1 \mathbf{z} y_2 x_2 = s$$

and where both the substrings highlighted in bold are generated by the previous R, then s can be realized as a concatenation of 5 substrings as $s = x_1 y_1 z y_2 x_2$, so that the first R is responsible for $y_1 z y_2$ within which the second R is responsible for the substring, z. In that case, the language must also contain the strings $x_1 y_1^n z y_2^n x_2$ for each $n = 0, 1, \ldots$

In fact, if the string s is "long enough", that will always happen! We can see that by examining the parse tree of a string. R would have the required property if it occurred twice in one path of the parse tree. That in turn is ensured if the length of the path exceeds the number of variables (pigeon-hole principle!). What is long enough? The precise number is not really important. What is important is that there is such a number.

The statement

Given a context free grammar generating a language L, there is a number N called the pumping length, such that, whenever the length of a string s exceeds N, it can be broken up as, $s = x_1 y_1 z y_2 x_2$, so that the strings $x_1 y_1^n z y_2^n x_2$ also belong to L for each $n = 0, 1, \ldots$. We can make no assumptions on where the break up occurs, except for the fact that both y_1 and y_2 cannot be empty (if they could, the lemma willl be useless!), and that the substring $y_1 z y_2$ has length less than N.

As in the previous pumping lemma, you apply it by using some foresight to find, for each possible pumping length, a string of bigger length that belong to the language. You then show that for each such string, no matter how you break it up as $x_1y_1zy_2x_2$, at least for some n, $x_1y_1^nzy_2^nx_2$ does not belong the language.

Example

The language over the alphabet, $\Sigma = \{a, b, c\}$, consisting of strings of the form $a^nb^nc^n$ cannot be context free. To see this, observe that if the pumping length is N, then $a^Nb^Nc^N$ exceeds that length. If the language were context free, then there is some unknown way to break it up as $x_1y_1zy_2x_2$ so that $x_1y_1^nzy_2^nx_2$ belongs to the language for every n. Notice that if either y_1 or y_2 contains repetitions of more than one character, then its power will have at least some characters out of order (eg. $(aabb)^2 = aabbaabb$ has some a's following b's). So we only need to bother about the case where both contain repetitions of only one character. In that case $x_1y_1^nzy_2^nx_2$ will increase the powers of at most two characters whlie leaving the third untouched, therefore such a string cannot belong to the language where all three powers are required to be the same. :::