Lecture 22

The Pumping Lemma for CFLs

There is a pumping lemma for CFLs similar to the one for regular sets. It can be used in the same way to show that certain sets are not context-free. Here is the official version; there will also be a corresponding game with the demon that will be useful in practice.

Theorem 22.1 **(Pumping lemma for CFLs)** *For every CFL A, there exists $k \geq 0$ such that every $z \in A$ of length at least k can be broken up into five substrings $z = uvwxy$ such that $vx \neq \epsilon$, $|vwx| \leq k$, and for all $i \geq 0$, $uv^iwx^iy \in A$.*

Informally, for every CFL A, every sufficiently long string in A can be subdivided into five segments such that the middle three segments are not too long, the second and fourth are not both null, and no matter how many extra copies of the second and fourth you pump in simultaneously, the string stays in A.

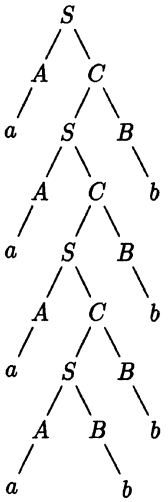
Note that this differs from the pumping lemma for regular sets in that we pump simultaneously on two substrings v and x separated by a substring w.

The key insight that gives this theorem is that for a grammar in Chomsky normal form, any parse tree for a very long string must have a very long path, and any very long path must have at least two occurrences of some

nonterminal. A *parse tree* or *derivation tree* of a string z is a tree representing the productions applied in a derivation of z from the start symbol S independent of the order of application. For example, consider the Chomsky grammar

$$S \to AC \mid AB, \qquad A \to a, \qquad B \to b, \qquad C \to SB$$

for $\{a^n b^n \mid n \geq 1\}$. Here is a parse tree for the string a^4b^4 in this grammar:



The productions can be applied in any order. For example, a leftmost derivation of a^4b^4 (always applying a production to the leftmost remaining nonterminal) would give

$$\begin{aligned} S &\to AC \to aC \to aSB \to aACB \to aaCB \to aaSBB \to aaACBB \\ &\to aaaCBB \to aaaSBBB \to aaaABBBB \to aaaaBBBB \\ &\to aaaabBBB \to aaaabbBB \to aaaabbbB \to aaaabbbb, \end{aligned}$$

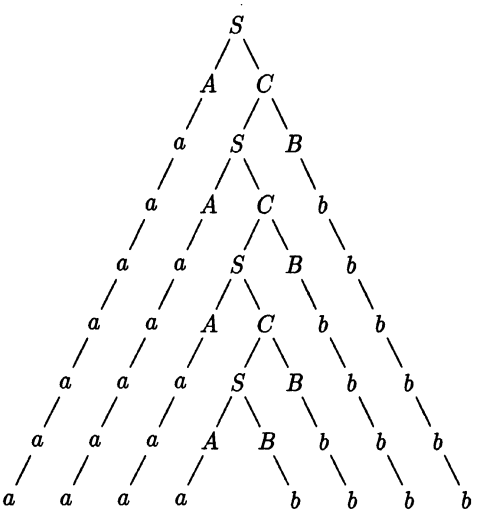
and a rightmost derivation would give

$$\begin{aligned} S &\to AC \to ASB \to ASb \to AACb \to AASBb \to AASbb \to AAACbb \\ &\to AAASBbb \to AAASbbb \to AAAABbbb \to AAAAbbbb \\ &\to AAAabbbb \to AAaabbbb \to Aaaabbbb \to aaaabbbb, \end{aligned}$$

but these two derivations have the same parse tree, namely the one pictured above.

Parse trees of Chomsky grammars for long strings must have long paths, because the number of symbols can at most double when you go down a level. This is because the right-hand sides of productions contain at most

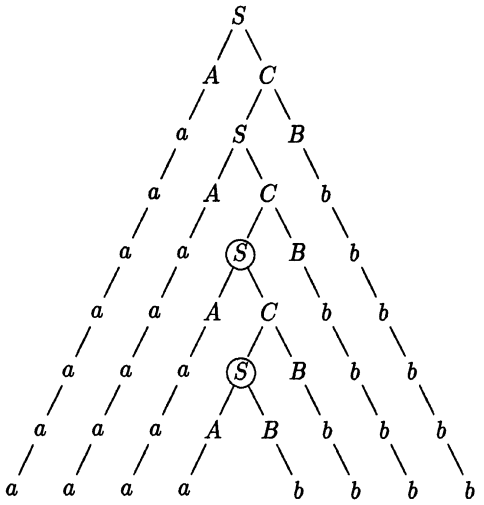
two symbols. For example, take the tree above and duplicate the terminals generated at each level on all lower levels, just to keep track of the symbols that have been generated so far:



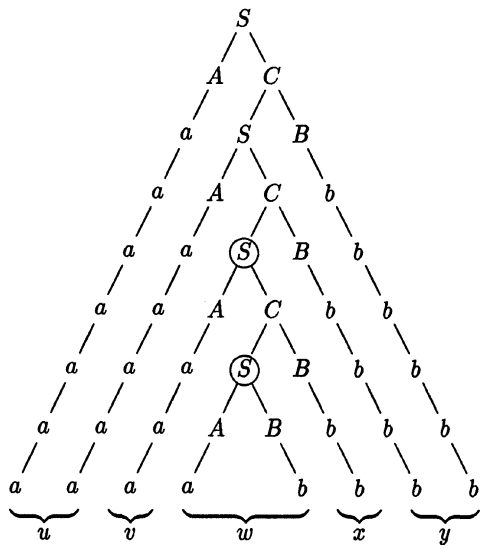
The number of symbols at each level is at most twice the number on the level immediately above. Thus at the very most, we can have one symbol at the top level (level 0), 2 at level 1, 4 at level 2, ..., 2^i at level i. In order to have at least 2^n symbols at the bottom level, the tree must be of depth[1] at least n; that is, it must have at least $n+1$ levels.

Proof of the pumping lemma. Let G be a grammar for A in Chomsky normal form. Take $k = 2^{n+1}$, where n is the number of nonterminals of G. Suppose $z \in A$ and $|z| \geq k$. By the argument above, any parse tree in G for z must be of depth at least $n+1$. Consider the longest path in the tree. (In the example above, the path from S at the root down to the leftmost b in the terminal string is such a path.) That path is of length at least $n+1$, therefore must contain at least $n+1$ occurrences of nonterminals. By the pigeonhole principle, some nonterminal occurs more than once along the path. Take the first pair of occurrences of the same nonterminal along the path, reading from bottom to top. In the example above, we would take the two circled occurrences of S:

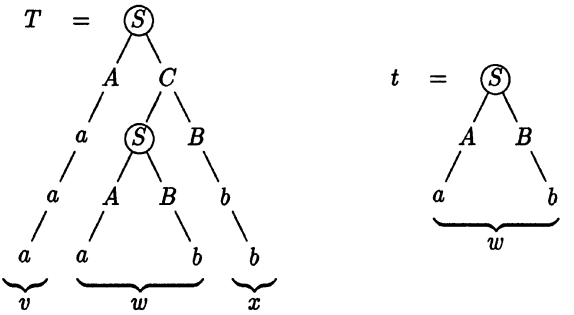
[1]The *depth* is the number of edges on the longest path from the root to a leaf.



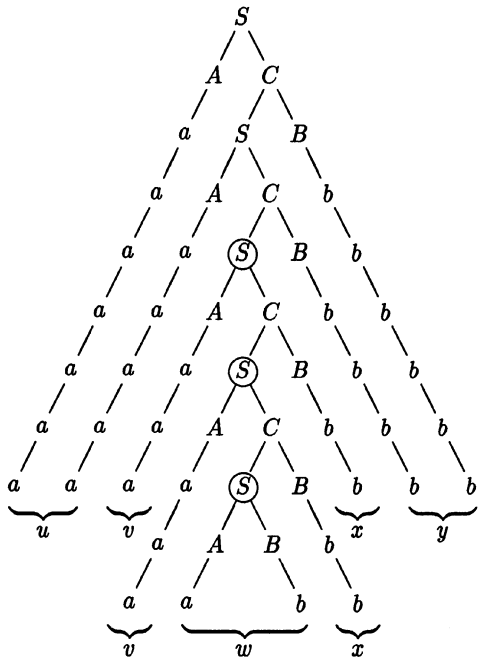
Say X is the nonterminal with two occurrences. Break z up into substrings $uvwxy$ such that w is the string of terminals generated by the lower occurrence of X and vwx is the string generated by the upper occurrence of X. In our running example, $w = ab$ is the string generated by the lower occurrence of S and $vwx = aabb$ is the string generated by the upper occurrence of S:



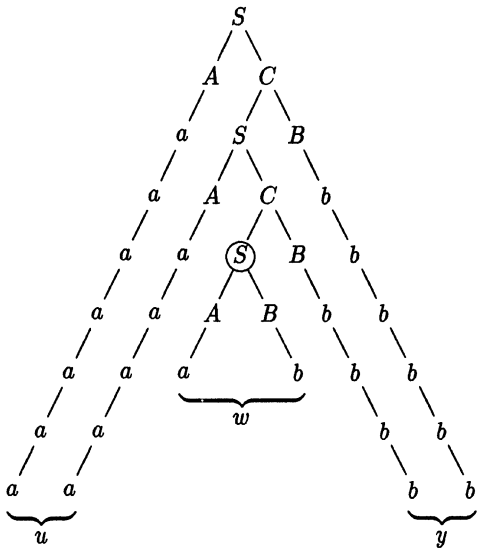
Thus in this example we have $u = aa$, $v = a$, $w = ab$, $x = b$, and $y = bb$. Let T be the subtree rooted at the upper occurrence of X and let t be the subtree rooted at the lower occurrence of X. In our example,



By removing t from the original tree and replacing it with a copy of T, we get a valid parse tree of uv^2wx^2y:



We can repeat this cutting out of t and replacing it with a copy of T as many times as we like to get a valid parse tree for uv^iwx^iy for any $i \geq 1$. We can even cut T out of the original tree and replace it with t to get a parse tree for $uv^0wx^0y = uwy$:



Note that $vx \neq \epsilon$; that is, v and x are not both null.

We also have $|vwx| \leq k$, since we chose the first repeated occurrence of a nonterminal reading from the bottom, and we must see such a repetition by the time we get up to height $n+1$. Since we took the longest path in the tree, the depth of the subtree under the upper occurrence of the repeated nonterminal X is at most $n+1$, therefore can have no more than $2^{n+1} = k$ terminals. □

Games with the Demon

Like its regular cousin, the pumping lemma for CFLs is most useful in its contrapositive form. In this form, it states that in order to conclude that A is *not* context-free, it suffices to establish the following property:

Property 22.2 *For all $k \geq 0$, there exists $z \in A$ of length at least k such that for all ways of breaking z up into substrings $z = uvwxy$ with $vx \neq \epsilon$ and $|vwx| \leq k$, there exists an $i \geq 0$ such that $uv^iwx^iy \notin A$.*

Property 22.2 is equivalent to saying that you have a winning strategy in the following game with the demon:

1. The demon picks $k \geq 0$.
2. You pick $z \in A$ of length at least k.

3. The demon picks strings u, v, w, x, y such that $z = uvwxy$, $|vx| > 0$, and $|vwx| \leq k$.

4. You pick $i \geq 0$. If $uv^i wx^i y \notin A$, then you win.

If you want to show that a given set A is not context-free, it suffices to show that you have a winning strategy in this game; that is, no matter what the demon does in steps 1 and 3, you have moves in steps 2 and 4 that can beat him.

Example 22.3 Let's use the pumping lemma to show that the set

$$A = \{a^n b^n a^n \mid n \geq 0\}$$

is not context-free. We'll do this by showing that we can always win the game with the demon.

Say the demon picks k in step 1. You have to argue that you can win no matter what k is. A good choice for you in step 2 is to pick $z = a^k b^k a^k$. Then $z \in A$ and $|z| = 3k \geq k$. Then in step 3, say the demon picks u, v, w, x, y such that $z = uvwxy$, $vx \neq \epsilon$, and $|vwx| \leq k$. You pick $i = 2$. In every case, you win: if the demon picked either v or x to contain at least one a and at least one b, then uv^2wx^2y is not of the form $a^*b^*a^*$, hence certainly not in A; if the demon picked v and x to contain only a's, then uv^2wx^2y has more than twice as many a's as b's, hence is not in A; if the demon picked v and x to contain only b's, then uv^2wx^2y has fewer than twice as many a's as b's, hence is not in A; and finally, if one of v or x contains only a's and the other contains only b's, then uv^2wx^2y cannot be of the form $a^m b^m a^m$, hence is not in A. In all cases you can ensure $uv^2wx^2y \notin A$, so you have a winning strategy. By the pumping lemma, A is not context-free. □

Example 22.4 Let's use the pumping lemma to show that the set

$$A = \{ww \mid w \in \{a, b\}^*\}$$

is not context-free. Since the family of CFLs is closed under intersection with regular sets (Homework 7, Exercise 2), it suffices to show that the set

$$\begin{aligned} A' &= A \cap L(a^*b^*a^*b^*) \\ &= \{a^n b^m a^n b^m \mid m, n \geq 0\} \end{aligned}$$

is not context-free.

Say the demon picks k. You pick $z = a^k b^k a^k b^k$. Call each of the four substrings of the form a^k or b^k a *block*. Then $z \in A'$ and $|z| \geq k$. Say the demon picks u, v, w, x, y such that $z = uvwxy$, $vx \neq \epsilon$, and $|vwx| < k$. No matter what the demon does, you can win by picking $i = 2$:

- If one of v or x contains both a's and b's (i.e., if one of v or x straddles a block boundary), then uv^2wx^2y is not of the form $a^*b^*a^*b^*$, thus is not in A'.
- If v and x are both from the same block, then uv^2wx^2y has one block longer than the other three, therefore is not in A'.
- If v and x are in different blocks, then the blocks must be adjacent; otherwise $|vwx|$ would be greater than k. Thus one of the blocks containing v or x must be a block of a's and the other a block of b's. Then uv^2wx^2y has either two blocks of a's of different size (if vx contains an a) or two blocks of b's of different size (if vx contains a b) or both. In any case, uv^2wx^2y is not of the form $a^nb^ma^nb^m$.

Since you can always ensure a win by playing this strategy, A' (and therefore A) is not a CFL by the pumping lemma.

Surprisingly, the complement of A, namely

$$\{a,b\}^* - \{ww \mid w \in \{a,b\}^*\},$$

is a CFL. Here is a CFG for it:

$$S \to AB \mid BA \mid A \mid B,$$
$$A \to CAC \mid a,$$
$$B \to CBC \mid b,$$
$$C \to a \mid b.$$

This grammar generates

(i) all strings of odd length (starting with productions $S \to A$ and $S \to B$); or

(ii) strings of the form $xayubv$ or $ubvxay$, where $x, y, u, v \in \{a,b\}^*$, $|x| = |y|$, and $|u| = |v|$.

The nonterminal A generates all strings of the form xay, $|x| = |y|$. The nonterminal B generates all strings of the form ubv, $|u| = |v|$. No string of the form (i) can be of the form ww, since ww is always of even length. No string of the form (ii) can be of the form ww, since there are occurrences of a and b separated by a distance of $n/2$, where n is the length of the string.

This example shows that the family of CFLs is not closed under complement. □

Note that in both these examples, your choice of $i = 2$ in step 4 was independent of the demon's move in step 3. This may not always be possible! However, keep in mind that you have the freedom to pick i in step 4 *after* you have seen what the demon did in step 3.

Historical Notes

The pumping lemma for CFLs is due to Bar-Hillel, Perles, and Shamir [8]. A somewhat stronger version was given by Ogden [96].