Lecture 10

Proving languages to be non-context free

In this lecture we will study a method through which certain languages can be proved to be non-context free. The method will appear to be quite familiar, because it closely resembles the one we discussed in Lecture 5 for proving certain languages to be nonregular.

10.1 The pumping lemma for context-free languages

Along the same lines as the method we discussed in Lecture 5 for proving some languages to be nonregular, we will start with a variant of the pumping lemma that holds for context-free languages.

The proof of this lemma is, naturally, different from the proof of the pumping lemma for regular languages, but there are similar underlying ideas. The main idea is that if you have a parse tree for the derivation of a particular string by some context-free grammar, and the parse tree is sufficiently deep, then there must be a variable that appears multiple times on some path from the root to a leaf—and by modifying the parse tree in certain ways, one obtains a similar type of pumping effect that we had in the case of the pumping lemma for regular languages.

Lemma 10.1 (Pumping lemma for context-free languages)**.** *Let Σ be an alphabet and let $A \subseteq \Sigma^*$ be a context-free language. There exists a positive integer n (called a* pumping length *of A) that possesses the following property. For every string $w \in A$ with $|w| \geq n$, it is possible to write $w = uvxyz$ for some choice of strings $u, v, x, y, z \in \Sigma^*$ such that*

1. *$vy \neq \varepsilon$,*
2. *$|vxy| \leq n$, and*
3. *$uv^i xy^i z \in A$ for all $i \in \mathbb{N}$.*

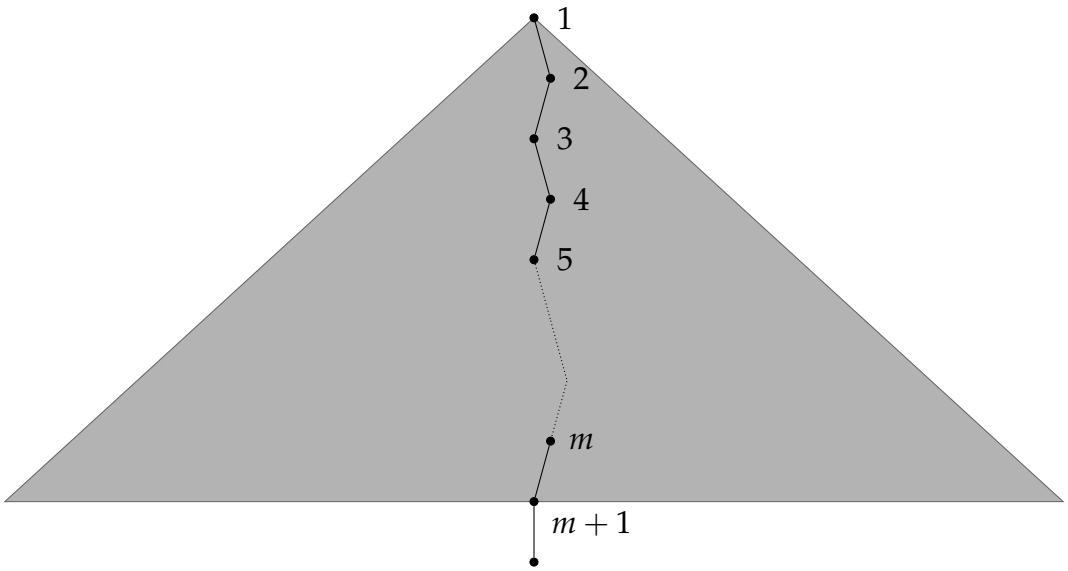

Figure 10.1: At least one path from the root to a leaf in a CNF parse tree for a string of length 2^m or more must have $m + 1$ or more variable nodes. If this were not so, the total number of variable nodes (which are collectively represented by the shaded region) would be at most $2^m - 1$, contradicting the fact that there must be at least 2^m variable nodes.

Proof. Given that A is context free, we know that there must exist a CFG G in Chomsky normal form such that $A = \mathrm{L}(G)$. Let m be the number of variables in G. We will prove that the property stated in the lemma holds for $n = 2^m$.

Suppose that a string $w \in A$ satisfies $|w| \geq n = 2^m$. As G is in Chomsky normal form, every parse tree for w has exactly $2|w| - 1$ variable nodes and $|w|$ leaf nodes. Hereafter let us fix any one of these parse trees, and let us call this tree T. For the sake of this proof, what is important about the size of T is that the number of variable nodes is at least 2^m. This is true because $2|w| - 1 \geq 2 \cdot 2^m - 1 \geq 2^m$. In fact, the last inequality must be strict because $m \geq 1$, but this makes no difference to the proof. Because the number of variable nodes in T is at least 2^m, there must exist at least one path in T from the root to a leaf along which there are at least $m + 1$ variable nodes—for if all such paths had m or fewer variable nodes, there could be at most $2^m - 1$ variable nodes in the entire tree.

Next, choose any path in T from the root to a leaf having the maximum possible length. (There may be multiple choices, but any one of them is fine.) We know that at least $m + 1$ variable nodes must appear in this path, as argued above—and because there are only m different variables in total, there must be at least one variable that appears multiple times along this path. In fact, we know that some

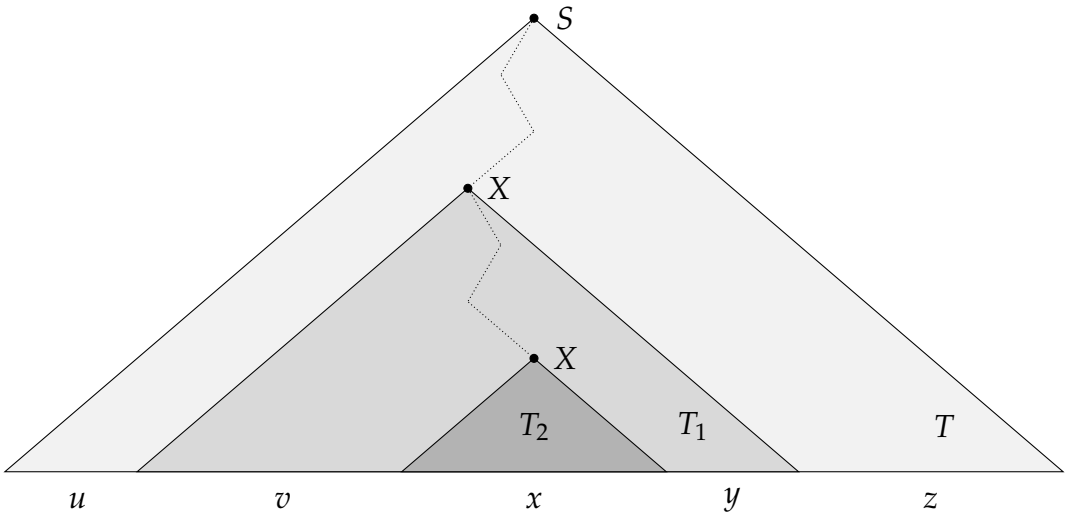

Figure 10.2: An illustration of the subtrees T_1 and T_2 of T.

variable (let us call it X) must appear at least twice within the $m+1$ variable nodes closest to the leaf on the path we have selected. Let T_1 and T_2 be the subtrees of T rooted at these two bottom-most occurrences of this variable X, with T_2 being the smaller of these two trees. By the way we have chosen these subtrees, we know that T_2 is a proper subtree of T_1, and T_1 is not very large: every path from the root of the subtree T_1 to one of its leaves can have at most $m+1$ variable nodes, and therefore T_1 has no more than $2^m = n$ leaf nodes.

Now, let x be the string for which T_2 is a parse tree (starting from the variable X) and let v and y be the strings formed by the leaves of T_1 to the left and right, respectively, of the subtree T_2, so that vxy is the string for which T_1 is a parse tree (also starting from the variable X). Finally, let u and z be the strings represented by the leaves of T to the left and right, respectively, of the subtree T_1, so that $w = uvxyz$. Figure 10.2 provides an illustration of the strings u, v, x, y, and z and how they related to the trees T, T_1, and T_2.

It remains to prove that u, v, x, y, and z have the properties required by the statement of the lemma. Let us first prove that $uv^ixy^iz \in A$ for all $i \in \mathbb{N}$. To see that $uxz = uv^0xy^0z \in A$, we observe that we can obtain a valid parse tree for uxz by replacing the subtree T_1 with the subtree T_2, as illustrated in Figure 10.3. This replacement is possible because both T_1 and T_2 have root nodes corresponding to the variable X. Along similar lines, we have that $uv^2xy^2z \in A$ because we can obtain a valid parse tree for this string by replacing the subtree T_2 with a copy of T_1, as suggested by Figure 10.4. By repeatedly replacing T_2 with a copy of T_1, a valid parse tree for any string of the form uv^ixy^iz is obtained.

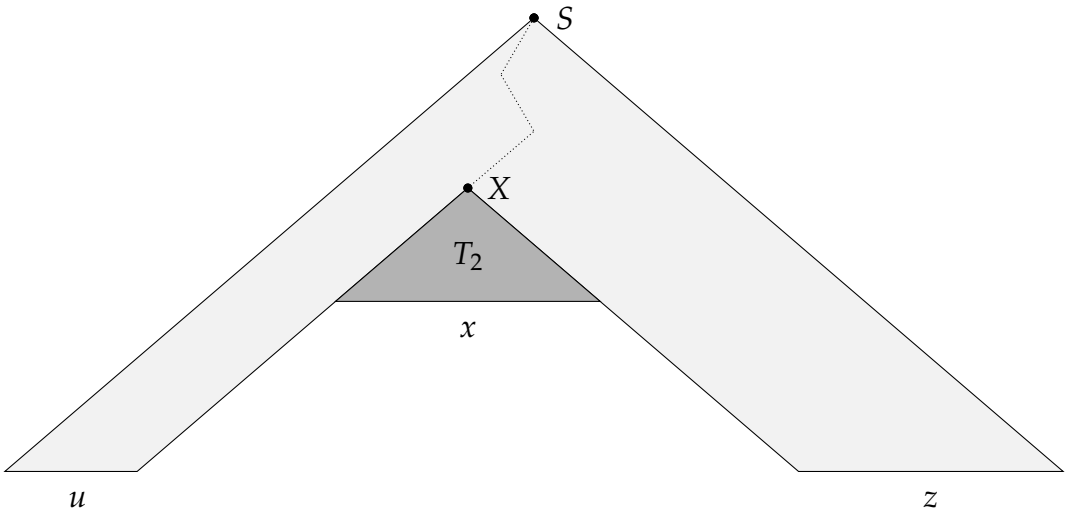

Figure 10.3: By replacing the subtree T_1 by the subtree T_2 in T, a parse tree for the string $uxz = uv^0xy^0z$ is obtained.

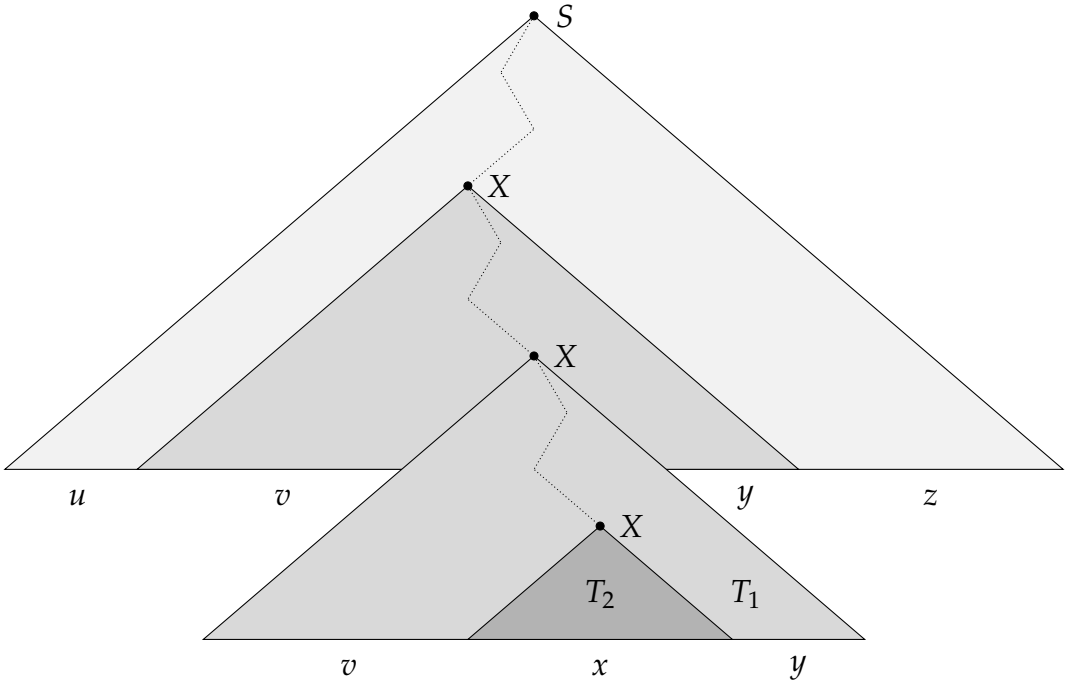

Figure 10.4: By replacing the subtree T_2 by the subtree T_1 in T, a parse tree for the string uv^2xy^2z is obtained. By repeatedly replacing T_2 with T_1 in this way, a parse tree for the string uv^ixy^iz is obtained for any positive integer $i \geq 2$.

Next, the fact that $vy \neq \varepsilon$ follows from the fact that every parse tree for a string corresponding to a CFG in Chomsky normal form has the same size. It therefore cannot be that the parse tree suggested by Figure 10.3 generates the same string as the one suggested by Figure 10.2, as the two trees have differing numbers of variable nodes. This implies that $uvxyz \neq uxz$, so $vy \neq \varepsilon$.

Finally, we have $|vxy| \leq n$ because the subtree T_1 has at most $2^m = n$ leaf nodes, as was already argued above. □

10.2 Using the context-free pumping lemma

Now that we have the pumping lemma for context-free languages in hand, we can prove that certain languages are not context free. The methodology is very similar to what we used in Lecture 5 to prove some languages to be nonregular. Some examples, stated as propositions, follow.

Proposition 10.2. *Let $\Sigma = \{0, 1, 2\}$ and let A be a language defined as follows:*

$$A = \left\{0^m 1^m 2^m \,:\, m \in \mathbb{N}\right\}. \tag{10.1}$$

The language A is not context free.

Proof. Assume toward contradiction that A is context free. By the pumping lemma for context-free languages, there must exist a pumping length n for A. We will fix such a pumping length n for the remainder of the proof.

Let

$$w = 0^n 1^n 2^n. \tag{10.2}$$

We have that $w \in A$ and $|w| = 3n \geq n$, so the pumping lemma guarantees that there must exist strings $u, v, x, y, z \in \Sigma^*$ so that $w = uvxyz$ and the three properties in the statement of that lemma hold: (i) $vy \neq \varepsilon$, (ii) $|vxy| \leq n$, and (iii) $uv^i xy^i z \in A$ for all $i \in \mathbb{N}$.

Now, given that $|vxy| \leq n$, it cannot be that the symbols 0 and 2 both appear in the string vy; the 0s and 2s are too far apart for this to happen. On the other hand, at least one of the symbols of Σ must appear within vy, because this string is nonempty. This implies that the string

$$uv^0 xy^0 z = uxz \tag{10.3}$$

must have strictly fewer occurrences of either 1 or 2 than 0, or strictly fewer occurrences of either 0 or 1 than 2. That is, if the symbol 0 does not appear in vy, then it must be that either

$$|uxz|_1 < |uxz|_0 \quad \text{or} \quad |uxz|_2 < |uxz|_0, \tag{10.4}$$

and if the symbol 2 does not appear in vy, then it must be that either

$$|uxz|_0 < |uxz|_2 \quad \text{or} \quad |uxz|_1 < |uxz|_2. \tag{10.5}$$

This, however, is in contradiction with the fact that $uv^0xy^0z = uxz$ is guaranteed to be in A by the third property.

Having obtained a contradiction, we conclude that A is not context free, as claimed. □

In some cases, such as the following one, a language can be proved to be non-context free in almost exactly the same way that it can be proved to be nonregular.

Proposition 10.3. *Let $\Sigma = \{0\}$ and recall the language*

$$\text{SQUARE} = \left\{0^{m^2} \,:\, m \in \mathbb{N}\right\} \tag{10.6}$$

defined in Lecture 5. The language SQUARE *is not context free.*

Proof. Assume toward contradiction that SQUARE is context free. By the pumping lemma for context-free languages, there must exist a pumping length $n \geq 1$ for SQUARE for which the property stated by that lemma holds. We will fix such a pumping length n for the remainder of the proof.

Define

$$w = 0^{n^2}. \tag{10.7}$$

We see that $w \in$ SQUARE and $|w| = n^2 \geq n$, so the pumping lemma tells us that there exist strings $u, v, x, y, z \in \Sigma^*$ so that $w = uvxyz$ and the following conditions hold:

1. $vy \neq \varepsilon$,
2. $|vxy| \leq n$, and
3. $uv^ixy^iz \in$ SQUARE for all $i \in \mathbb{N}$.

There is only one symbol in the alphabet Σ, so it is immediate that $vy = 0^k$ for some choice of $k \in \mathbb{N}$. Because $vy \neq \varepsilon$ and $|vy| \leq |vxy| \leq n$ it must be the case that $1 \leq k \leq n$. Observe that

$$uv^ixy^iz = 0^{n^2+(i-1)k} \tag{10.8}$$

for each $i \in \mathbb{N}$. In particular, if we choose $i = 2$, then we have

$$uv^2xy^2z = 0^{n^2+k}. \tag{10.9}$$

However, because $1 \leq k \leq n$, it cannot be that $n^2 + k$ is a perfect square. This is because $n^2 + k$ is larger than n^2, but the next perfect square after n^2 is

$$(n+1)^2 = n^2 + 2n + 1, \tag{10.10}$$

which is strictly larger than $n^2 + k$ because $k \leq n$. The string uv^2xy^2z is therefore *not* contained in SQUARE, which contradicts the third condition stated by the pumping lemma, which guarantees us that $uv^ixy^iz \in$ SQUARE for all $i \in \mathbb{N}$.

Having obtained a contradiction, we conclude that SQUARE is not context free, as claimed. □

Remark 10.4. We will not discuss the proof, but it turns out that every context-free language over a single-symbol alphabet must be regular. By combining this fact with the fact that SQUARE is nonregular, we obtain a different proof that SQUARE is not context free.

Here is one more example of a proof that a particular language is not context free using the pumping lemma for context-free languages. For this one things get a bit messy because there are multiple cases to worry about as we try to get a contradiction, which turns out to be fairly common when using this method. Of course, one has to be sure to get a contradiction in *all* of the cases in order to have a valid proof by contradiction, so be sure to keep this in mind.

Proposition 10.5. *Let* $\Sigma = \{0, 1, \#\}$ *and define a language* B *over* Σ *as follows:*

$$B = \{r\#s \,:\, r, s \in \{0,1\}^*,\ r \text{ is a substring of } s\}. \tag{10.11}$$

The language B *is not context free.*

Proof. Assume toward contradiction that B is context free. By the pumping lemma for context-free languages, there exists a pumping length n for B. We will fix such a pumping length n for the remainder of the proof.

Let

$$w = 0^n1^n\#0^n1^n. \tag{10.12}$$

It is the case that $w \in B$ (because 0^n1^n is a substring of itself) and $|w| = 4n + 1 \geq n$. The pumping lemma therefore guarantees that there exist strings $u, v, x, y, z \in \Sigma^*$ so that $w = uvxyz$ and the three properties in the statement of that lemma hold: (i) $vy \neq \varepsilon$, (ii) $|vxy| \leq n$, and (iii) $uv^ixy^iz \in B$ for all $i \in \mathbb{N}$.

There is just one occurrence of the symbol # in w, so it must appear in one of the strings u, v, x, y, or z. We will consider each case separately:

Case 1: the # lies within u. In this case we have that all of the symbols in v and y appear to the right of the symbol # in w. It follows that

$$uv^0xy^0z = 0^n1^n\#0^{n-j}1^{n-k} \tag{10.13}$$

for some choice of integers j and k with $j + k \geq 1$, because by removing v and y from w we must have removed at least one symbol to the right of the symbol # (and none from the left of that symbol). The string (10.13) is not contained in B, even though the third property guarantees it is, and so we have a contradiction in this case.

Case 2: the # lies within v. This is an easy case: because the # symbol lies in v, the string $uv^0xy^0z = uxz$ does not contain the symbol # at all, so it cannot be in B. This is in contradiction with the third property, which guarantees that $uv^0xy^0z \in B$, and so we have a contradiction in this case.

Case 3: the # lies within x. In this case, we know that $vxy = 1^j\#0^k$ for some choice of integers j and k for which $j + k \geq 1$. The reason why vxy must take this form is that $|vxy| \leq n$, so this substring cannot both contain the symbol # and reach either the first block of 0s or the last block of 1s, and the reason why $j + k \geq 1$ is that $vy \neq \varepsilon$. If it happens that $j \geq 1$, then we may choose $i = 2$ to obtain a contradiction, as

$$uv^2xy^2z = 0^n1^{n+j}\#0^{n+k}1^n, \tag{10.14}$$

which is not in B because the string to the left of the # symbol has more 1s than the string to the right of the # symbol. If it happens that $k \geq 1$, then we may choose $i = 0$ to obtain a contradiction: we have

$$uv^0xy^0z = 0^n1^{n-j}\#0^{n-k}1^n \tag{10.15}$$

in this case, which is not contained in B because the string to the left of the # symbol has more 0s than the string to the right of the # symbol.

Case 4: the # lies within y. This case is identical to case 2—the string uv^0xy^0z cannot be in B because it does not contain the symbol #.

Case 5: the # lies within z. In this case we have that all of the symbols in v and y appear to the left of the symbol # in w. Because $vy \neq \varepsilon$, it follows that

$$uv^2xy^2z = r\#0^n1^n \tag{10.16}$$

for some string r that has length strictly larger than $2n$. The string (10.16) is not contained in B, even though the third property guarantees it is, and so we have a contradiction in this case.

Having obtained a contradiction in all of the cases, we conclude that there must really be a contradiction—so B is not context free, as claimed. □

10.3 Non-context-free languages and closure properties

In the previous lecture it was stated that the context-free languages are not closed under either intersection or complementation. That is, there exist context-free languages A and B such that neither $A \cap B$ nor $\overline{A}$ are context free. We can now verify these claims.

First, let us consider the case of intersection. Suppose we define languages A and B as follows:

$$\begin{aligned} A &= \bigl\{0^n 1^n 2^m \,:\, n, m \in \mathbb{N}\bigr\}, \\ B &= \bigl\{0^n 1^m 2^m \,:\, n, m \in \mathbb{N}\bigr\}. \end{aligned} \tag{10.17}$$

These are certainly context-free languages—a CFG generating A is given by

$$\begin{aligned} S &\to X\,Y \\ X &\to 0\,X\,1 \;\Big|\; \varepsilon \\ Y &\to 2\,Y \;\Big|\; \varepsilon \end{aligned} \tag{10.18}$$

and a CFG generating B is given by

$$\begin{aligned} S &\to X\,Y \\ X &\to 0\,X \;\Big|\; \varepsilon \\ Y &\to 1\,Y\,2 \;\Big|\; \varepsilon \end{aligned} \tag{10.19}$$

On the other hand, the intersection $A \cap B$ is not context free, as our first proposition from the previous section established.

Having proved that the context-free languages are not closed under intersection, it follows immediately that the context-free languages are not closed under complementation. This is because we already know that the context-free languages are closed under union, and if they were also closed under complementation we would conclude that they must also be closed under intersection by De Morgan's laws.

Finally, let us observe that one can sometimes use closure properties to prove that certain languages are not context free. For example, consider the language

$$D = \bigl\{w \in \{0,1,2\}^* \,:\, |w|_0 = |w|_1 = |w|_2\bigr\}. \tag{10.20}$$

It would be possible to prove that D is not context free using the pumping lemma in a similar way to the first proposition from the previous section. A simpler way to conclude this fact is as follows. We assume toward contradiction that D is context free. Because the intersection of a context-free language and a regular language

must always be context free, it follows that $D \cap \mathrm{L}(0^*1^*2^*)$ is context free (because $\mathrm{L}(0^*1^*2^*)$ is the language matched by a regular expression and is therefore regular). However,

$$D \cap \mathrm{L}(0^*1^*2^*) = \left\{0^m 1^m 2^m \,:\, m \in \mathbb{N}\right\}, \tag{10.21}$$

which we already know is not context free. Having obtained a contradiction, we conclude that D is not context free, as required.