CS331 Clarifications

Sept 6, 2001 **Supratik Chakraborty**

There was some confusion towards the end of Thursday's class regarding the statement of Pumping Lemma for CFLs (PL-CFL). Subsequently, this lead to some more confusion when we were trying to show that certain non-context-free languages satisfy PL-CFL. Hopefully, this note will clarify the doubts. In case of additional doubts, send email to supratik@cse.iitb.ac.in

The statement of PL-CFL is as follows:
Given a CFL $\mathbf{L}$***, there exists an integer*** $\mathbf{n > 0}$ ***such that for all words*** $\mathbf{w \in L}$ ***with*** $\mathbf{|w| \geq n}$***, there exists a decomposition*** $\mathbf{w = uvzxy}$ ***with*** $\mathbf{|vzx| \leq n}$ ***and*** $\mathbf{|vx| \geq 1}$***, such that*** $\mathbf{uv^izx^iy \in L}$ ***for all*** $\mathbf{i \geq 0}$***.***

Note that the Lemma doesn't require $|z| \geq 1$ in general. In other words, it allows $z = \epsilon$. There was some confusion regarding this, because the proof studied in class started from a Chomsky-Normal Form grammar without ϵ-productions. As a result, it seemed that z in PL-CFL cannot be ϵ.

However, if you go over the proof outline of PL-CFL covered in class, all arguments continue to apply even if we had ϵ-transitions and/or unit-productions in the grammar. Thus, given a grammar $G = (\Sigma, V, P, S)$, we may simply remove useless symbols to get $G' = (\Sigma', V', P', S)$ with $L(G) = L(G')$. Each production rule in P' is now of the form $A \rightarrow \alpha$ where $\alpha \in (V' \cup \Sigma')^*$. In particular α could be ϵ for one or more production rules in P'. Let M be the maximum length of the right-hand side of a production rule in P', i.e., $M = \max_{A \rightarrow \alpha \in P'} |\alpha|$.

Now, recall the argument we used in class to show that if a terminal string $w \in \Sigma^+$ has a parse tree of height k due to a CNF grammar, then $|w| \leq 2^{k-1}$. Essentially, this was an argument relating the height and number of leaves of a binary tree. Now let's ask what happens if the tree is not binary, but one in which each node can have a maximum of M children? It's not hard to see that for such a tree of height k, the maximum number of leaves is M^k. Therefore, if a string $w \in \Sigma^*$ has a parse tree of height k due to grammar G' referred to above, then $|w| \leq M^k$. We are now in a position to derive PL-CFL starting from G'. Note that G' may contain ϵ- and/or unit-productions.

To complete the parallel with the proof studied in class, we choose the PL constant n to be equal to M^{m+1}, where $|V'| = m$. Choose a string $w \in L(G')$ such that $|w| \geq M^{m+1}$. It is easy to see that *any parse tree* for w must have height $\geq (m+1)$ by the argument relating the number of leaves to the height of a tree. Consider the longest path Π from the root to a leaf in a parse tree of w (the leaf could be ϵ since ϵ-transitions have not been eliminated from P'). It follows that $|\Pi| \geq m+1$. Now, each vertex in Π except the final one is a non-terminal symbol. Thus, there are at least $m+1$ non-terminal symbols in this path, and by the pigeon-hole principle, two of them must be identical.

As done in class, let us now start moving up from the leaf towards the root along Π. We stop at the first instance of a non-terminal symbol repeating twice along this path. Let the corresponding non-terminal symbol be X. Therefore, we have at least two instances of X along Π, as shown in Fig. 1, where the second instance $X^{(2)}$ is a descendant of the first instance $X^{(1)}$. Once again, as done in class, let us split the string w into u, v, z, x and y, as shown in Fig. 1. Since $X^{(1)}$ is the first instance of a non-terminal that appears twice as we move up from the leaf along Π, it follows that the tree rooted at $X^{(1)}$ is of height $\leq m+1$. Therefore, by the height vs number of children argument, the length of vzx is at most M^{m+1}, i.e., the PL-CFL constant. ***Observe that z can be ϵ here***.

Is it possible to have both v and x as ϵ? Suppose the answer is "Yes". In that case, we could definitely come up with another parse tree for w in which path Π could be shortened: simply replace the subtree rooted at $X^{(1)}$ by that rooted at $X^{(2)}$. By doing so, two possibilities arise:

1. Path Π still continues to be of length $\geq m+1$. In this case, we simply repeat the above argument again for the new shortened path Π.

2. Path Π becomes shorter than $m+1$. However, since $|w| \geq n = M^{m+1}$, there must exist some other path Π' in the tree with length $\geq m+1$. This is because a parse tree with all paths of length m or less has height $\leq m$. And a string derived from such a parse tree cannot have length $\geq n = M^{m+1}$. Having established the existence of Π' with $|\Pi'| \geq m+1$, we can now repeat the entire argument above for Π'.

Since $|w| \geq M^{m+1}$, we are guaranteed to eventually find a path Π of length $\geq m+1$, with two instances of a non-terminal X such that the corresponding decomposition of w as $uvzxy$ (as in Fig. 1) has either v or x as non-ϵ. Once

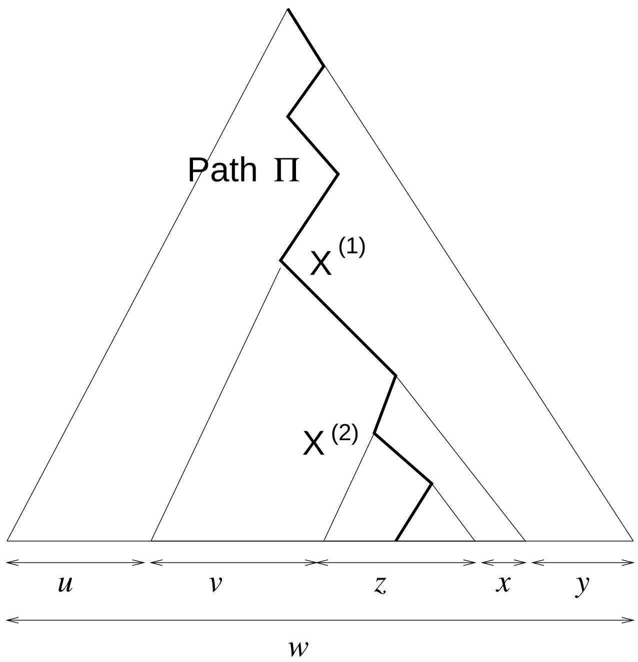

Figure 1:

we find such a path, we can apply the same argument used in class to show that uv^izx^iy must be in $L(G')$ for all $i \geq 0$.

Thus, PL-CFL does not, in general, require z to be non-ϵ. All it requires are: $w = uvzxy$ with $|vzx| \leq n$ and $|vx| \geq 1$.

How does this differ from what we studied in class? In class, we took a CNF grammar after removing all ϵ-productions, unit productions and useless symbols. As a result, z in the proof of PL-CFL was forced to be non-ϵ. The PL-CFL constant in the proof studied in class was $n = 2^m$, where m denotes the number of variables in the CNF grammar. In contrast, the PL-CFL constant in the proof outlined above is M^{m+1}, where M is the maximum length of the right hand side of a production rule, and m is the number of variables in the grammar. Thus, for the same context-free language L, we may come up with two CFGs G_1 and G_2: one of them (say G_1) without ϵ-productions and the other (say G_2) with ϵ-productions. If we try to show the validity of PL-CFL for L by constructing parse trees based on G_1, we may come up with some PL constant n_1. This could be completely different from the PL constant n_2 that we come up with when trying to show the validity of PL-CFL for L by constructing parse trees based on G_2. Nevertheless, both are valid approaches to show that PL-CFL holds for L. Also in the former case, z would not be allowed to be ϵ since all ϵ-productions have been removed, while in the latter case z could very well be ϵ. Thus for a CFL L, the question of whether z can or cannot be ϵ during the application of PL-CFL depends very much on the grammar for L that we have in mind. The statement of PL-CFL as such does not specify that z cannot be ϵ.

Recall that every CFL L such that $L - \{\epsilon\} \neq \phi$ can be generated from a CFG without ϵ productions (with the possible exception of $S \to \epsilon$, which is needed only when deriving the empty string). It follows that for every CFL L such that $L - \{\epsilon\} \neq \phi$, there exists a PL constant $n > 0$ such that in applying PL-CFL to strings of this language, z will never need to be ϵ. However, this doesn't rule out the application of PL-CFL with some other (possibly larger) n in which z can indeed be ϵ – this would correspond to parse trees derived from some other CFG for L, in which ϵ-productions are present.

Now let us try to show that the languagte $L = \{b^j c^k d^l \mid j, k, l \ \geq \ 1\} \cup \{a^i b^j c^j d^j \mid i, j \ \geq \ 1\}$ satisfies PL-CFL. We will see in a later class that L is not context-free.

For L, we choose the PL constant to be 4. We must now be able to show that for all strings $w \in L$ with $|w| \geq 4$, it is possible to decompose w as $uvzxy$ with $|vzx| \leq 4$, $|vx| \geq 1$ and (for the fun of it) $|z| \geq 1$, such that $uv^izx^iy \in L$ for all $i \geq 0$.

The following are the different possible choices for w:

1. $w = b^j c^k d^l$ with $j + k + l \geq 4$ and all of $j, k, l \ \geq \ 1$.
   There are three subcases here:

   (a) $k = l = 1$: Since $j + k + l \geq 4$, it follows that $j \geq 2$. Choose $u = b^{j-2}$, $v = b$, $z = bc$, $x = \epsilon$ and $y = d$. It is easy to see that $uzy \in L$ and $uv^izx^iy \in L$ for all $i \geq 1$.

   (b) $k \geq 2$: Choose $u = b^{j-1}$, $v = \epsilon$, $z = bc$, $x = c$, $y = c^{k-2}d^l$. Once again, uzy and uv^izx^iy are in L for all $i \geq 1$.

   (c) $l \geq 2$: Choose $u = b^j c^{k-1}$, $v = \epsilon$, $z = cd$, $x = d$, $y = d^{l-2}$. Once again, uzy and uv^izx^iy are in L for all $i \geq 1$.

2. $w = a^i b^j c^j d^j$ with $i, j \ \geq \ 1$.
   Choose $u = a^{i-1}$, $v = a$, $z = b$, $x = \epsilon$, $y = b^{j-1}c^j d^j$. Clearly, uzy and uv^izx^iy are in L for all $i \geq 1$.

Thus, L satisfies PL-CFL even if we require $|z| \geq 1$.