Class notes on the pumping lemma

First, state the theorem:

**Theorem 1 (The pumping lemma)** If $L$ is a context-free language, then there is a positive integer $k$, such that for any string $w \in L$ with length $|w| \geq k$, we have the following:

1. There are strings $u, v, x, y, z$ with $w = uvxyz$, so that also:
2. $|vxy| \leq k$, and (either $v \neq e$ or $y \neq e$);
3. $uxz \in L$ and $uvx^2yz \in L$.

We'll postpone the proof for a bit. First let’s restate the lemma in contrapositive form, which is the way you use it to show languages to be non-context free.

**Theorem 2 (Contrapositive form of pumping lemma)** Let $L$ be a language satisfying the following condition. For all positive integers $k$, there is a string $w \in L$ with $|w| \geq k$ such that:

1. For all decompositions $w = uvxyz$:
2. If $|vxy| \leq k$, and (either $v \neq e$ or $y \neq e$), then either $uxz \notin L$ or $uvx^2yz \notin L$.

Then $L$ is not a context-free language.

Using the contrapositive form we show $L = \{0^n1^n2^n \mid n \geq 0\}$ is not a context-free language. To do it, we let an enemy select an integer $k$. and in a carefully planned response, we choose $w = 0^k1^k2^k$. This $w$ meets the length requirement, so now we let the enemy decompose $w$ into five pieces $w = uvxyz$. Assuming only that $|vxy| \leq k$, and (either $v \neq e$ or $y \neq e$), we have to show how to pump up or down to get out of the language. This depends on covering all possible enemy decompositions, so we try to come up with a finite number of cases into which any of these decompositions must fall, and then in each case showing which way to pump out. Here are the cases:
1. One of the nonempty strings $v$ or $y$ crosses a boundary between 0’s and 1’s, or 1’s and 2’s. That is, either $v \notin 0^* \cup 1^* \cup 2^*$, or the same holds for $y$. Without loss of generality, suppose it’s $v$. Consider the string $uv^2xy^2z$. In the $vv$ part of this, we will have a substring of the first $v$ with two types of characters in it, and then in the second $v$ we again have the same two types of characters in a substring This means that the $vv$ substring will have too many alternations of character types to keep $uvvxyyz \in L$.

2. Both $v$ and $y$ are in $0^* \cup 1^* \cup 2^*$. Then when we pump down to the string $uxz$, we will remove characters from at most 2 out of the three blocks $0^k, 1^k, 2^k$ of $w$. Thus $uxz \notin L$.

The cases are pictured here. The first two lines illustrate case 1, and the second two lines illustrate case 2.

To summarize the strategy for proving languages non-CF:

- Let the enemy choose a $k$. You have no control over this.
- Cleverly choose a $w \in L$, Its length has to be bigger than $k$. It has to depend on $k$, and you cannot say just what $k$ is.
- Let the enemy choose any decomposition $w = uvxyz$. You cannot choose this. It is the enemy’s turn.
• You have to show that any of the enemy’s possible decompositions will be pumpable out of the language, either up or down. To do this you have to divide up the enemy’s possible decompositions into a finite number of exhaustive cases, and then argue that in each of your cases you can pump out somehow. Here it is helpful to draw box pictures as above.

• To aid in all of this, you may always assume that $v$ or $y$ is non-$e$, and that $|vxy| \leq k$.

Now we go back to the proof of the PL. First, recall a result from the homework: For any $n$ if $T$ is a finite tree branching at each internal node at most $p$, and $T$ has height $n$, then the number of leaves of $T$ is at most $p^n$.

To prove the PL, we assume there is a CFG $G = (N, \Sigma, R, S)$ generating $L$. Let $p$ be the length of (any) longest right hand side of a rule of $G$. Then any derivation tree of $G$ can branch apart at most $p$ at any interior node. Let $n$ be the number of nonterminal symbols of $G$: $n = |N|$. We now choose $k = p^n + 1$. Suppose $w \in L(G)$, and $|w| \geq k$. Let $T$ be a derivation tree for $w$. It certainly has more than $k$ leaves, so it must have height strictly greater than $n = |N|$. Thus on some path of $T$ there are more than $|N|$ interior nodes. These nodes are the pigeons, and $N$ is the set of pigeonholes, so there exist at least two distinct nodes on this path with the same nonterminal symbol labeling them, by the pigeonhole principle. The situation is pictured on the next page.
As can be seen, the substring $x$ of $w$ is generated by the lower $A$, and the substring $vxy$ of $w$ is generated by the upper $A$. If we cut out the subtree dominated by the upper $A$ and paste in the lower subtree, the resulting tree is still legitimate for the grammar and generates $uxz$. If, on the other hand, we make a copy of the upper $A$ subtree, cut out the lower $A$ subtree and paste in the copied tree, we get a new tree generating $uv^2xy^2z$. This guarantees the pumping property, but what about guaranteeing that $v$ or $y$ is nonempty, and $|vxy| \leq k$? The answer comes from choosing the original parse tree to be the smallest one generating $w$, in terms of the total number of nodes. With this tree, $v$ and $y$ cannot both be empty, because if they were, we could pump down the tree, getting a smaller tree generating $w = uvxyz = uxz$. And to get $|vxy| \leq k$, look at subtrees dominated by an $A$ which repeats underneath. There must be some subtree of this form such that on every path of the subtree, strictly underneath the top $A$ there are no repetitions. (If there were, just move down to where the repetition starts. You cannot go down forever.) The height of this special subtree is no more than $|N|$, and it dominates $vxy$, so by the HW, $|vxy| \leq p^{|N|} \leq k = p^{|N|} + 1$. The proof is complete.