Note: related lemma for regular languages is not as good as this.

Pumping Lemma for Context-Free Languages

If \( L \) is a CFL then there is an integer \( p \) such that:

\[ \forall w \in L \text{ with } |w| \geq p \]

we can write \( w = uvxyz \) such that:

1. \( |vy| \geq 1 \) (can strengthen to \( |v| \neq 0 \))
2. \( |vxy| \leq p \)
3. \( \forall i \geq 0, \ uv^ixy^iz \in L \)

Proof: Let \( G \) be a CFG with \( L = L(G) \) and assume that \( G \) is in Chomsky Normal Form.

Let \( V \) be the number of variables in \( G \)

Suppose that \( w \in L \) has a parse tree of height \( > |V| \).

\[ \Rightarrow \text{ root-leaf path length } > |V| \]

must contain repeated variable
by the **pigeonhole principle**, choose longest repeat.

Note: Since Chomsky Normal Form
- every leaf has a symbol in $Z$
- every parent of each leaf has 1 child
- every other internal node has 2 children
- a tree with height $h$ has $2^{h+1}$ children

Let $p = 2^{|W|}$: an maybe empty

If $|w| > p$ then $w$ requiresQue tree
  height $\geq |W| + 1$

$\implies$ Que tree has repeated variable
  on a path at height at most $|W| + 1$
  above leaves

Break up $w = uvxyz$ as in picture.
Since this is Chomsky normal form, we must have either \( v \) or \( y \) (or both) non-empty since no unit rules.

\[ \therefore \circ \text{ is true} \]

Since top \( A \) which generates \( vxy \)

has height \( 1 + |v| + 1 \)

\[ 1 + |vxy| = 2 |v| = \circ \]

\[ \therefore \circ \text{ is true} \]

Finally, for \( \circ \), we know the case \( uv^i x y^i z \)

we can repeat the rule section that generates \( u \) and \( v \) any number of times.

\[ u x z = uv^0 x y^0 z \]

(Note if \( v = \Xi \) we can rewrite \( w = u, v_1, x, y, z \)

where \( u_1 = uvx \)

\[ x_1 = y \]

\[ y_1 \]
How to use this to show not context-free:
Show: A special way of breaking up w into \( uv^ixy^i z \), for \( i \) size.

\[ L = \{ x \# x : x \in \{0,1\}^* \} \] is not a CFL.

Let \( p \) be the pumping length for \( L \).

Consider \( w = 0^p \# 0^{2p} \in L \).

What are options for \( vxy \) as part of \( w \)?

\[
\begin{align*}
&\text{If } v \text{ or } y \text{ contains } \# : \\
&\quad \text{For } i \neq 1, \ uv^i xy^i z \not\in L \text{ but before } \# \text{ wasn't match} \\
&\quad \text{part before } \# \text{ won't match} \\
&\quad \text{part after} \\
&\quad \text{same as above since parts won't match} \\
&\quad \text{If } v \text{ or } y \text{ contains } \# : \\
&\quad \text{For } i \neq 1, \ uv^i xy^i z \not\in L \text{ has too many/few } \#
\end{align*}
\]

- If \( v \) or \( y \) contains \#:
  - For \( i \neq 1 \), have too many/few \#.

- If \( v \) in 1st block and \( y \) in 2nd block:
  - Then since \( 1vx \# y \in L \).
  - \( v \) has only 1's.
  - \( y \) has only 0's.
and again # of 1's and 0's won't match when pumped it will

L is not a CFL

Another example \( L = \{ 0^n1^n2^n : n \geq 0 \} \) is not a CFL

Similar idea:

Consider \( 0^n1^n2^n \):

Pumping either makes up the order 0, 1, 2 or only the number of one or two of these symbols will change so don't get a string in \( L \).

We now move on to computational complexity.
**Time Complexity**

**Definition** The **running time** of a **NTM** \( M \) is the function \( T : \mathbb{N} \rightarrow \mathbb{N} \) given by:

\[
T(n) = \max \left\{ \# \text{ steps } M \text{ takes on any input } w \in \Sigma^* \text{ with } |w| = n \right\}
\]

This gives the definitions for both deterministic and nondeterministic TMs.

**Definition** For \( T : \mathbb{N} \rightarrow \mathbb{N} \) define

\[
\text{NTIME}(T(n)) = \exists A : \text{there is a multitape NTM that decides language } A \text{ with running time that is } O(T(n))
\]

Note: text uses single tape, but multitape is used by researchers more like our model.
Examples: \( A = \{x \# x : x \in \{0,1\}^* \} \)

1-tape TM can't do this better

1-tape TM from 1st TM we produced running time \( O(n^2) \) - need to shuttle back & forth

2-tape TM: copy part before \# to tape 2
time \( O(n) \) compare tapes 1 and 2

\( A \in \text{TIME}(n) \) linear time

Recall: Simulate of \( k \)-tape TM by 1-tape TM

\[
\begin{align*}
\text{k-tape TM} & \quad \Rightarrow \quad \text{1-tape TM} \\
T(n) & \quad \Rightarrow \quad O(T^2(n))
\end{align*}
\]

Best possible simulation even to go from 2-tape to 1-tape because of above example

Can actually prove:

\[
\begin{align*}
\text{k-tape TM} & \quad \Rightarrow \quad \text{2-tape TM} \\
T(n) & \quad \Rightarrow \quad O(T(n) \log T(n))
\end{align*}
\]

Handwritten notes:

- Lewis Stearns 1965
**Polynomial Time**

Edmonds, Cobham 1965:
- polynomial-time = good algorithm
- more than polynomial-time = bad algorithm

**Definition**

\[ P = \bigcup_{k} \text{TIME}(n^k) \]

all languages that can be decided in time \( O(n^k) \) for some constant \( k \).

**Definition**

\[ NP = \bigcup_{k} \text{NTIME}(n^k) \]

**Question:** Does \( P \neq NP \)?

Cook 1971
Karp 1972
Lera 1973
Recall: \[ \text{NTM} \quad \text{time} \quad T(n) \quad \Rightarrow \quad \text{1-tape TM} \quad \text{time} \quad 2^{O(T(n))} \]

If \( P=NP \) then one could get a vastly better simulation:

- \( (T(n))^k \) for some \( k \)
- for each language \( A \)
- \( k \) might depend on \( A \).