Equivalence of PDA’s and CFG’s

Th. L is context-free if and only some PDA recognizes L.

Proof: if L is context-free, we construct a PDA to simulate the grammar of L to see if input w can be generated. The PDA can nondeterministically guess rules of G to apply, and accept only if it generates the input string. The PDA’s stack is used to hold intermediate strings of terminals and variables generated so far. As terminals are generated in the beginning of the string, they are checked against the input as its being read.

Informal Description of Pfor CFL L:

1. Place the marker symbol $ and the start var. on the stack.
2. Keep repeating the following:
   a. If the top of the stack is a variable A, nondeterministically select one of the rules of A, and substitute the RHS of rule for A on the stack.
   b. If the top of the stack is a terminal a, then read the next input symbol. If it is an a, pop and repeat. If it’s not a match, reject (this branch of nondeterminism)
   c. If top of stack is $, enter accept state.
Example CFG to PDA

\[ S \rightarrow aTb \mid b \quad T \rightarrow Ta \mid \epsilon \quad (Ex. 2.14 p. 110) \]

Pumping Lemma for CFL's

If A is a context-free language, then there is a no. p (pumping length) where, if s is any string in A of length at least p, s may be divided into 5 pieces u,v,x,y,z, \( s = uvxyz \), such that all of the following hold:

1. for each \( i \geq 0 \), \( uv^i x y^i z \) is in A
2. \( |vy| > 0 \)
3. \( |vxy| \leq p \)

Condition 1 lets us "pump out" elements in A

Note that either v or y can be \( \epsilon \), but both cannot be (by 2)
(Without 2, the lemma is trivially true, with \( v = y = \epsilon \ ))

Condition 3 assures us we can make vxy small, if needed.
These are the only cases. End of proof.

Consider \(uvvxyyz\). It contains symbols out of order in the part that repeats and contains at least 2 symbols. Contradiction.

Case 2: Either \(v\) or \(y\) contains at least 2 symbols. (E.g., \(v\) contains \(a\)'s and \(b\)'s)

Consider \(uvvxyyz\). It contains symbols out of order in the part that repeats and contains at least 2 symbols. Contradiction.

These are the only cases. End of proof.

Ex. \(v = ab\). Then \(uababxyyz\) has substring \(abab\), which is not allowed.
### Pumping Lemma Example

L = {ww | w in {0,1}∗} is not context-free.

Suppose L were CF. Then let p be the pumping length given by the pumping lemma.

Let s = 0^p1^j0^j in L.

Note that |s| > p, so s = uvxyz as in the lemma.

**Case 1:** vxy is in the first half of the string
Then uv^2xyz will have the second half of the string start with a 1; but the first half starts with 0. So it cannot be of form ww. Contradiction.

**Case 2:** vxy is in the second half.
Follows similarly to case 1.

---

### Pumping Lemma Example, continued

**Case 3:** vxy is in the middle of the string.
Then uxz is of the form 0^p1^j0^j, where
i and j are not both p. This string is not of form ww. Contradiction.

*Qu:* What about uv^3xyz?

Could we have used 0^p1^0^p for s in the proof?
Proof of the Pumping Lemma

Proof Idea: Let $G$ be a CFG generating $A$, $V$ the vars. of $G$. Suppose $G$ has at most $b$ symbols on the RHS of any rule. Then parse trees for $G$ have at most $b$-way branching, and any tree of height $\leq h$ is the tree for a string of length at most $b^h$.

Ex. $b=2$, $h=2$:

Consider any $s$ in $A$ that’s "very long" ($\geq b^{\mid V \mid +2}$).

Since $s$ is in $A$, it has a parse tree, and the parse tree is "very tall" ($\geq \mid V \mid +2$) since $s$ is "very long".

Therefore, the parse tree must have a long path ($\geq \mid V \mid +2$)

---

Consider the variables along this long path. There must be a repeat of some variables on the path, since there are $\mid V \mid$ variables in $G$, and there are $\mid V \mid +1$ on the long path (by the pigeonhole principle).

Let $R$ be the last repeated variable from the leaf. (Take last repeat to satisfy length condition on $vxy$)
Surgery on Parse Trees

Use PDA’s for Parsing?

Nondeterministic PDA’s not practical

To simulate, have to keep track of

Set of possible states
All possible stack configurations

Too much memory!
Deterministic PDA's

Subclass of PDA's where at most 1 next configuration ε moves are allowed.

Ex. DPDA
\{ wcw^R | w \in \{a,b\}^* \}

Ex. Nondet. PDA
\{ ww^R | w \in \{a,b\}^* \}