Every language accepted by PDA is CF

For all pairs of states $p, q$, create $A_{pq}$

Simplifying Assumptions:
1. Only one accept state
2. PDA empties stack before accepting
3. Each transition: exactly one push XOR one pop

If you’re unsure why this is okay, think of this: what if I added this requirement an hour before homework is due? How could you modify a PDA to fit this?
Let’s create $A_{pq}$

- What is the first move? The last move?
  
  First is push, last pop

- Does first symbol pushed remain until end?

If yes?:

If no:

For any/all where $p \rightarrow r$ and $r \rightarrow q$, empty the stack
Proof of Correctness

Claim: if rule $A_{pq}$ generates $x$, then $x$ can bring PDA from $p$ with an empty stack to $q$ with an empty stack.

Proof: by induction on how many steps.

Basis: $A_{pp} \rightarrow \varepsilon$ is only

Inductive Hypothesis: true for up to $k$ steps. What if I have $k + 1$ steps?
What is the first step?

Suppose first step is $A_{pq} \rightarrow aA_{rs}b$

What if first step was $A_{pq} \rightarrow A_{pr}A_{rq}$?

$X = y z$
Proof: $A_{pq}$ generates all such strings

Induction on number of steps in computation.

**Basis:**

**Inductive Hypothesis:** suppose true for computation of length at most $k$. What if I have a $k + 1$ step computation?
The $k + 1$ step computation

Is the stack empty only at start and end?