

I) 0-PDA is an NFA

a) Since 1-PDA can accept languages that an NFA cannot 1-PDA is more powerful. Therefore if we can construct a 2-PDA that accepts a non-context free language then 2-PDA is more powerful.

Let $L = \{a^i b^i c^i | i \geq 0\}$ Let $P = \text{Pumping Length}$

wv^2xy^2z is not in L because if v or y contains only 1 symbol $a, b,$ & c will not be equal. Also if v and y contain at least 2 different symbols they will be mixed in a way not accepted by L . Therefore L is not context free.

However with 2 stacks the following language L can be accepted.

- Push all a s into stack 1
- Push all b s into stack 2 if ' a ' is read go to qreject
- Pop stack 1 & 2 for each ' c ' in the input stream
If both stacks are empty at end of input go to qaccept otherwise go to qreject. If any a s or b s are read in this state go to qreject.

Since L can be accepted by 2-PDA but not 1-PDA, 2-PDA is more powerful.

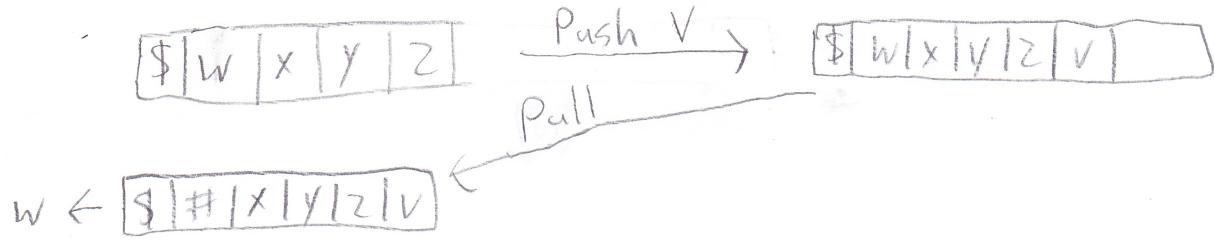
b) 3-PDA is not more powerful than a Turing machine. Since we can simulate a Turing machine with a 2-PDA by using 1 stack to represent the contents of the tape to the left of the current position & the other stack to represent right of the current position.

2) Since a Turing with stay put cannot read anything it has written to the tape, it behaves like a DFA. Since we can construct a DFA that accepts the same language as a Turing machine with stay put instead of Left. The Turing only accepts regular languages since DFAs only take regular languages

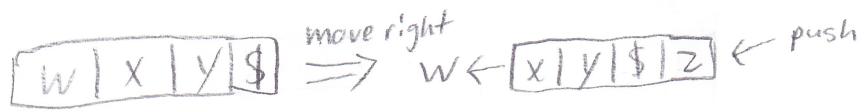
3) We can simulate a queue automata with a Turing Machine by using the tape as a simulated queue.

- To push the head goes to the first blank on the tape & writes to the blank

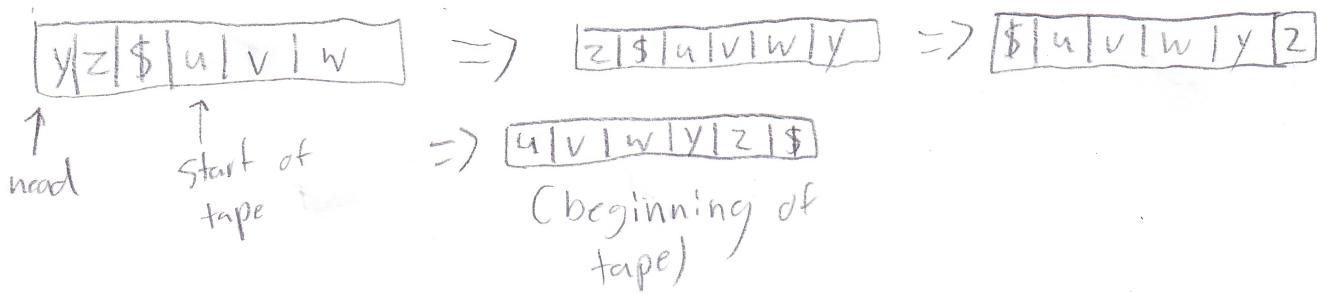
- To pull the head moves to the left of the tape & writes over the pulled symbol with a special character "#" for subsequent pulls the head returns to the left of the tape & pulls the leftmost non "#" symbol



To simulate a turing machine we read the input tape into the queue. Then we push \$ to represent the begining of the tape. To move right we simply pull a character from the head & push onto the tail.



To move left you will simply pop & push the character you popped until you pop \$ which indicates you are back at the beginning of the tape.
Then move head right to desired location



Since each machine can simulate the other the two must be equivalent