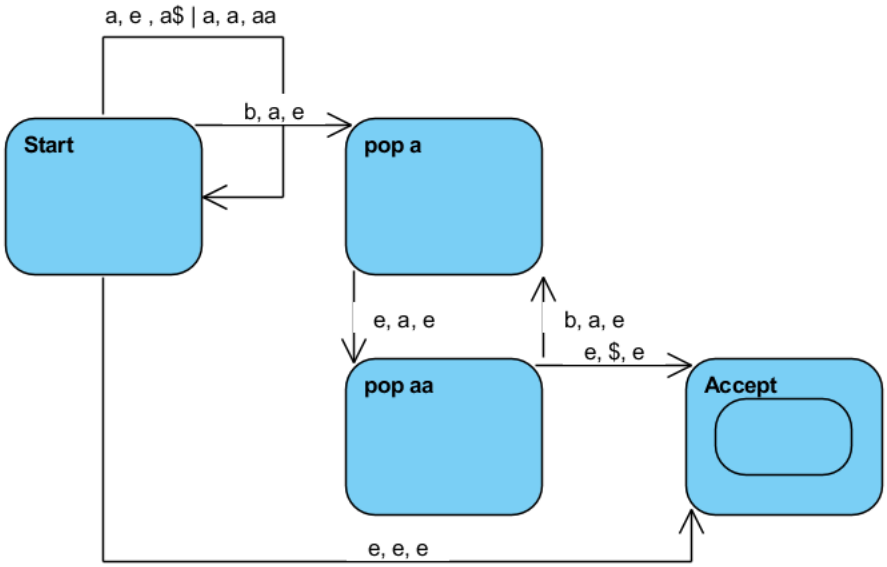
Michael Wang
HW 7

1)



The first symbol encountered must be an a, which will initialize the stack with a $ followed by an a, explicitly marking when the stack is empty. Any a encountered afterwards will push an additional a onto the stack. Upon encountering a b, two a's will be popped. Subsequent b's will also pop two a's. If and only if the stack has only $ remaining when the input string has been fully read will the DPDA accept. This ensures that exactly twice as many a's as b's are in the string. For the case that there are zero characters in the input string, the DPDA immediately accepts.

2)

Assume L2 is a CFL. Then there exists some length p that satisfies the pumping lemma. Consider the string S = $a^p b^{2p} c^p$ that is in L2 and has magnitude greater than p. S can be split into pieces uvxyz such that for all I $uv^i xy^i z$ is in L2 with either v or y being non-empty and vxy having magnitude less than p. The cases that must be considered are:

1. vxy is entirely within a^p, b^{2p}, or c^p
2. v and y contain a's and b's
3. v and y contain bs and c's

These cases are exhaustive since xyz cannot have length greater than p so v or y containing 'a' means neither can contain 'c' and vice versa as that would require vyx to cover all of b^{2p} which is longer than p. It is clear that in case 1 pumping any amount would cause the letter to have length something other than p disqualifying it from the language. In case 2, a string containing just a, just b, or both must be pumpable, but pumping any such string would either make the a string not have the same length as the c string or the b string not have exactly twice the length of the c string. The exact same logic applies for case 3 with a and c reversed. Therefore, no string vxy exists and the initial assumption is proven false. Thus L2 is not a CFL.