- Suppose x has both 0s and 1s, i.e., it spans the boundary of 1^p and 0^{2p}. Then, in uv^2wx^2y (of length $> 6p$), position $2p+1$ from the begining is a 0 while position $2p+1$ from the end is a 1. Thus, $uv^2wx^2y \notin A$.
- Suppose $x = \epsilon$ and v contains both 0s and 1s. Then uv^2wx^2y (of length $> 6p$) again has 0 in position $2p+1$ from the begining but 1 at position $2p+1$ from the end and so $uv^2wx^2y \notin A$.
- Suppose $x \neq \epsilon$ but contains only 0s, or vwx is completely contained in the last third of z. Then consider $z_0 = uv^{3p+4}wx^{3p+4}y$. Now, $|z_0| > 9p+4$. Moreover, the symbol at position $3p+1$ from the begining is a 1, while the symbol at position $3p+1$ from the end is a 0. Thus, $z_0 \notin A$.

■

Problem 4. [Category: OPTIONAL: Design] Design a PDA to recognize the language $C = \{x\#y \mid x, y \in \{0,1\}^* \text{ and } x \neq y\}$; thus, $C \subseteq \{0,1,\#\}^*$. You need not prove that your construction is correct, but you should clearly explain the intuition behind your construction. **[15 points]**

Solution: Observe that two strings $x, y \in \{0,1\}^*$ are not equal iff either (a) $|x| \neq |y|$, or (b) there is some i such that $x_i \neq y_i$, where x_i is the ith symbol of x and y_i is the ith symbol of y. On an input string $x\#y$, our PDA will nondeterministically guess whether $x \neq y$ because of condition (a) or because of condition (b). The PDA will check if its guess was correct. We will now describe how the PDA can check (a) and (b).

For condition (a), the PDA will count the number of symbols in x by pushing something onto the stack for each symbol read. After it reads # it will pop symbols, as it reads y to check how the length of y compares with that of x. If after reading all of y the stack is not empty, then $|x| > |y|$, and if after reading some of y the stack becomes empty then $|y| > |x|$. It will accept in either of these cases.

To check condition (b), the PDA will "guess" the position i and push symbols as it reads from x until it reaches the position it thinks does not match with y. It will read the symbol x_i and store it in its state. It will then ignore the rest of x, and once it reads #, it will start popping the stack as it reads symbols of y, so that it finds the same position i in y. When it reads y_i it will compare this with x_i that it remembered in its state, and if they are not equal, it will accept, else it will reject.

This algorithm can be implemented as a PDA that is shown in Figure 2. The stack alphabet is $\{\$, A\}$. The PDA can be understood as follows. From the initial state, we push a bottom of stack symbol \$ and move to state q. In state q, the PDA nondeterministically decides to either check for condition (a) (in which case it moves through states A, B, C, D) or check for condition (b) (in which case it moves through states $1, 2a, 3a, 4a, 2b, 3b, 4b$). We count the length of x by pushing symbols in state A, and move to state B of reading #. We compare the length of y with that x in B. If we go to state C it means that $|y| < |x|$ and if we go to state D then $|y| > |x|$. Checking for condition (b) proceeds as follows. The PDA nondeterministically decides if it has encountered the position i to check, and if so it moves to either state $2a$, if an 0 is read, or to state $2b$, if a 1 is read; it counts the position i by pushing symbols A onto the state as it reads symbols from x. The computation from $2a$ and $2b$ are similar and we describe only one of them. In state $2a$, we ignore the rest of the input until we read the # in which case we move to state $3a$ to check the ith symbol of y. In state $3a$ we count the symbols (by popping from the stack), and when the stack is empty (i.e., \$ is on the top) then the PDA knows that this is the ith symbol. It checks that this symbol is 1, and if so, it ignores the rest of the input and accepts.

■

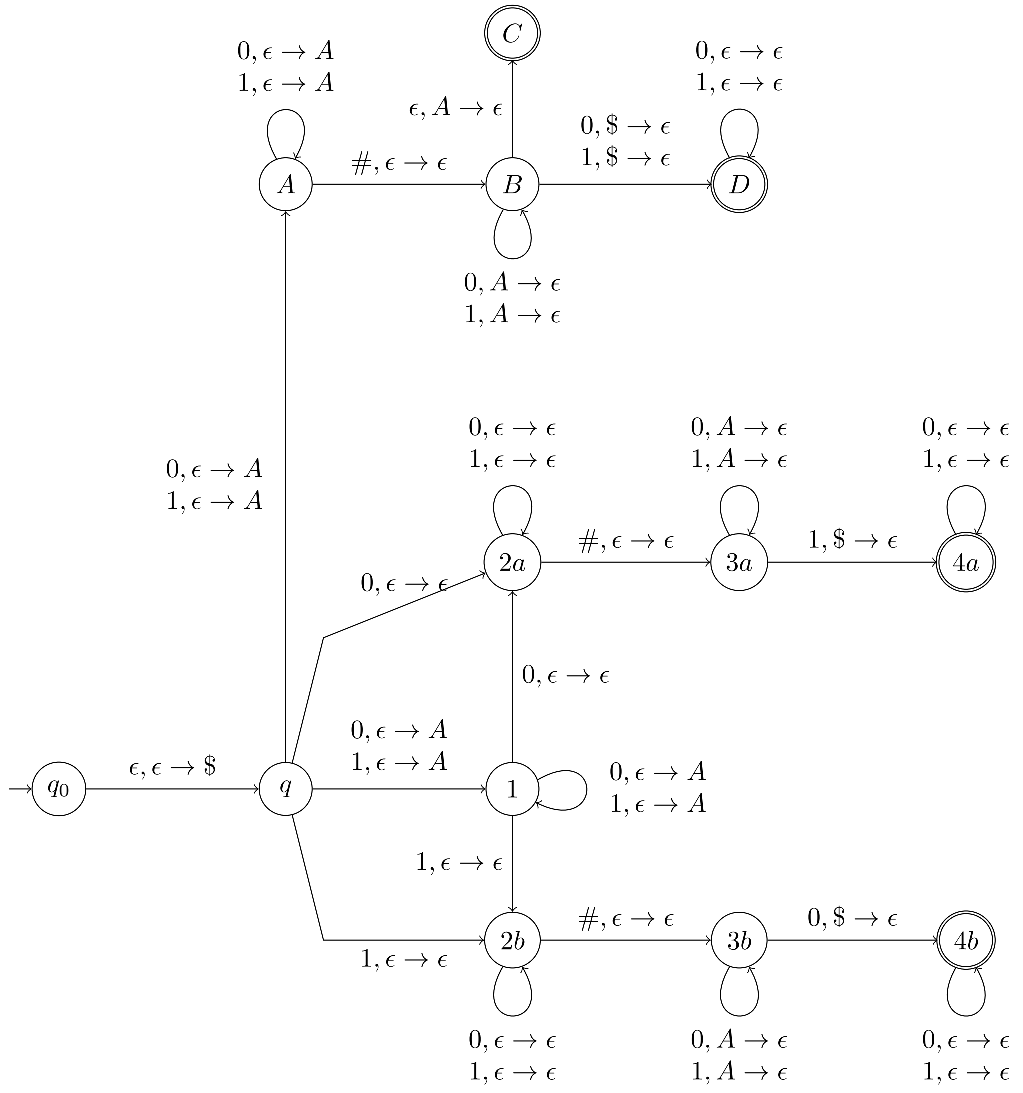
C
$0, \epsilon \to A$
$1, \epsilon \to A$
$\epsilon, A \to \epsilon$
$0, \epsilon \to \epsilon$
$1, \epsilon \to \epsilon$
$0, \$ \to \epsilon$
$1, \$ \to \epsilon$
A
$\#, \epsilon \to \epsilon$
B
D
$0, A \to \epsilon$
$1, A \to \epsilon$
$0, \epsilon \to \epsilon$
$1, \epsilon \to \epsilon$
$0, A \to \epsilon$
$1, A \to \epsilon$
$0, \epsilon \to \epsilon$
$1, \epsilon \to \epsilon$
$0, \epsilon \to A$
$1, \epsilon \to A$
$2a$
$\#, \epsilon \to \epsilon$
$3a$
$1, \$ \to \epsilon$
$4a$
$0, \epsilon \to \epsilon$
$0, \epsilon \to \epsilon$
$0, \epsilon \to A$
$1, \epsilon \to A$
$0, \epsilon \to A$
$1, \epsilon \to A$
q_0
$\epsilon, \epsilon \to \$$
q
1
$1, \epsilon \to \epsilon$
$2b$
$\#, \epsilon \to \epsilon$
$3b$
$0, \$ \to \epsilon$
$4b$
$1, \epsilon \to \epsilon$
$0, \epsilon \to \epsilon$
$1, \epsilon \to \epsilon$
$0, A \to \epsilon$
$1, A \to \epsilon$
$0, \epsilon \to \epsilon$
$1, \epsilon \to \epsilon$

Figure 2: PDA for Problem 2