High-Level Explanation:
Identify a string in the DFA, and a substring(Y) with size >= the number of states of that DFA.
This forces the substring to contain the content of the loop, and since loops can repeat any number of times, additional instances of this loop can be "pumped" into the substring.
Because the substring contains the content of the loop, the pumped content can be forced to be a substring of that substring.
If the L does not accept all of the newly constructed strings, then L cannot be represented as a DFA.

DFA Impossibility Proof:

1. Assume D is a DFA that recognizes language L. Let P be the number of states in D.
2. Choose a string S in L that can be split into XYZ, where |Y| > P.
3. By the pigeonhole principle, there must be at least one state in D that is visited multiple times in Y.
4. Due to the existence of a repeated state, the string Y can be split into $Y_1Y_2Y_3$, where reading Y_2 starts and ends at that repeated state, meaning Y_2 corresponds to a loop in the DFA.
5. Because Y_2 starts and ends in the same state, and the string $XY_1Y_2Y_3Z$ is in L, the string $XY_1Y_2^iY_3Z$ must also be in L, for any non-negative integer i by the pumping lemma.
6. If there is a $XY_1Y_2^iY_3Z$ not in L for any non-integer i, this results in a contradiction, meaning there exists no D that recognizes L.

If Y contains exactly one unique symbol a, then $Xa^{\mathrm{len(Y1)+len(Y2)+len(Y3)}}Z$ must be in L for some positive constant l and every non-negative i. Simplifying, $XYa^{l*i}Z$ must be in L for some positive constant l and non-negative constant c, for every non-negative i. If this is not true, then there is no DFA that recognizes L.