1. A FSM (states, transitions, etc.) can be encoded using a string of 0's and 1's, by picking all the binary strings in lexicographic order (0,1,00,01,10,11,000,001,...) and specify the FSA (bijective mapping between natural numbers and FSM) repeating the following steps:

1. start with i=1, m=1

2. generate the next binary string Si in lexicographic order

3.1. if Si is a valid encoding of an FSM then output Si as the m-th FSM and set m=m+1;

3.2. if Si is not a valid encoding of an FSM then ignore it

4. set i=i+1 and go to step 2

by this way each natural number (m=1,2,3,...) corresponds to a FSA, and each FSA has a corresponding m, since we scan all possible binary strings. So, the number of FSA is countable.

Finite state machines are built out of finite sets. One can represent a set of states as a string, an alphabet as a string, the transition function as a string (it's a finite table, because the domain is finite), the initial state and set of final states can also be encoded as a string. Therefore, there are fewer DFAs than strings; in other words, the number of possible DFAs is countably infinite.