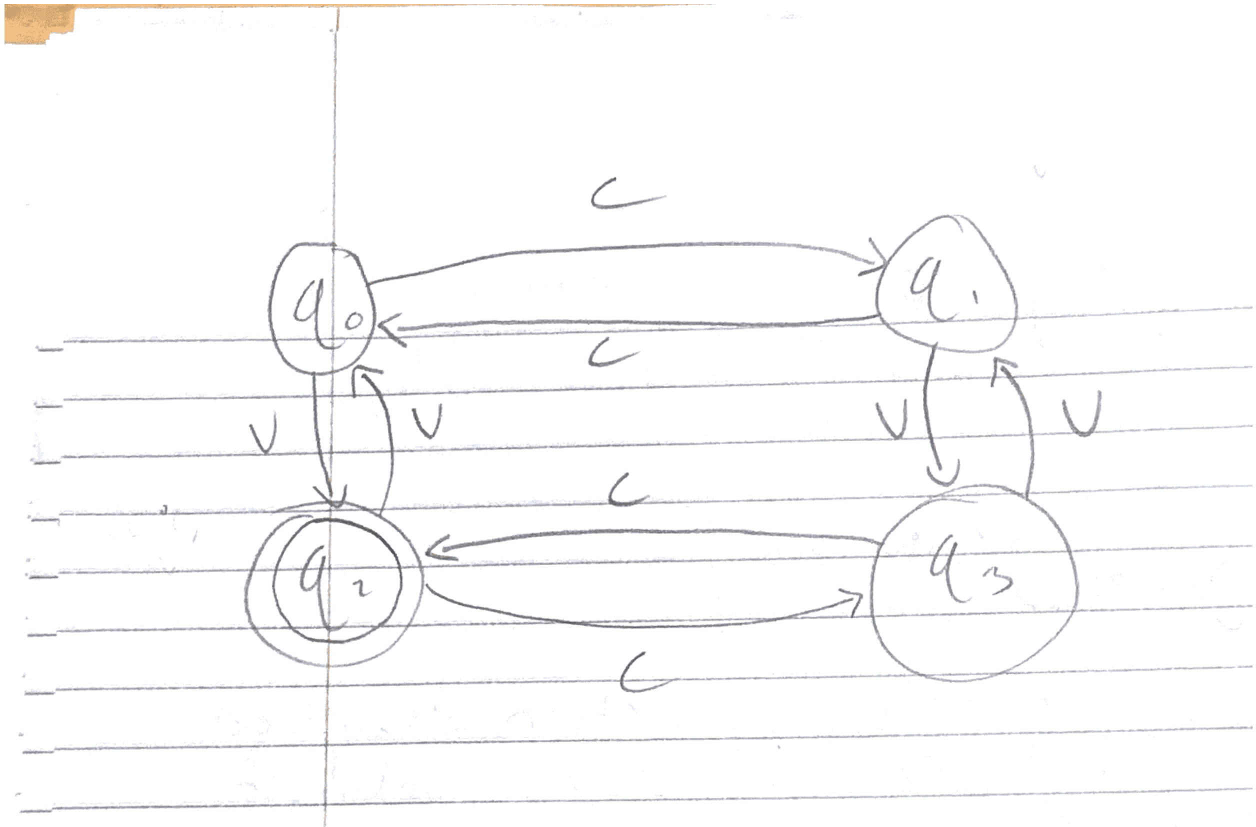
1A.



2.

Neither of the languages provided are regular languages. We use the Myhill-Nerode Theorem to prove this. Note that a language is regular if and only if there is an FSA that generates it. We prove that there is no FSA that generates either of these languages.

First we pick a stringset to consider for the theorem. We choose the stringset:

{a^n | n ≥ 0}

First, we see that all of these strings are in the language for any values of n, as a string of any number of a's can be reversed and be the exact same string, allowing for ww^R to be in the language, where w is the string of a's. We recognize that for any number n, a new equivalence class exists for the language, as the only suffix that exists for the string a^n is a^n. Thus, there is no other string that shares the same prefix with this string. Since there are an infinite number of n's, there are an infinite number of equivalence classes. Therefore, the Myhill-Nerode Theorem does not hold and there is not an FSA that generates the Language ww^R. Thus, the first language given is not regular.

The same logic can be used for the second Language given (the ww language). The same logic is used as for the previous language. We consider the stringset:

{a^n | n ≥ 0}

Since for any given n, there exists only one suffix that is suitable for that n, and this suffix is specific to that n and only that n, there will be infinitely many equivalence classes in the stringset, and therefore the language is not regular.

Optional part:
We can represent the equivalence classes of prefixes by the states in the FSA we used in question 1. The states represent a given number of C's and V's that have been input so far, thus the four equivalence classes are:

(EC = Equivalence class)

-EC for strings with an even number of C's and V's: this corresponds to the initial state of the FSA we made
-EC for strings with an odd number of C's and an even number of V's: this corresponds to state 1 in the FSA we made
-EC for strings with an even number of C's and an odd number of V's: this corresponds to the final state 2 in the FSA we made
-EC for strings with an odd number of both C's and V's: this corresponds to the state 3 in the FSA we made

Natural languages show this sort of restriction (along with many more). But as described in the lecture, we see restrictions like this with all kinds of sentences in English. For example, the types of phrases that are allowed to come after other kinds of phrases (See the John examples in the lecture 3). "John" is in a different equivalence class than "the fact that John" or "someone said John".