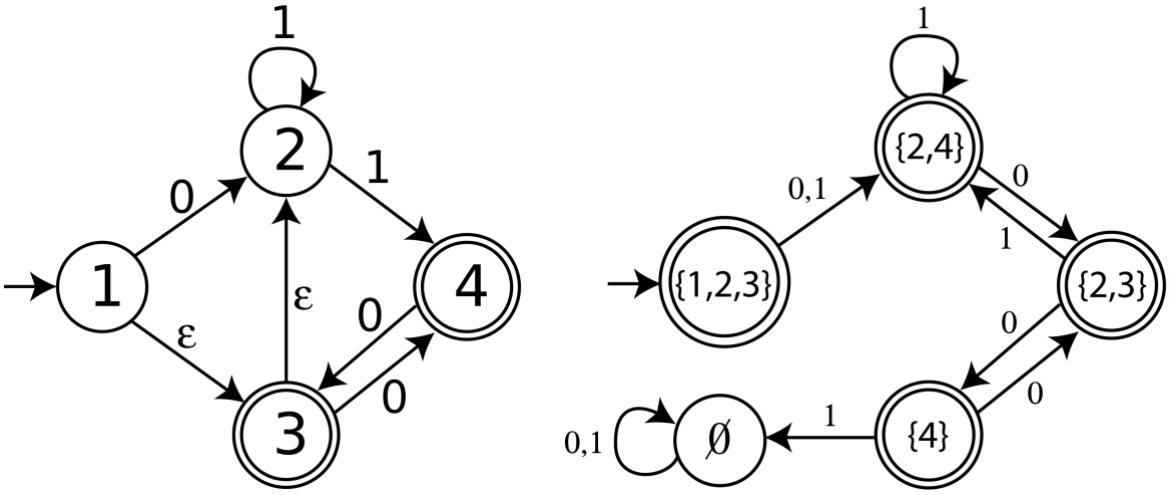
NFA to DFA Conversion

Although NFAs may appear to be more capable than DFAs they recognize the exact same class of languages, known as regular languages. If a language is regular, one can construct a DFA that recognizes it. As such, if there exists an NFA for a language, that NFA can be converted into an equivalent DFA.

The construction used to create this equivalent DFA is known as the Power Set Construction. This name comes from the fact that an equivalent DFA may have up to as many states as the power set of the states of the original NFA (or 2^n).

In the Power Set Construction, we start by defining our start state, whose name will be the set of all nodes reachable by an ε-transition from the start state, which includes the original start state. In the example below, the start state would be called {1,2,3}, because we start at state 1, and we can reach states 2 and 3 without reading any characters.



Example NFA to DFA Construction (source: Wikipedia article for Powerset Construction).
Left is the initial NFA, Right is the transformed DFA.

Once the start state is created, we will follow an iterative process of creating new states until no new states can be created. From every state, create a transition for every symbol in our input alphabet. The transitions for a state in the new DFA are equal to the transitions to all states reachable in the NFA from its set components. In state {1,2,3} in the above example, if we read a 0, we can reach state 2 (from state 1) and state 4 (from state 3), thus the equivalent transition in the DFA would be to {2,4} on 0.

Similarly, since state 2 can reach both 2 and 4 on 1, the transition from {1,2,3} on 1 would also be to state {2,4}. If there were no states reachable on reading some symbol, the transition would instead be to the empty set state, which in turn should have a transition to itself on every symbol in the alphabet and be considered a rejecting state.

Finally, every state in the new DFA which has in its name one of the accepting states in the NFA, will be an accepting state in the new DFA. Since {1,2,3} in the above example has 3 in it, it's an accepting state, as is {2,4}, because 4 was an accepting state in the NFA.