Subset Construction: NFA to DFA Conversion

- This algorithm is used to convert any NFA to a DFA.
- In general, if an NFA has Q states, then the equivalent DFA may have up to 2^Q states.
    - For example, if NFA has two states, then the equivalent DFA may have up to four states.
- The straight forward method for this conversion is to do the following:
    - Create the power set of the NFA states. This means, if the NFA has two states, q1 and q2, then the power set is: ({q1}, {q2}, {q1, q2}, ϕ).
    - If the NFA has three states, q1, q2, and q3, then the power set is: ({q1}, {q2}, {q3}, {q1, q2}, {q1, q3}, {q2, q3}, {q1, q2, q3}, ϕ).
    - Each state in the power set will become a state in the resultant DFA. For example, if the NFA has two states q1 and q2, then the four states of the DFA are: d1 → {q1}, d2 → {q2}, d12 → {q1, q2}, and dϕ → ϕ.
    - For each DFA state, we find the destination state of each transition for each symbol of the alphabet. For example, if Σ = {0, 1}, then we need to find: δ(d1, 0), δ(d1, 1), δ(d2, 0), δ(d2, 1), δ(d12, 0), δ(d12, 1), δ(dϕ, 0), and δ(dϕ, 1).
    - The previous step is based on using the given NFA. For each transition on each symbol, we find all possible states at which this transition may end at including those reachable through the ϵ-transition(s). For example, if we have q1 – a –> q2 – ϵ –> q3 – ϵ –> q4 – ϵ –> q5, then having an 'a' symbol at state q1 will result in the following set {q2, q3, q4, q5}.
    - For composite states in DFA, the final destination given one symbol is the union of all end states of each state of the composite state based on the NFA.
    - For example, for state d12, the final state for a transition on symbol '0' is the union of all states for each of q1 and q2 (using the NFA). The union of all states will be one of the states in the power set that is given a unique DFA state. pay attention to ϵ-transition(s) when finding the possible end states for any transition.
    - Once all destination states for all symbols in the alphabet are found, the DFA can be constructed using the found information.

- The start state of the DFA is the state that represents q0 and its ϵ-closure, where q0 is the start state of the NFA. This means that in order to find the start state in the DFA, we find the ϵ-closure (q0) and add q0. ϵ-closure is the set of all states reachable from q0 via ϵ-transition. In other words, the start state in DFA is the set of the power set that has the start state in NFA and all states that can be reached via ϵ-transition(s) via q0.

- The accept state in the DFA is any state that has the accept state in the NFA. For example, if q1 is an accept state in NFA, then d1, and d12 are two accept states in DFA.

Observations:

1. The method described above is good for NFA that has a few states (Q = 2 ~ 3). For NFA with states of 4 or more, one needs to examine an exponential number of DFA states. For example, if Q (NFA) = 10, then Q (DFA) = 2^{10} = 1024. This is not possible and there should be another way to handle such cases.
2. Most of NFAs (with Q states) can be converted to DFA with states less than 2^Q. in the example covered in the class, the NFA has two states and the equivalent DFA has only three states. A reduction was possible because one DFA state was unreachable and hence, it could be eliminated. This leads to a systematic way to find an equivalent DFA of an NFA that has many states, i.e., four or more.

The systematic way is to follow these steps:

- Find the DFA seed state (d0) which is the ϵ-closure of the start state in the NFA.
- For each symbol in the alphabet, we find all transitions from d0. In this step, we must take the ϵ-closure of the answer. This means that we need to consider all states reachable to the state that the symbol takes us including all states reachable via ϵ-transitions. Then, the new set of states is considered as a newly discovered DFA state, e.g. d1.
- Based on the transitions on each symbol for d0, we may discover a new state for the DFA. These newly discovered states are added to the list of the DFA states.
- We repeat the steps carried out for d0 for each of the newly discovered states. In each step, we keep observing if we are getting new DFA states or not. If yes, we add the new state and find ϵ-closure(δ(dn, x)) where dn is the new DFA state and x is one symbol.
- We repeat this until no more new states are discovered.
- The table generated from this algorithm is the transition table of the equivalent DFA which can be programmed easily.