DFA Minimization

Theory of Computing Phase 1

Supervised by

Dr. Zeinab Abdelhalim

Eng. Ahmed Jamal

Bahaa Shamoon Atia
202001572

Ehab Yahia Zakaria
202001294

Basma Mahfouz
202000920

Adham Waleed Fouad
202000857

Table of Contents:

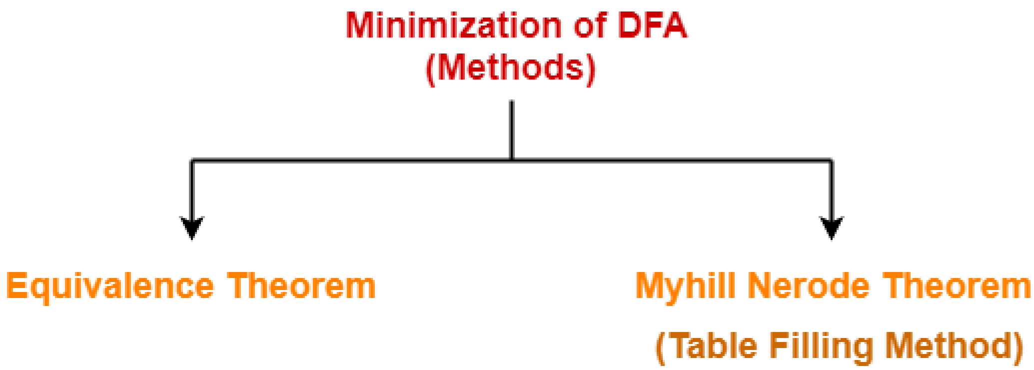
1. What is DFA minimization

DFA (Deterministic Finite Automaton) minimization is the process of reducing the number of states in a deterministic finite automaton while preserving its essential behavior. The goal is to create an equivalent DFA with the minimum possible number of states that still recognize the same language as the original DFA. Minimizing a DFA simplifies its structure, making it more manageable and efficient for various applications.

The primary motivation for DFA minimization is to optimize automata for better performance in terms of memory usage, computational speed, and ease of analysis. The process involves identifying and merging equivalent states, where two states are considered equivalent if they lead to the same outcomes for any given input string. By merging these equivalent states, redundant information is eliminated, resulting in a more compact representation of the language accepted by the automaton.

2. Methods used for DFA minimization

There are two popular methods for minimizing a DFA.



We will use the Equivalence Theorem to Minimize DFA

3. Minimization of DFA Using Equivalence Theorem

There are 6 Essential Steps for this method.

Step-01:

- Eliminate all the dead states and inaccessible states from the given DFA (if any)

Step-02:

- Draw a state transition table for the given DFA.

Step-03:

Now, start applying the equivalence theorem.

- Take a counter variable k and initialize it with value 0.
- Divide Q (set of states) into two sets such that one set contains all the non-final states, and the other set contains all the final states.

  - This partition is called P0.

Step-04:

- Increment k by 1.
- Find Pk by partitioning the different sets of Pk-1 .
- In each set of Pk-1 , consider all the possible pair of states within each set and if the two states are distinguishable, partition the set into different sets in P

Step-05:

- Repeat step-04 until no change in partition occurs.
- In other words, when you find Pk = Pk-1, stop.

Step-06:

- All those states which belong to the same set are equivalent.
- The equivalent states are merged to form a single state in the minimal DFA.

Number of states in Minimal DFA = Number of sets in P_k

4. Code

The code was written with Jupyter

4.1 input Format

We will insert the state transition table, The name of the starting state and the ending states

```
dfa_example = {
    'q0': {'a': 'q1', 'b': 'q2'},
    'q1': {'a': 'q1', 'b': 'q3'},
    'q2': {'a': 'q1', 'b': 'q2'},
    'q3': {'a': 'q1', 'b': 'q4'},
    'q4': {'a': 'q1', 'b': 'q2'}
}

final_states_example = ['q4']
start_state_example = 'q0'
```

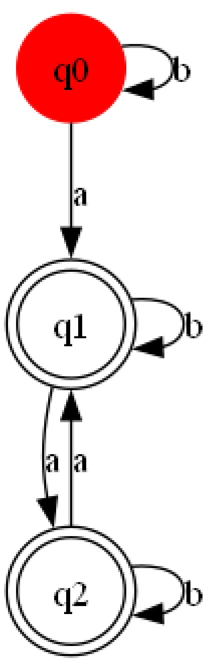
4.1 Output Format

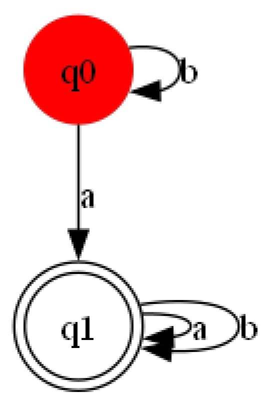
There are 2 output formats. (Graphical – text)

Graphical:

We draw the DFA before and after Minimization**.** We used “graphviz” library to draw the DFA.

Before **After**





Red circle is starting states.

Text:

We printed the state transition tables.

```
Original DFA:
State    a       b
→q0      q1      q2
q1       q1      q3
q2       q1      q2
q3       q1      q4
q4*      q1      q2

Minimized DFA:
State    a       b
q4*      q0      q2
→q0      q0      q4
q2       q0      q2
```

→ Starting State

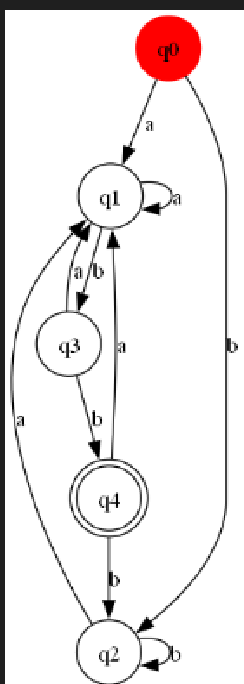
* Final State

Snip Shot of the code:

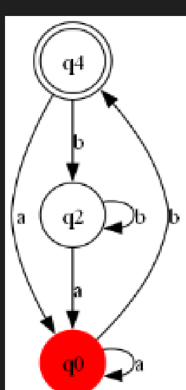
```
dfa_example = {
    'q0': {'0': 'q1', '1': 'q2'},
    'q1': {'0': 'q2', '1': 'q0'},
    'q2': {'0': 'q0', '1': 'q1'}
}

final_states_example = ['q0']
start_state_example ='q0'

minimized_dfa_example = minimize_dfa(dfa_example, final_states_example)

print("Original DFA:")
print_dfa(dfa_example, start_state_example,final_states_example)
print("\nMinimized DFA:")
print_dfa(minimized_dfa_example, start_state_example,final_states_example)
print("\n")
plot_dfa(dfa_example,final_states_example,start_state_example, "Original")
image = Image.open("Original_dfa.png")
display(image)

plot_dfa(minimized_dfa_example,final_states_example,start_state_example, "Minimized")
image = Image.open("Minimized_dfa.png")
display(image)
```

Snip shot output:

```
Original DFA:
State   a       b
→q0     q1      q2
q1      q1      q3
q2      q1      q2
q3      q1      q4
q4*     q1      q2

Minimized DFA:
State   a       b
q4*     q0      q2
→q0     q0      q4
q2      q0      q2


Original.
```




```
Minimized.
```



References

1. *https://www.youtube.com/watch?v=0XaGAkY09Wc*
2. *https://www.gatevidyalay.com/minimization-of-dfa-minimize-dfa-example/*