NFA-DFA 변환기 구현

2020112082 오현석

형식언어

송수환 교수님

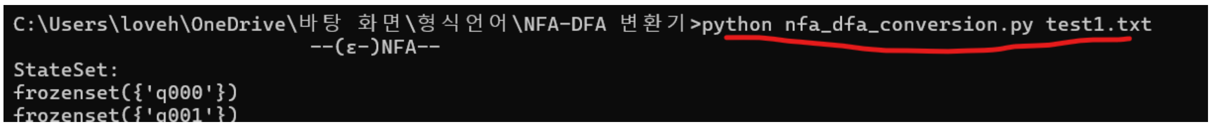
2024.05.19

1. 실행 방법 (윈도우 기준)

a) 실행 코드와 테스트 파일이 든 폴더를 압축 해제

b) 해당 폴더에서 cmd실행

c) python nfa_dfa_conversion.py test1.txt 명령어로 실행 (테스트 케이스는 1부터 10까지 존재)



```
C:\Users\loveh\OneDrive\바탕 화면\형식언어\NFA-DFA 변환기>python nfa_dfa_conversion.py test1.txt
                                        --(ε-)NFA--
StateSet:
frozenset({'q000'})
frozenset({'q001'})
```

2. 중요 코드

변환기에서 중요한 코드는 nfa에서 dfa로 변환하는 함수(a)와 그 중 epsilon-closure을 구하는 함수(b), 그리고 dfa를 최소화시켜주는 함수(c)이다.

(a) dfa_conversion 함수

파일에서 읽어온 NFA 정보를 이용하여 DFA로 변환하는 함수이다. 스택을 사용하고 있으며 stack을 startstate의 epsilon-closure로 초기화한다. while문을 통해 스택의 상태들을 가져와 해당 상태가 갈 수 있는 모든 상태를 고려하며 이것을 하나의 상태로 분류한다. epsilon-closure까지 합집합으로 합쳐 하나의 새로운 DFA 상태를 만든다. 마지막으로 final_state가 포함된 새로운 dfa상태들을 dfa_final_state로 저장하여 새로운 final_state를 만든다.

```
def dfa_conversion(self):
    dfa_states = []  # DFA 상태 저장 리스트
    dfa_delta = {}  # DFA 함수 저장 딕셔너리
    dfa_start_state = self.epsilon_closure(self.StartState)  # 시작 상태의 입실론-closure 를 구함
    # print(frozenset(dfa_start_state))
    stack = [frozenset(dfa_start_state)]  # 스택으로 DFA 상태를 구함, 시작 상태로 초기화
    # print(stack)
    while stack:
        current_dfa_state = stack.pop()  # 스택에서 DFA 상태를 pop
        if current_dfa_state not in dfa_states:
            dfa_states.append(current_dfa_state)  # dfa_states 리스트에 저장

            # 모든 가능한 DFA 상태를 탐색
            for symbol in self.TerminalSet:
                next_dfa_state = set()

                # 입실론-closure 까지 고려하여 상태를 저장
                for nfa_state in current_dfa_state:#현재 상태에서 갈 수 있는 모든 상태
                    transitions = self.DeltaFunctions.get((frozenset({nfa_state}), symbol), set())
                    for transition_state in transitions:#그 중 입실론-closure 까지 고려하여 합집합
```

```
                            next_dfa_state |= self.epsilon_closure(transition_state)

                    if next_dfa_state:
                        next_dfa_state = frozenset(next_dfa_state)
                        if next_dfa_state not in dfa_states:
                            stack.append(next_dfa_state)  #이렇게 만들어진 새로운 상태가 dfa_state 리스트에
없다면 스택에 저장하고 다시 loop
                        dfa_delta[(current_dfa_state, symbol)] = next_dfa_state  # DFA 상태 전이 함수
업데이트

        # dfa_state 에서 finalState 를 포함한 상태를 dfa_final_state 로 저장
        dfa_final_states = [state for state in dfa_states if any(s in self.FinalState for s in state)]
```

(b) epsilon_closure 함수

특정 상태의 epsilon-closure을 구하는 함수이다. 스택을 사용하며 while문으로 반복한다. 현재 상태를 스택에서 받아와 현재 상태가 epsilon을 보고 다음 상태가 존재한다면 스택에 넣고 반복하여 epsilon-closure을 구할 수 있다.

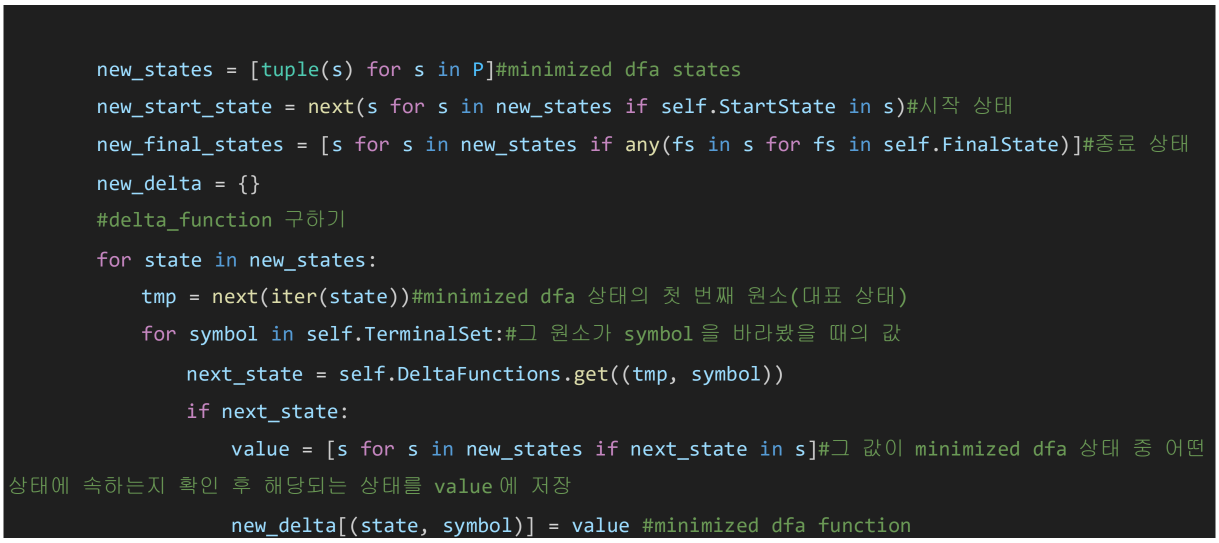
```
    def epsilon_closure(self, state):
            visited = set()  # 방문한 상태를 기록
            stack = [state]  # 스택으로 입실론-closure 를 찾아냄

            while stack:
                current_state = stack.pop()  #스택에서 상태를 받아옴
                if current_state not in visited:
                    visited.add(current_state)  #해당 상태를 visited 로

                    #입실론으로 갈 수 있는 모든 상태 탐색
                    epsilon_transitions = self.DeltaFunctions.get((frozenset({current_state}), 'ε'),
set())
                    for next_state in epsilon_transitions:
                        if next_state not in visited:
                            stack.append(next_state)  # 해당 상태가 visited 되지 않았다면 stack 에 추가 후
다시 loop
            return visited
```

(c) reduced_dfa 함수

DFA의 상태수를 최소화하는 함수이다. 처음에는 리스트 P에 상태를 finalstate와 그렇지 않은 상태들로 나누었다. P는 새로운 최소화된 DFA 상태를 저장할 리스트이다. 이때 다른 리스트 stack에 같은 집합 내용을 저장하였다. 이는 반복문을 통해 상태들을 합치고 나누어 최소화하기 위함이다.

이렇게 초기화된 stack에서 A를 꺼내와 각 심볼에 대한 상태들의 값이 A에 존재한다면 즉, A(새로운 상태 집합)에 도달할 수 있는 상태들을 X 집합에 저장한다. P안의 집합을 Y로 순회하며 X와 Y의 교집합과 차집합을 구한다. X는 A에 도달할 수 있는 상태들이고 Y는 P의 집합이므로 둘의 교집합은 같은 상태집합이 되고 둘의 차집합은 다른 상태집합이 된다. 교집합과 차집합이 둘 다 존재한다면 서로 다른 두 집합이 생기는 것으로 P와 stack에 저장한다. 이렇게 계속 반복한다면 결론적으로 P에 최소화된 상태 집합 리스트가 저장된다. 이를 이용해 시작상태와 종료상태 그리고 델타 함수를 구할 수 있다.

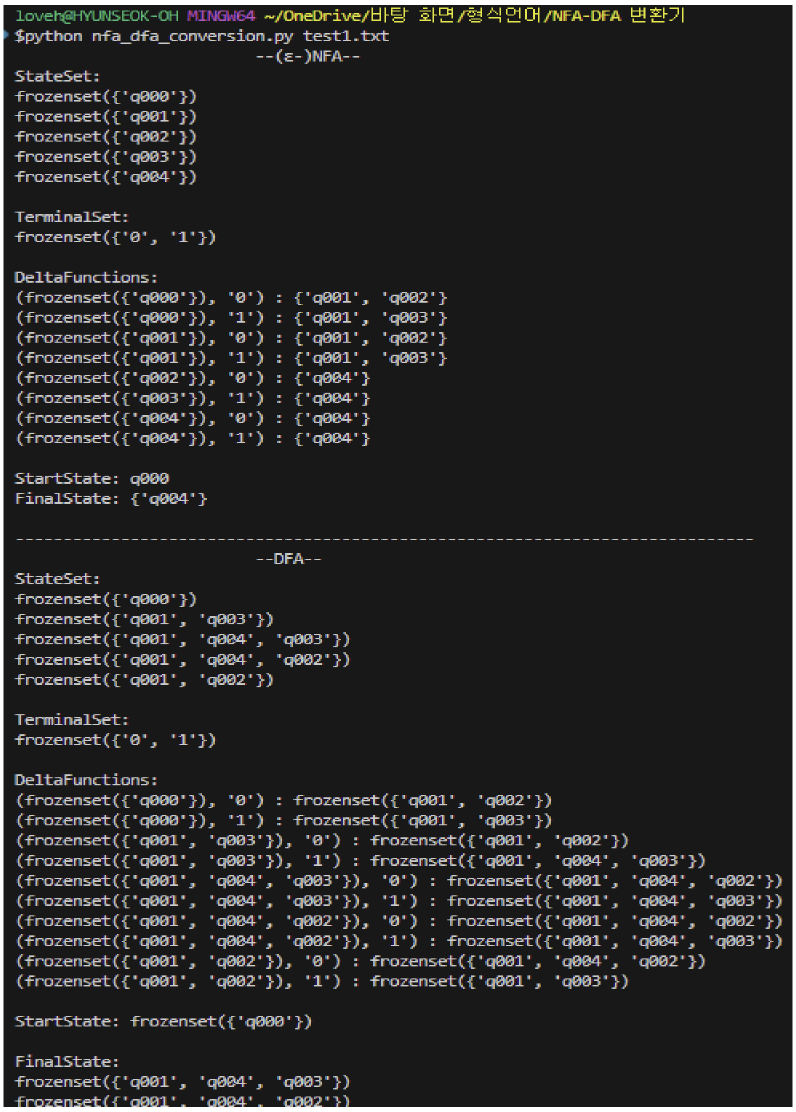
```
def reduced_dfa(self):

        #상태를 finalstate 와 그렇지 않은 상태로 나누기
        P = [set(self.FinalState), set(self.StateSet) - set(self.FinalState)]#실제 최소화된 dfa 상태가
저장될 리스트
        stack = [set(self.FinalState), set(self.StateSet) - set(self.FinalState)]#

        while stack:
            A = stack.pop()
            for symbol in self.TerminalSet:
                X = {state for state in self.StateSet if self.DeltaFunctions.get((state, symbol)) in
A}# A(새로운 상태 집합) 에 도달할 수 있는 state 들

                for Y in P.copy():#P 의 세트 Y 와 X 의 차집합, 교집합을 이용해 그룹 분리
                    intersection = X & Y
                    difference = Y - X

                    if intersection and difference:#차집합과 교집합이 둘 다 존재 -> 서로 다른 두 그룹
                        P.remove(Y)
                        P.append(intersection)
                        P.append(difference)

                        if Y in stack:#계속해서 loop 를 하기 위해 stack 에도 차집합, 교집합 추가
                            stack.remove(Y)
                            stack.append(intersection)
                            stack.append(difference)
                        else:
                            if len(intersection) <= len(difference):
                                stack.append(intersection)
                            else:
                                stack.append(difference)
```



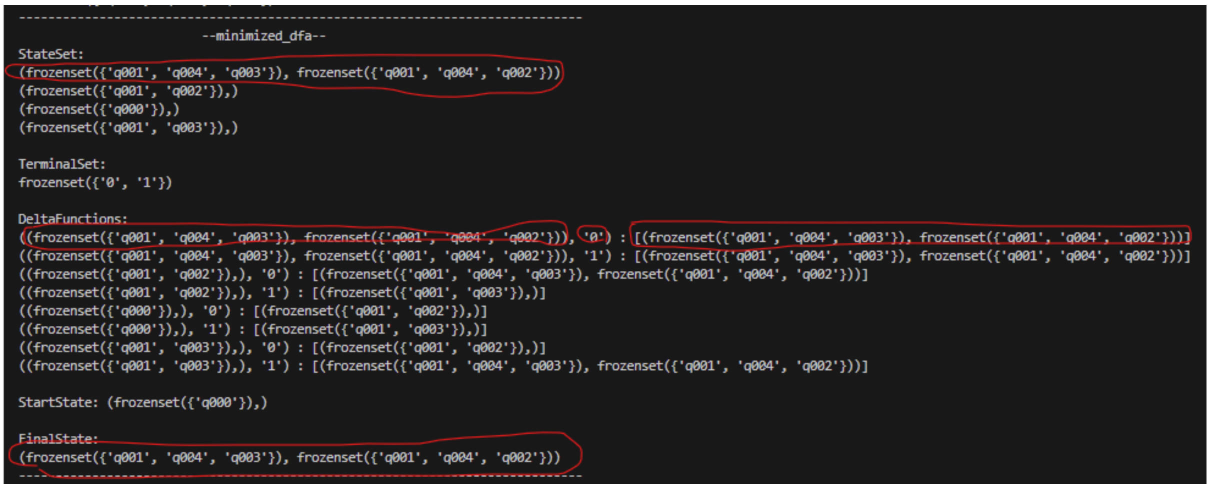
```
        new_states = [tuple(s) for s in P]#minimized dfa states
        new_start_state = next(s for s in new_states if self.StartState in s)#시작 상태
        new_final_states = [s for s in new_states if any(fs in s for fs in self.FinalState)]#종료 상태
        new_delta = {}
        #delta_function 구하기
        for state in new_states:
            tmp = next(iter(state))#minimized dfa 상태의 첫 번째 원소(대표 상태)
            for symbol in self.TerminalSet:#그 원소가 symbol 을 바라봤을 때의 값
                next_state = self.DeltaFunctions.get((tmp, symbol))
                if next_state:
                    value = [s for s in new_states if next_state in s]#그 값이 minimized dfa 상태 중 어떤
상태에 속하는지 확인 후 해당되는 상태를 value 에 저장
                    new_delta[(state, symbol)] = value #minimized dfa function
```

4. 실험 및 검증

1) test1.txt



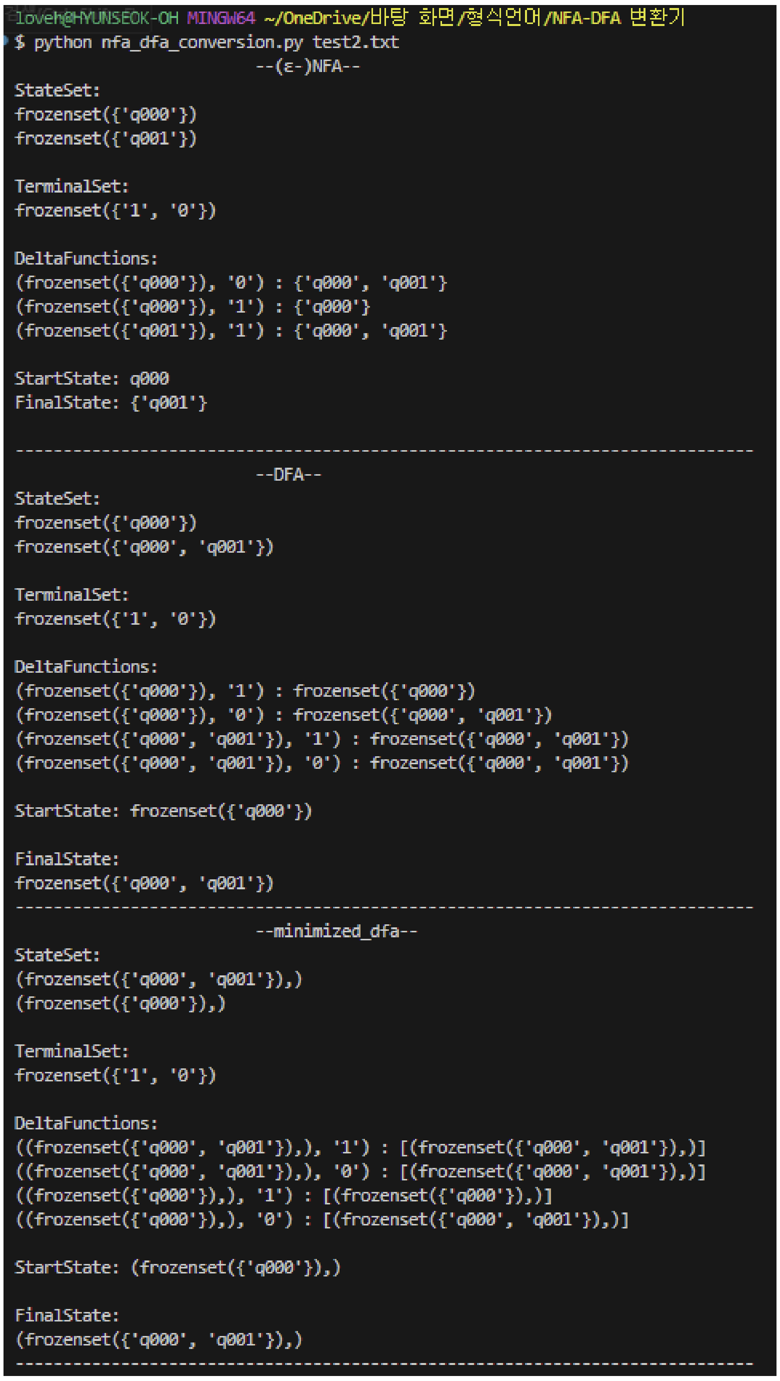
```
loveh@HYUNSEOK-OH MINGW64 ~/OneDrive/바탕 화면/형식언어/NFA-DFA 변환기
$python nfa_dfa_conversion.py test1.txt
                              --(ε-)NFA--
StateSet:
frozenset({'q000'})
frozenset({'q001'})
frozenset({'q002'})
frozenset({'q003'})
frozenset({'q004'})

TerminalSet:
frozenset({'0', '1'})

DeltaFunctions:
(frozenset({'q000'}), '0') : {'q001', 'q002'}
(frozenset({'q000'}), '1') : {'q001', 'q003'}
(frozenset({'q001'}), '0') : {'q001', 'q002'}
(frozenset({'q001'}), '1') : {'q001', 'q003'}
(frozenset({'q002'}), '0') : {'q004'}
(frozenset({'q003'}), '1') : {'q004'}
(frozenset({'q004'}), '0') : {'q004'}
(frozenset({'q004'}), '1') : {'q004'}

StartState: q000
FinalState: {'q004'}

----------------------------------------------------------------------------
                              --DFA--
StateSet:
frozenset({'q000'})
frozenset({'q001', 'q003'})
frozenset({'q001', 'q004', 'q003'})
frozenset({'q001', 'q004', 'q002'})
frozenset({'q001', 'q002'})

TerminalSet:
frozenset({'0', '1'})

DeltaFunctions:
(frozenset({'q000'}), '0') : frozenset({'q001', 'q002'})
(frozenset({'q000'}), '1') : frozenset({'q001', 'q003'})
(frozenset({'q001', 'q003'}), '0') : frozenset({'q001', 'q002'})
(frozenset({'q001', 'q003'}), '1') : frozenset({'q001', 'q004', 'q003'})
(frozenset({'q001', 'q004', 'q003'}), '0') : frozenset({'q001', 'q004', 'q002'})
(frozenset({'q001', 'q004', 'q003'}), '1') : frozenset({'q001', 'q004', 'q003'})
(frozenset({'q001', 'q004', 'q002'}), '0') : frozenset({'q001', 'q004', 'q002'})
(frozenset({'q001', 'q004', 'q002'}), '1') : frozenset({'q001', 'q004', 'q003'})
(frozenset({'q001', 'q002'}), '0') : frozenset({'q001', 'q004', 'q002'})
(frozenset({'q001', 'q002'}), '1') : frozenset({'q001', 'q003'})

StartState: frozenset({'q000'})

FinalState:
frozenset({'q001', 'q004', 'q003'})
frozenset({'q001', 'q004', 'q002'})
```



```
----------------------------------------------------------------------------
                         --minimized_dfa--
StateSet:
(frozenset({'q001', 'q004', 'q003'}), frozenset({'q001', 'q004', 'q002'}))
(frozenset({'q001', 'q002'}),)
(frozenset({'q000'}),)
(frozenset({'q001', 'q003'}),)

TerminalSet:
frozenset({'0', '1'})

DeltaFunctions:
((frozenset({'q001', 'q004', 'q003'}), frozenset({'q001', 'q004', 'q002'})), '0') : [(frozenset({'q001', 'q004', 'q003'}), frozenset({'q001', 'q004', 'q002'}))]
((frozenset({'q001', 'q004', 'q003'}), frozenset({'q001', 'q004', 'q002'})), '1') : [(frozenset({'q001', 'q004', 'q003'}), frozenset({'q001', 'q004', 'q002'}))]
((frozenset({'q001', 'q002'}),), '0') : [(frozenset({'q001', 'q004', 'q003'}), frozenset({'q001', 'q004', 'q002'}))]
((frozenset({'q001', 'q002'}),), '1') : [(frozenset({'q001', 'q003'}),)]
((frozenset({'q000'}),), '0') : [(frozenset({'q001', 'q002'}),)]
((frozenset({'q000'}),), '1') : [(frozenset({'q001', 'q003'}),)]
((frozenset({'q001', 'q003'}),), '0') : [(frozenset({'q001', 'q002'}),)]
((frozenset({'q001', 'q003'}),), '1') : [(frozenset({'q001', 'q004', 'q003'}), frozenset({'q001', 'q004', 'q002'}))]

StartState: (frozenset({'q000'}),)

FinalState:
(frozenset({'q001', 'q004', 'q003'}), frozenset({'q001', 'q004', 'q002'}))
----------------------------------------------------------------------------
```

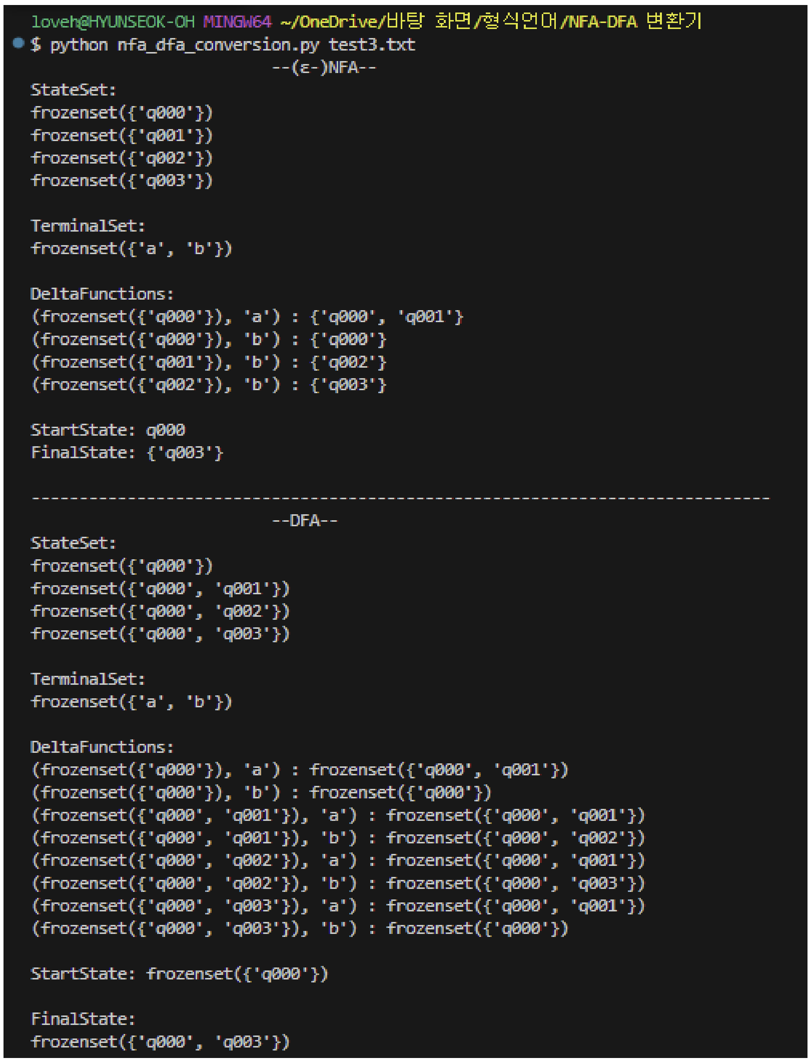
(빨간 선이 하나의 그룹이다. 한번에 인식이 힘들어 추가하였다.)

2)test2.txt

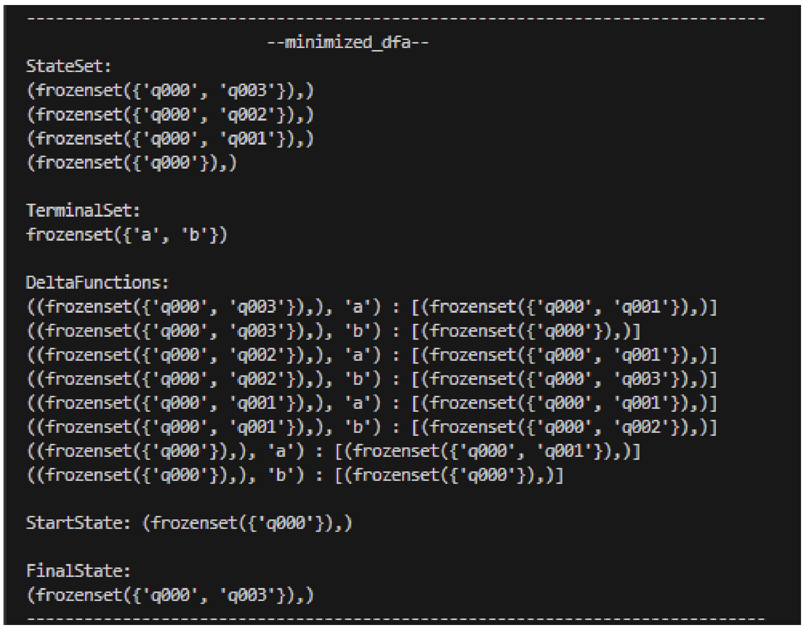


```
loveh@HYUNSEOK-OH MINGW64 ~/OneDrive/바탕 화면/형식언어/NFA-DFA 변환기
$ python nfa_dfa_conversion.py test2.txt
                          --(ε-)NFA--
StateSet:
frozenset({'q000'})
frozenset({'q001'})

TerminalSet:
frozenset({'1', '0'})

DeltaFunctions:
(frozenset({'q000'}), '0') : {'q000', 'q001'}
(frozenset({'q000'}), '1') : {'q000'}
(frozenset({'q001'}), '1') : {'q000', 'q001'}

StartState: q000
FinalState: {'q001'}

----------------------------------------------------------------------------
                          --DFA--
StateSet:
frozenset({'q000'})
frozenset({'q000', 'q001'})

TerminalSet:
frozenset({'1', '0'})

DeltaFunctions:
(frozenset({'q000'}), '1') : frozenset({'q000'})
(frozenset({'q000'}), '0') : frozenset({'q000', 'q001'})
(frozenset({'q000', 'q001'}), '1') : frozenset({'q000', 'q001'})
(frozenset({'q000', 'q001'}), '0') : frozenset({'q000', 'q001'})

StartState: frozenset({'q000'})

FinalState:
frozenset({'q000', 'q001'})
----------------------------------------------------------------------------
                          --minimized_dfa--
StateSet:
(frozenset({'q000', 'q001'}),)
(frozenset({'q000'}),)

TerminalSet:
frozenset({'1', '0'})

DeltaFunctions:
((frozenset({'q000', 'q001'}),), '1') : [(frozenset({'q000', 'q001'}),)]
((frozenset({'q000', 'q001'}),), '0') : [(frozenset({'q000', 'q001'}),)]
((frozenset({'q000'}),), '1') : [(frozenset({'q000'}),)]
((frozenset({'q000'}),), '0') : [(frozenset({'q000', 'q001'}),)]

StartState: (frozenset({'q000'}),)

FinalState:
(frozenset({'q000', 'q001'}),)
----------------------------------------------------------------------------
```

3) test3.txt

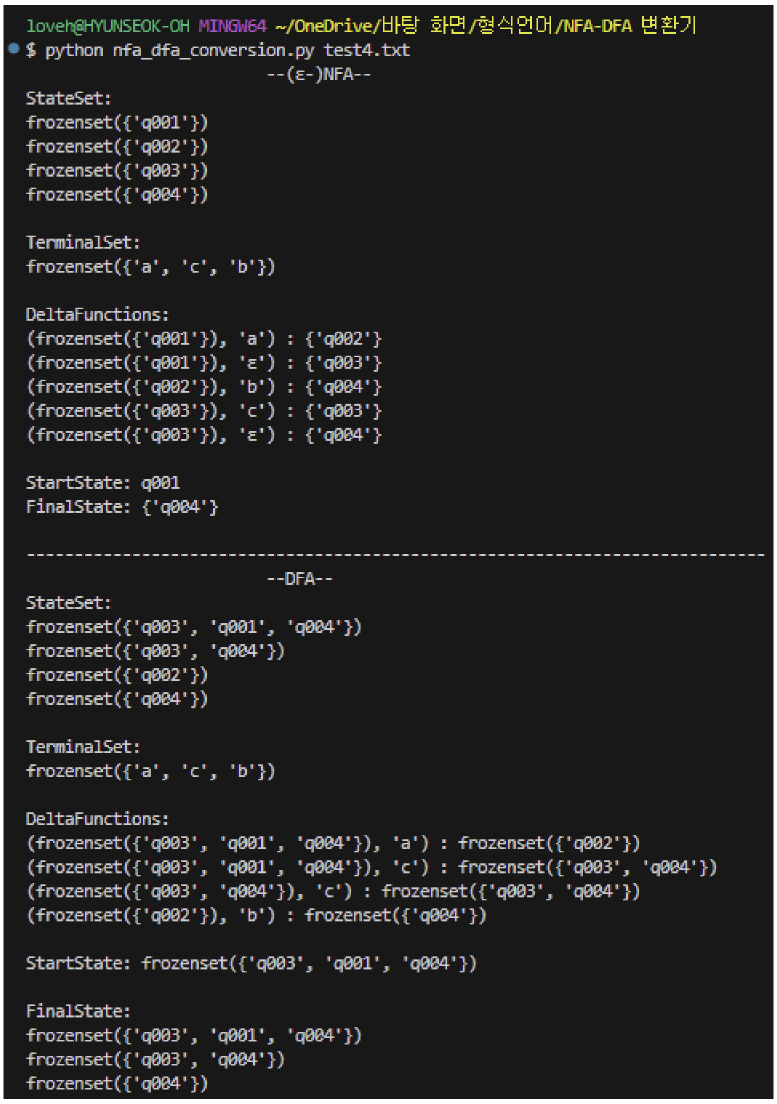


```
loveh@HYUNSEOK-OH MINGW64 ~/OneDrive/바탕 화면/형식언어/NFA-DFA 변환기
$ python nfa_dfa_conversion.py test3.txt
                          --(ε-)NFA--
StateSet:
frozenset({'q000'})
frozenset({'q001'})
frozenset({'q002'})
frozenset({'q003'})

TerminalSet:
frozenset({'a', 'b'})

DeltaFunctions:
(frozenset({'q000'}), 'a') : {'q000', 'q001'}
(frozenset({'q000'}), 'b') : {'q000'}
(frozenset({'q001'}), 'b') : {'q002'}
(frozenset({'q002'}), 'b') : {'q003'}

StartState: q000
FinalState: {'q003'}

-----------------------------------------------------------------------------
                          --DFA--
StateSet:
frozenset({'q000'})
frozenset({'q000', 'q001'})
frozenset({'q000', 'q002'})
frozenset({'q000', 'q003'})

TerminalSet:
frozenset({'a', 'b'})

DeltaFunctions:
(frozenset({'q000'}), 'a') : frozenset({'q000', 'q001'})
(frozenset({'q000'}), 'b') : frozenset({'q000'})
(frozenset({'q000', 'q001'}), 'a') : frozenset({'q000', 'q001'})
(frozenset({'q000', 'q001'}), 'b') : frozenset({'q000', 'q002'})
(frozenset({'q000', 'q002'}), 'a') : frozenset({'q000', 'q001'})
(frozenset({'q000', 'q002'}), 'b') : frozenset({'q000', 'q003'})
(frozenset({'q000', 'q003'}), 'a') : frozenset({'q000', 'q001'})
(frozenset({'q000', 'q003'}), 'b') : frozenset({'q000'})

StartState: frozenset({'q000'})

FinalState:
frozenset({'q000', 'q003'})
```

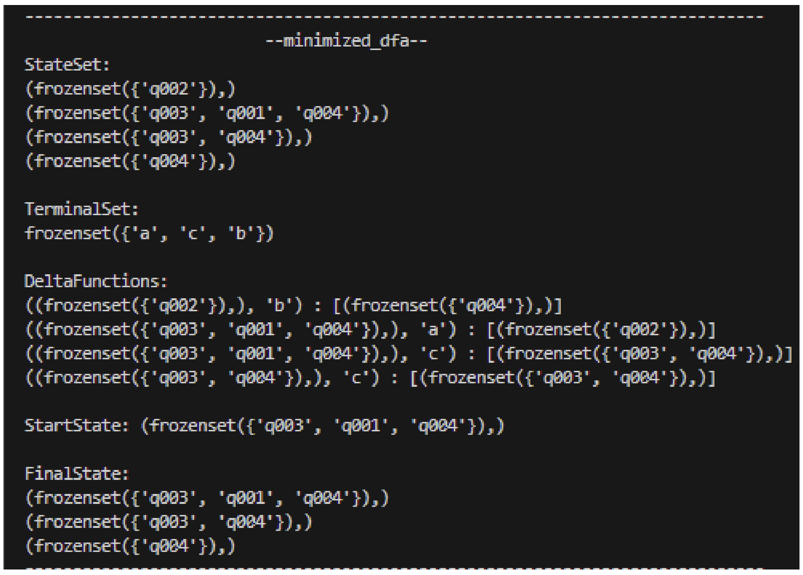


```
-----------------------------------------------------------------------------
                          --minimized_dfa--
StateSet:
(frozenset({'q000', 'q003'}),)
(frozenset({'q000', 'q002'}),)
(frozenset({'q000', 'q001'}),)
(frozenset({'q000'}),)

TerminalSet:
frozenset({'a', 'b'})

DeltaFunctions:
((frozenset({'q000', 'q003'}),), 'a') : [(frozenset({'q000', 'q001'}),)]
((frozenset({'q000', 'q003'}),), 'b') : [(frozenset({'q000'}),)]
((frozenset({'q000', 'q002'}),), 'a') : [(frozenset({'q000', 'q001'}),)]
((frozenset({'q000', 'q002'}),), 'b') : [(frozenset({'q000', 'q003'}),)]
((frozenset({'q000', 'q001'}),), 'a') : [(frozenset({'q000', 'q001'}),)]
((frozenset({'q000', 'q001'}),), 'b') : [(frozenset({'q000', 'q002'}),)]
((frozenset({'q000'}),), 'a') : [(frozenset({'q000', 'q001'}),)]
((frozenset({'q000'}),), 'b') : [(frozenset({'q000'}),)]

StartState: (frozenset({'q000'}),)

FinalState:
(frozenset({'q000', 'q003'}),)
-----------------------------------------------------------------------------
```

4) test4.txt

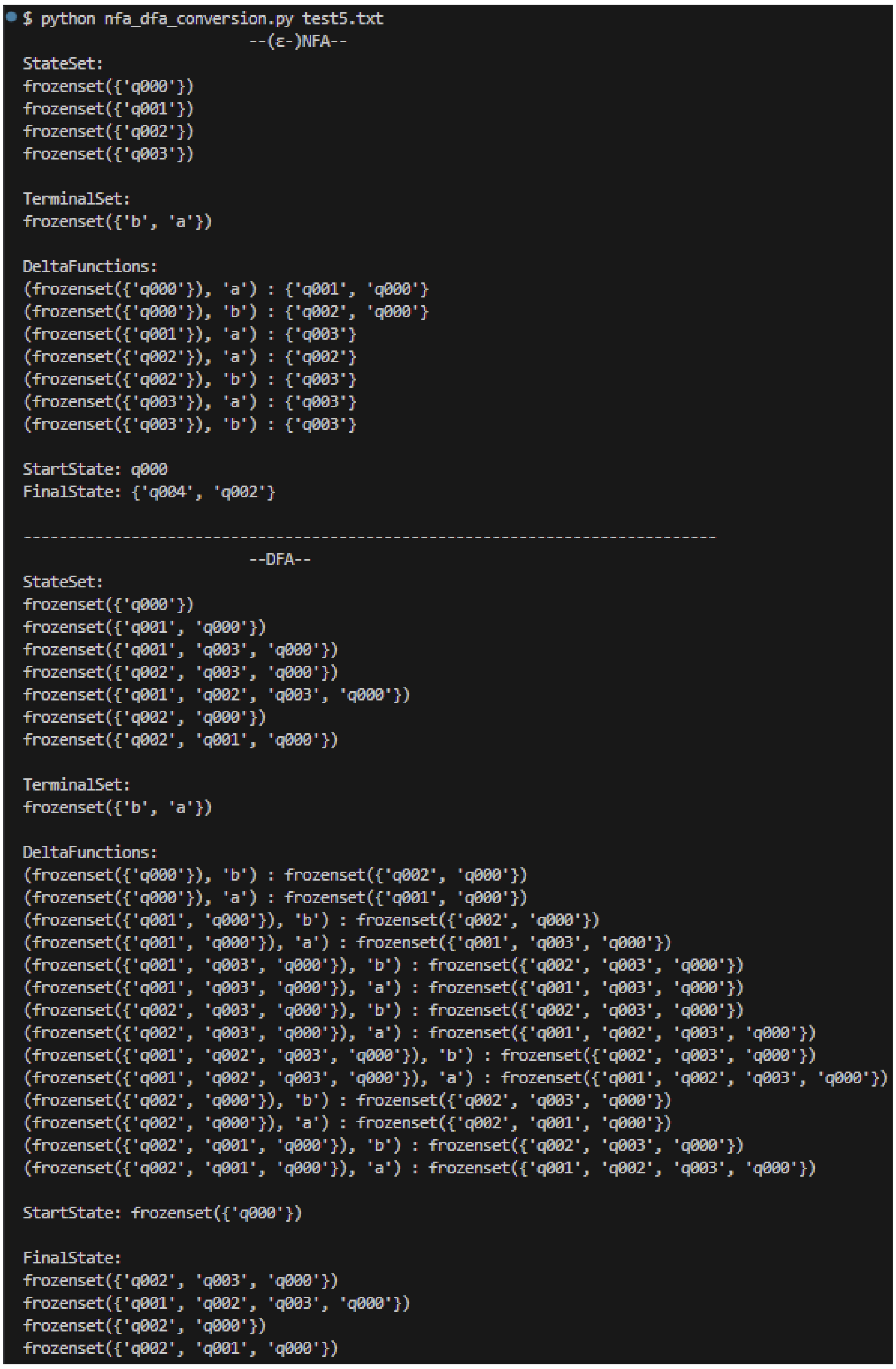


```
loveh@HYUNSEOK-OH MINGW64 ~/OneDrive/바탕 화면/형식언어/NFA-DFA 변환기
$ python nfa_dfa_conversion.py test4.txt
                             --(ε-)NFA--
StateSet:
frozenset({'q001'})
frozenset({'q002'})
frozenset({'q003'})
frozenset({'q004'})

TerminalSet:
frozenset({'a', 'c', 'b'})

DeltaFunctions:
(frozenset({'q001'}), 'a') : {'q002'}
(frozenset({'q001'}), 'ε') : {'q003'}
(frozenset({'q002'}), 'b') : {'q004'}
(frozenset({'q003'}), 'c') : {'q003'}
(frozenset({'q003'}), 'ε') : {'q004'}

StartState: q001
FinalState: {'q004'}

----------------------------------------------------------------------------
                             --DFA--
StateSet:
frozenset({'q003', 'q001', 'q004'})
frozenset({'q003', 'q004'})
frozenset({'q002'})
frozenset({'q004'})

TerminalSet:
frozenset({'a', 'c', 'b'})

DeltaFunctions:
(frozenset({'q003', 'q001', 'q004'}), 'a') : frozenset({'q002'})
(frozenset({'q003', 'q001', 'q004'}), 'c') : frozenset({'q003', 'q004'})
(frozenset({'q003', 'q004'}), 'c') : frozenset({'q003', 'q004'})
(frozenset({'q002'}), 'b') : frozenset({'q004'})

StartState: frozenset({'q003', 'q001', 'q004'})

FinalState:
frozenset({'q003', 'q001', 'q004'})
frozenset({'q003', 'q004'})
frozenset({'q004'})
```

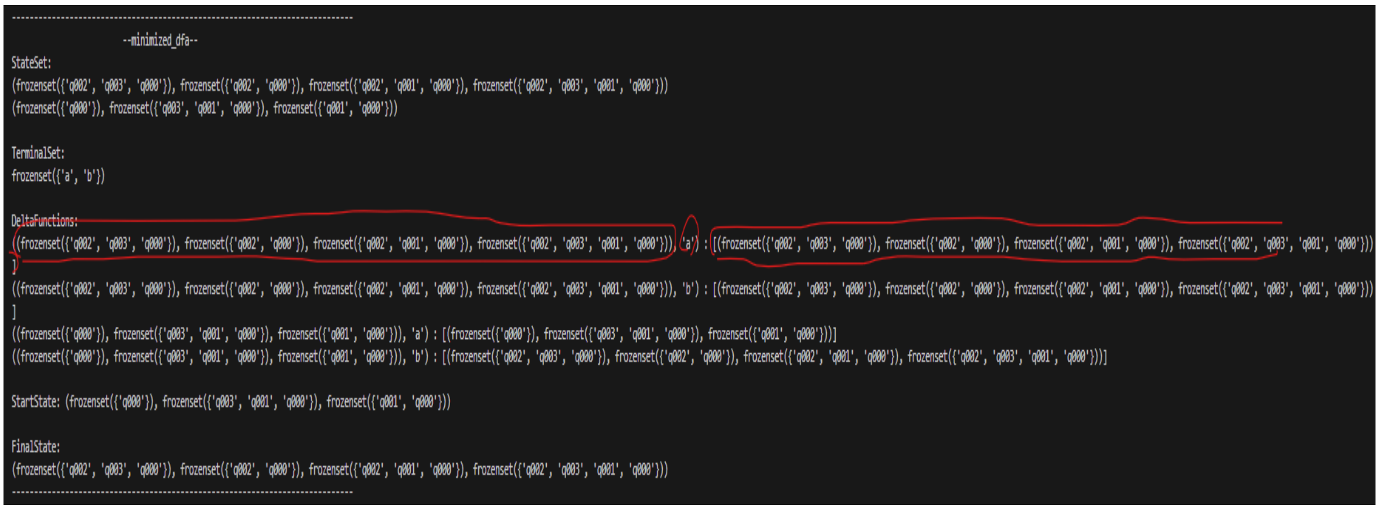


```
----------------------------------------------------------------------------
                             --minimized_dfa--
StateSet:
(frozenset({'q002'}),)
(frozenset({'q003', 'q001', 'q004'}),)
(frozenset({'q003', 'q004'}),)
(frozenset({'q004'}),)

TerminalSet:
frozenset({'a', 'c', 'b'})

DeltaFunctions:
((frozenset({'q002'}),), 'b') : [(frozenset({'q004'}),)]
((frozenset({'q003', 'q001', 'q004'}),), 'a') : [(frozenset({'q002'}),)]
((frozenset({'q003', 'q001', 'q004'}),), 'c') : [(frozenset({'q003', 'q004'}),)]
((frozenset({'q003', 'q004'}),), 'c') : [(frozenset({'q003', 'q004'}),)]

StartState: (frozenset({'q003', 'q001', 'q004'}),)

FinalState:
(frozenset({'q003', 'q001', 'q004'}),)
(frozenset({'q003', 'q004'}),)
(frozenset({'q004'}),)
----------------------------------------------------------------------------
```

5) test5.txt

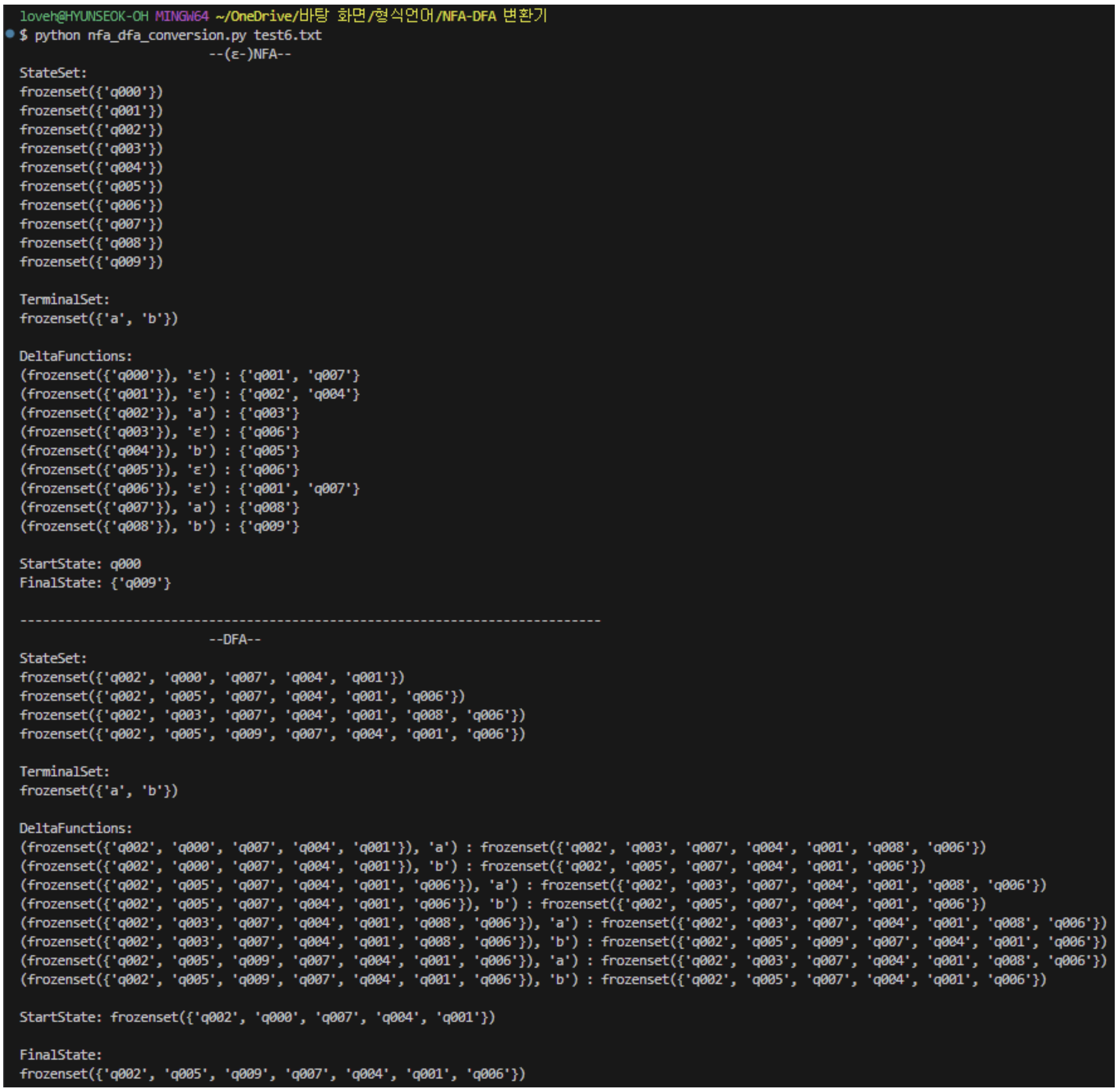


```
$ python nfa_dfa_conversion.py test5.txt
                         --(ε-)NFA--
StateSet:
frozenset({'q000'})
frozenset({'q001'})
frozenset({'q002'})
frozenset({'q003'})

TerminalSet:
frozenset({'b', 'a'})

DeltaFunctions:
(frozenset({'q000'}), 'a') : {'q001', 'q000'}
(frozenset({'q000'}), 'b') : {'q002', 'q000'}
(frozenset({'q001'}), 'a') : {'q003'}
(frozenset({'q002'}), 'a') : {'q002'}
(frozenset({'q002'}), 'b') : {'q003'}
(frozenset({'q003'}), 'a') : {'q003'}
(frozenset({'q003'}), 'b') : {'q003'}

StartState: q000
FinalState: {'q004', 'q002'}

-----------------------------------------------------------------------------
                         --DFA--
StateSet:
frozenset({'q000'})
frozenset({'q001', 'q000'})
frozenset({'q001', 'q003', 'q000'})
frozenset({'q002', 'q003', 'q000'})
frozenset({'q001', 'q002', 'q003', 'q000'})
frozenset({'q002', 'q000'})
frozenset({'q002', 'q001', 'q000'})

TerminalSet:
frozenset({'b', 'a'})

DeltaFunctions:
(frozenset({'q000'}), 'b') : frozenset({'q002', 'q000'})
(frozenset({'q000'}), 'a') : frozenset({'q001', 'q000'})
(frozenset({'q001', 'q000'}), 'b') : frozenset({'q002', 'q000'})
(frozenset({'q001', 'q000'}), 'a') : frozenset({'q001', 'q003', 'q000'})
(frozenset({'q001', 'q003', 'q000'}), 'b') : frozenset({'q002', 'q003', 'q000'})
(frozenset({'q001', 'q003', 'q000'}), 'a') : frozenset({'q001', 'q003', 'q000'})
(frozenset({'q002', 'q003', 'q000'}), 'b') : frozenset({'q002', 'q003', 'q000'})
(frozenset({'q002', 'q003', 'q000'}), 'a') : frozenset({'q001', 'q002', 'q003', 'q000'})
(frozenset({'q001', 'q002', 'q003', 'q000'}), 'b') : frozenset({'q002', 'q003', 'q000'})
(frozenset({'q001', 'q002', 'q003', 'q000'}), 'a') : frozenset({'q001', 'q002', 'q003', 'q000'})
(frozenset({'q002', 'q000'}), 'b') : frozenset({'q002', 'q003', 'q000'})
(frozenset({'q002', 'q000'}), 'a') : frozenset({'q002', 'q001', 'q000'})
(frozenset({'q002', 'q001', 'q000'}), 'b') : frozenset({'q002', 'q003', 'q000'})
(frozenset({'q002', 'q001', 'q000'}), 'a') : frozenset({'q001', 'q002', 'q003', 'q000'})

StartState: frozenset({'q000'})

FinalState:
frozenset({'q002', 'q003', 'q000'})
frozenset({'q001', 'q002', 'q003', 'q000'})
frozenset({'q002', 'q000'})
frozenset({'q002', 'q001', 'q000'})
```

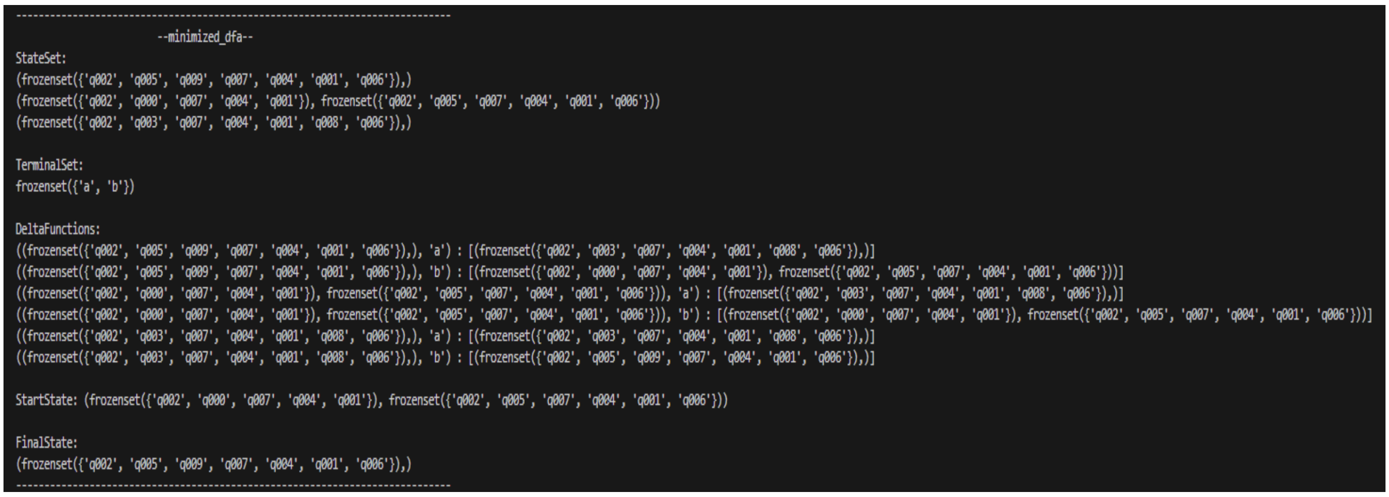


```
---------------------------------------------------------------------------
                        --minimized_dfa--
StateSet:
(frozenset({'q002', 'q003', 'q000'}), frozenset({'q002', 'q000'}), frozenset({'q002', 'q001', 'q000'}), frozenset({'q002', 'q003', 'q001', 'q000'}))
(frozenset({'q000'}), frozenset({'q003', 'q001', 'q000'}), frozenset({'q001', 'q000'}))

TerminalSet:
frozenset({'a', 'b'})

DeltaFunctions:
((frozenset({'q002', 'q003', 'q000'}), frozenset({'q002', 'q000'}), frozenset({'q002', 'q001', 'q000'}), frozenset({'q002', 'q003', 'q001', 'q000'})), 'a') : [(frozenset({'q002', 'q003', 'q000'}), frozenset({'q002', 'q000'}), frozenset({'q002', 'q001', 'q000'}), frozenset({'q002', 'q003', 'q001', 'q000'}))
]
((frozenset({'q002', 'q003', 'q000'}), frozenset({'q002', 'q000'}), frozenset({'q002', 'q001', 'q000'}), frozenset({'q002', 'q003', 'q001', 'q000'})), 'b') : [(frozenset({'q002', 'q003', 'q000'}), frozenset({'q002', 'q000'}), frozenset({'q002', 'q001', 'q000'}), frozenset({'q002', 'q003', 'q001', 'q000'}))
]
((frozenset({'q000'}), frozenset({'q003', 'q001', 'q000'}), frozenset({'q001', 'q000'})), 'a') : [(frozenset({'q000'}), frozenset({'q003', 'q001', 'q000'}), frozenset({'q001', 'q000'}))]
((frozenset({'q000'}), frozenset({'q003', 'q001', 'q000'}), frozenset({'q001', 'q000'})), 'b') : [(frozenset({'q002', 'q003', 'q000'}), frozenset({'q002', 'q000'}), frozenset({'q002', 'q001', 'q000'}), frozenset({'q002', 'q003', 'q001', 'q000'}))]

StartState: (frozenset({'q000'}), frozenset({'q003', 'q001', 'q000'}), frozenset({'q001', 'q000'}))

FinalState:
(frozenset({'q002', 'q003', 'q000'}), frozenset({'q002', 'q000'}), frozenset({'q002', 'q001', 'q000'}), frozenset({'q002', 'q003', 'q001', 'q000'}))
---------------------------------------------------------------------------
```

6) test6.txt



```
loveh@HYUNSEOK-OH MINGW64 ~/OneDrive/바탕 화면/형식언어/NFA-DFA 변환기
$ python nfa_dfa_conversion.py test6.txt
                         --(ε-)NFA--
StateSet:
frozenset({'q000'})
frozenset({'q001'})
frozenset({'q002'})
frozenset({'q003'})
frozenset({'q004'})
frozenset({'q005'})
frozenset({'q006'})
frozenset({'q007'})
frozenset({'q008'})
frozenset({'q009'})

TerminalSet:
frozenset({'a', 'b'})

DeltaFunctions:
(frozenset({'q000'}), 'ε') : {'q001', 'q007'}
(frozenset({'q001'}), 'ε') : {'q002', 'q004'}
(frozenset({'q002'}), 'a') : {'q003'}
(frozenset({'q003'}), 'ε') : {'q006'}
(frozenset({'q004'}), 'b') : {'q005'}
(frozenset({'q005'}), 'ε') : {'q006'}
(frozenset({'q006'}), 'ε') : {'q001', 'q007'}
(frozenset({'q007'}), 'a') : {'q008'}
(frozenset({'q008'}), 'b') : {'q009'}

StartState: q000
FinalState: {'q009'}

-----------------------------------------------------------------------------
                         --DFA--
StateSet:
frozenset({'q002', 'q000', 'q007', 'q004', 'q001'})
frozenset({'q002', 'q005', 'q007', 'q004', 'q001', 'q006'})
frozenset({'q002', 'q003', 'q007', 'q004', 'q001', 'q008', 'q006'})
frozenset({'q002', 'q005', 'q009', 'q007', 'q004', 'q001', 'q006'})

TerminalSet:
frozenset({'a', 'b'})

DeltaFunctions:
(frozenset({'q002', 'q000', 'q007', 'q004', 'q001'}), 'a') : frozenset({'q002', 'q003', 'q007', 'q004', 'q001', 'q008', 'q006'})
(frozenset({'q002', 'q000', 'q007', 'q004', 'q001'}), 'b') : frozenset({'q002', 'q005', 'q007', 'q004', 'q001', 'q006'})
(frozenset({'q002', 'q005', 'q007', 'q004', 'q001', 'q006'}), 'a') : frozenset({'q002', 'q003', 'q007', 'q004', 'q001', 'q008', 'q006'})
(frozenset({'q002', 'q005', 'q007', 'q004', 'q001', 'q006'}), 'b') : frozenset({'q002', 'q005', 'q007', 'q004', 'q001', 'q006'})
(frozenset({'q002', 'q003', 'q007', 'q004', 'q001', 'q008', 'q006'}), 'a') : frozenset({'q002', 'q003', 'q007', 'q004', 'q001', 'q008', 'q006'})
(frozenset({'q002', 'q003', 'q007', 'q004', 'q001', 'q008', 'q006'}), 'b') : frozenset({'q002', 'q005', 'q009', 'q007', 'q004', 'q001', 'q006'})
(frozenset({'q002', 'q005', 'q009', 'q007', 'q004', 'q001', 'q006'}), 'a') : frozenset({'q002', 'q003', 'q007', 'q004', 'q001', 'q008', 'q006'})
(frozenset({'q002', 'q005', 'q009', 'q007', 'q004', 'q001', 'q006'}), 'b') : frozenset({'q002', 'q005', 'q007', 'q004', 'q001', 'q006'})

StartState: frozenset({'q002', 'q000', 'q007', 'q004', 'q001'})

FinalState:
frozenset({'q002', 'q005', 'q009', 'q007', 'q004', 'q001', 'q006'})
```

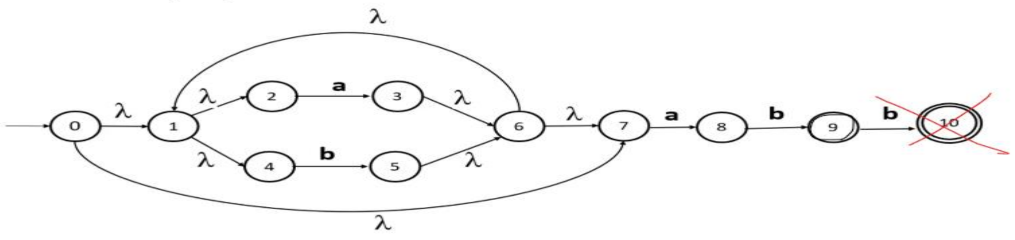


```
-----------------------------------------------------------------------------
                         --minimized_dfa--
StateSet:
(frozenset({'q002', 'q005', 'q009', 'q007', 'q004', 'q001', 'q006'}),)
(frozenset({'q002', 'q000', 'q007', 'q004', 'q001'}), frozenset({'q002', 'q005', 'q007', 'q004', 'q001', 'q006'}))
(frozenset({'q002', 'q003', 'q007', 'q004', 'q001', 'q008', 'q006'}),)

TerminalSet:
frozenset({'a', 'b'})

DeltaFunctions:
((frozenset({'q002', 'q005', 'q009', 'q007', 'q004', 'q001', 'q006'}),), 'a') : [(frozenset({'q002', 'q003', 'q007', 'q004', 'q001', 'q008', 'q006'}),)]
((frozenset({'q002', 'q005', 'q009', 'q007', 'q004', 'q001', 'q006'}),), 'b') : [(frozenset({'q002', 'q000', 'q007', 'q004', 'q001'}), frozenset({'q002', 'q005', 'q007', 'q004', 'q001', 'q006'}))]
((frozenset({'q002', 'q000', 'q007', 'q004', 'q001'}), frozenset({'q002', 'q005', 'q007', 'q004', 'q001', 'q006'})), 'a') : [(frozenset({'q002', 'q003', 'q007', 'q004', 'q001', 'q008', 'q006'}),)]
((frozenset({'q002', 'q000', 'q007', 'q004', 'q001'}), frozenset({'q002', 'q005', 'q007', 'q004', 'q001', 'q006'})), 'b') : [(frozenset({'q002', 'q000', 'q007', 'q004', 'q001'}), frozenset({'q002', 'q005', 'q007', 'q004', 'q001', 'q006'}))]
((frozenset({'q002', 'q003', 'q007', 'q004', 'q001', 'q008', 'q006'}),), 'a') : [(frozenset({'q002', 'q003', 'q007', 'q004', 'q001', 'q008', 'q006'}),)]
((frozenset({'q002', 'q003', 'q007', 'q004', 'q001', 'q008', 'q006'}),), 'b') : [(frozenset({'q002', 'q005', 'q009', 'q007', 'q004', 'q001', 'q006'}),)]

StartState: (frozenset({'q002', 'q000', 'q007', 'q004', 'q001'}), frozenset({'q002', 'q005', 'q007', 'q004', 'q001', 'q006'}))

FinalState:
(frozenset({'q002', 'q005', 'q009', 'q007', 'q004', 'q001', 'q006'}),)
-----------------------------------------------------------------------------
```

검증

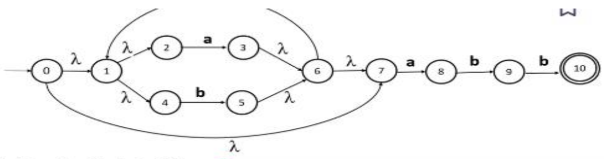
Example 1

- Convert the RE (a+b)*abb to a minimized DFA



λ-closure(0) = {0, 1, 2, 4, 7}	λ-closure(4) = {4}	λ-closure(8) = {8}
λ-closure(1) = {1, 2, 4}	λ-closure(5) = {1, 2, 4, 5, 6, 7}	λ-closure(9) = {9}
λ-closure(2) = {2}	λ-closure(6) = {1, 2, 4, 6, 7}	~~λ-closure(10) = {10}~~
λ-closure(3) = {1, 2, 3, 4, 6, 7}	λ-closure(7) = {7}	

Example 1



λ-closure(0) = {0, 1, 2, 4, 7} – **A**
λ-closure(δ(A, a)) = λ-closure(3, 8) = {1, 2, 3, 4, 6, 7, 8} – **B**
λ-closure(δ(A, b)) = λ-closure(5) = {1, 2, 4, 5, 6, 7} – **C**
λ-closure(δ(B, a)) = λ-closure(3, 8) = {1, 2, 3, 4, 6, 7, 8} – **B**
λ-closure(δ(B, b)) = λ-closure(5, 9) = {1, 2, 4, 5, 6, 7, 9} – **D**
λ-closure(δ(C, a)) = λ-closure(3, 8) = {1, 2, 3, 4, 6, 7, 8} – **B**
λ-closure(δ(C, b)) = λ-closure(5) = {1, 2, 4, 5, 6, 7} – **C**
λ-closure(δ(D, a)) = λ-closure(3, 8) = {1, 2, 3, 4, 6, 7, 8} – **B**
~~λ-closure(δ(D, b)) = λ-closure(5, 10) = {1, 2, 4, 5, 6, 7, 10} – E~~
λ-closure(δ(E, a)) = λ-closure(3, 8) = {1, 2, 3, 4, 6, 7, 8} – **B**
λ-closure(δ(E, b)) = λ-closure(5) = {1, 2, 4, 5, 6, 7} – **C**

DFA Minimization – Tabulation Method

States	a	b
-> A	B	C
B	B	D
C	B	C
D	B	E
~~* E~~	~~B~~	~~C~~

Mark/Reduce Procedure

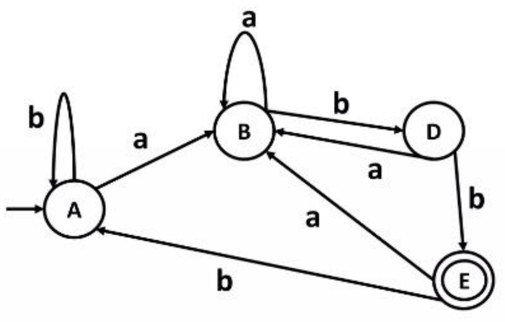
B	✗			
C		✗		
D	✗	✗	✗	
E	✗	✗	✗	✗
	A	B	C	D

Pair – (AC)

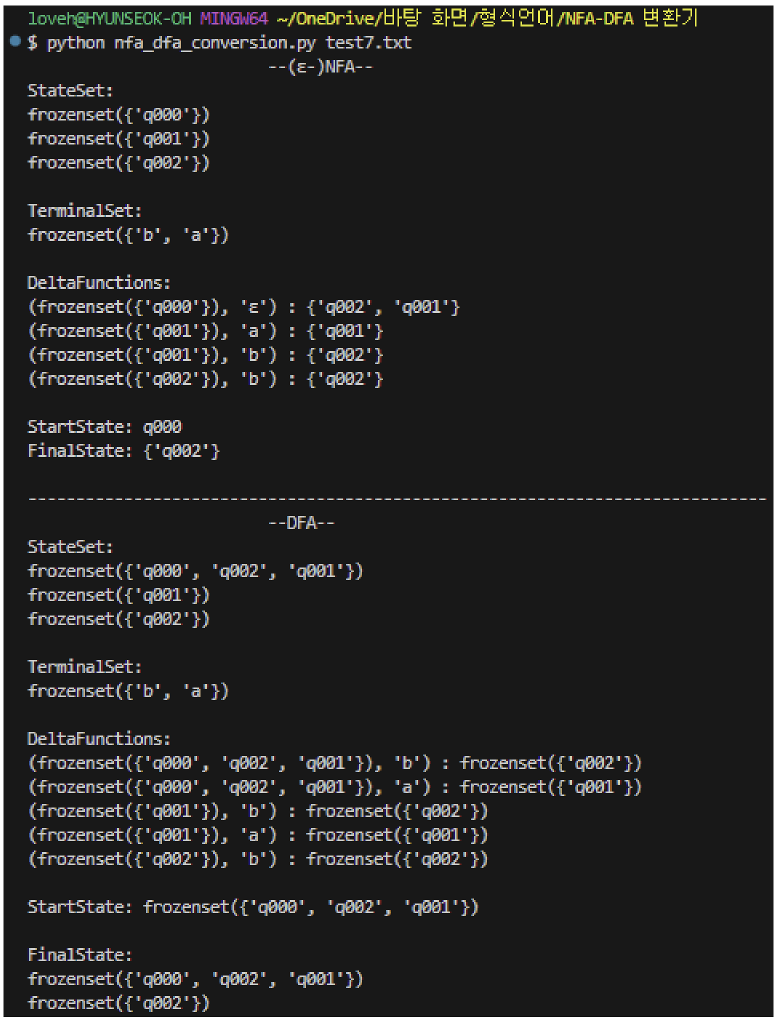
Minimized DFA

States	a	b
-> A	B	C
B	B	D
C	B	C
D	B	E
~~*E~~	~~B~~	~~C~~

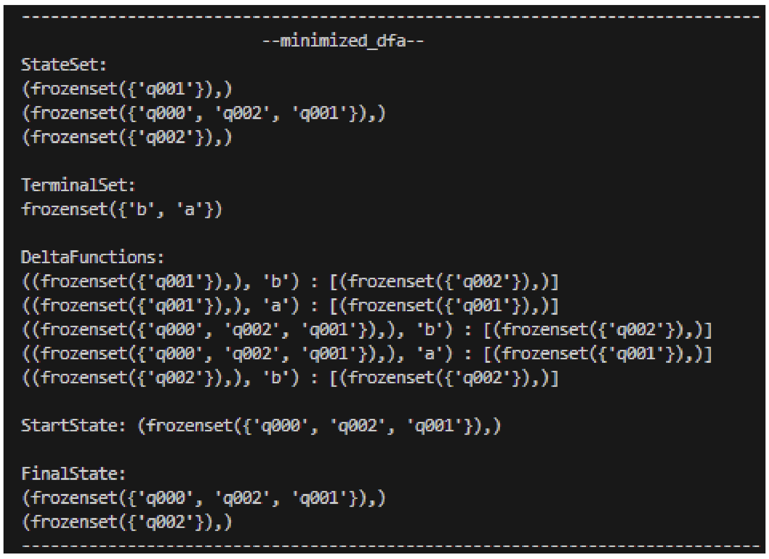
States	a	b
-> A	B	A
B	B	D
D	B	E
~~*E~~	~~B~~	~~A~~



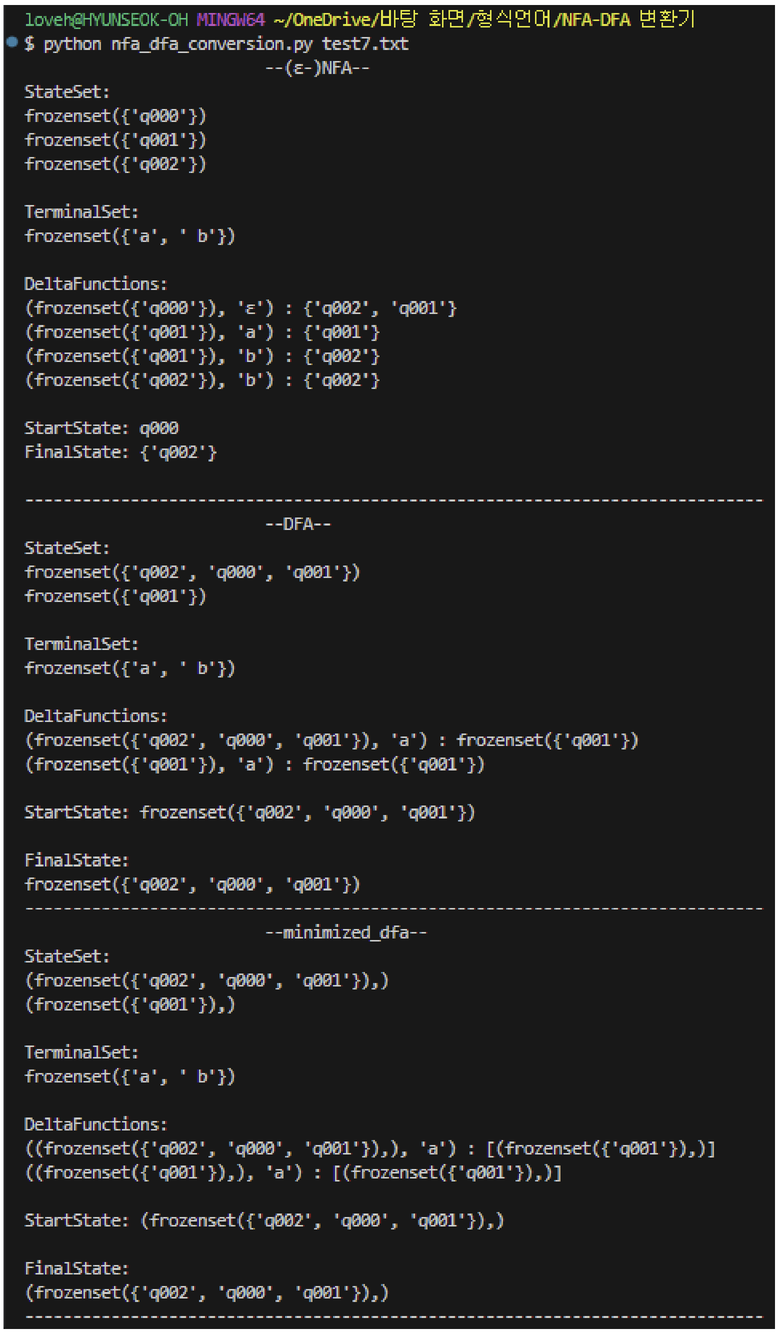
7) test7.txt



```
loveh@HYUNSEOK-OH MINGW64 ~/OneDrive/바탕 화면/형식언어/NFA-DFA 변환기
$ python nfa_dfa_conversion.py test7.txt
                          --(ε-)NFA--
StateSet:
frozenset({'q000'})
frozenset({'q001'})
frozenset({'q002'})

TerminalSet:
frozenset({'b', 'a'})

DeltaFunctions:
(frozenset({'q000'}), 'ε') : {'q002', 'q001'}
(frozenset({'q001'}), 'a') : {'q001'}
(frozenset({'q001'}), 'b') : {'q002'}
(frozenset({'q002'}), 'b') : {'q002'}

StartState: q000
FinalState: {'q002'}

----------------------------------------------------------------------------
                          --DFA--
StateSet:
frozenset({'q000', 'q002', 'q001'})
frozenset({'q001'})
frozenset({'q002'})

TerminalSet:
frozenset({'b', 'a'})

DeltaFunctions:
(frozenset({'q000', 'q002', 'q001'}), 'b') : frozenset({'q002'})
(frozenset({'q000', 'q002', 'q001'}), 'a') : frozenset({'q001'})
(frozenset({'q001'}), 'b') : frozenset({'q002'})
(frozenset({'q001'}), 'a') : frozenset({'q001'})
(frozenset({'q002'}), 'b') : frozenset({'q002'})

StartState: frozenset({'q000', 'q002', 'q001'})

FinalState:
frozenset({'q000', 'q002', 'q001'})
frozenset({'q002'})
```

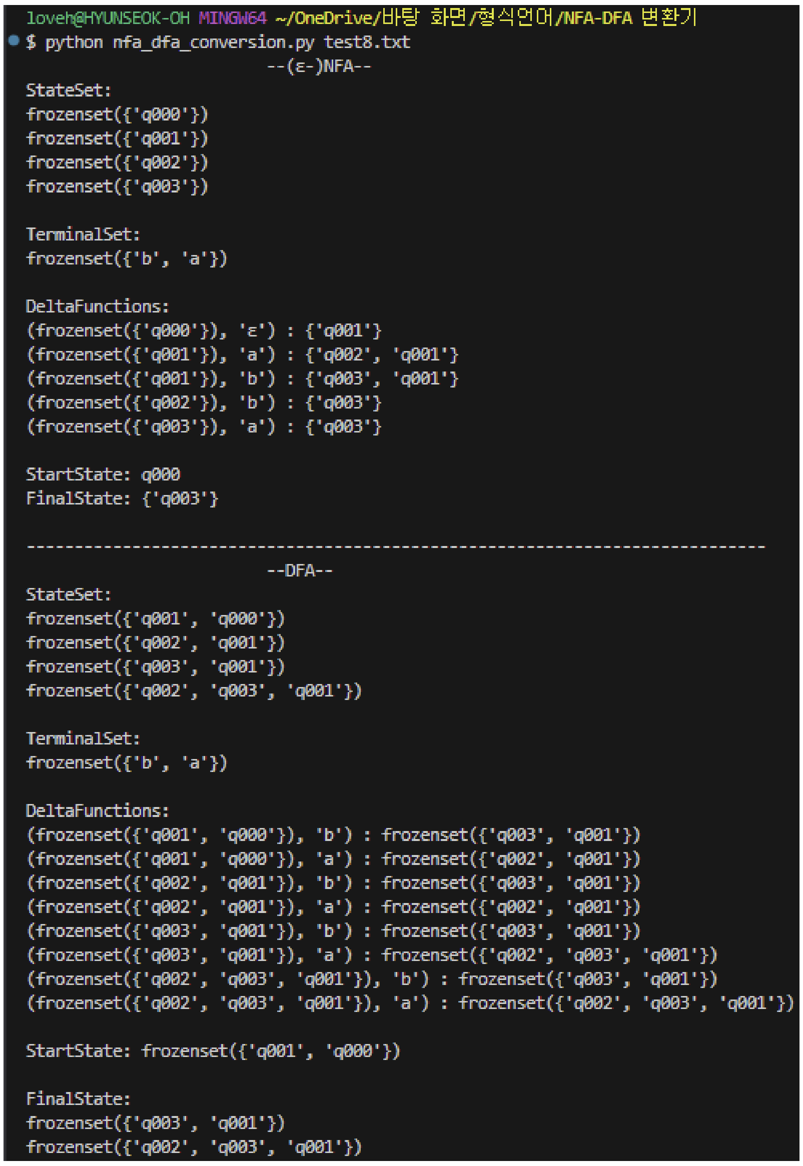


```
----------------------------------------------------------------------------
                         --minimized_dfa--
StateSet:
(frozenset({'q001'}),)
(frozenset({'q000', 'q002', 'q001'}),)
(frozenset({'q002'}),)

TerminalSet:
frozenset({'b', 'a'})

DeltaFunctions:
((frozenset({'q001'}),), 'b') : [(frozenset({'q002'}),)]
((frozenset({'q001'}),), 'a') : [(frozenset({'q001'}),)]
((frozenset({'q000', 'q002', 'q001'}),), 'b') : [(frozenset({'q002'}),)]
((frozenset({'q000', 'q002', 'q001'}),), 'a') : [(frozenset({'q001'}),)]
((frozenset({'q002'}),), 'b') : [(frozenset({'q002'}),)]

StartState: (frozenset({'q000', 'q002', 'q001'}),)

FinalState:
(frozenset({'q000', 'q002', 'q001'}),)
(frozenset({'q002'}),)
----------------------------------------------------------------------------
```

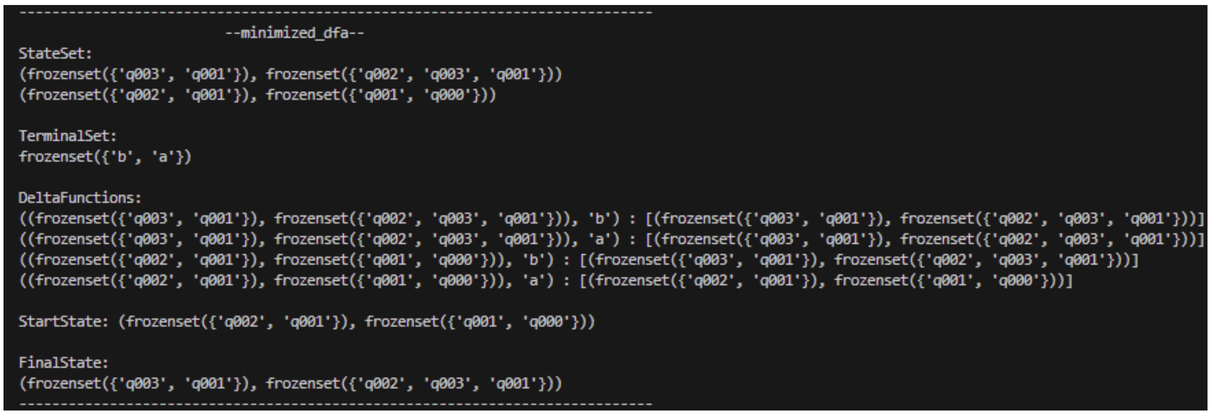


```
loveh@HYUNSEOK-OH MINGW64 ~/OneDrive/바탕 화면/형식언어/NFA-DFA 변환기
$ python nfa_dfa_conversion.py test7.txt
                         --(ε-)NFA--
StateSet:
frozenset({'q000'})
frozenset({'q001'})
frozenset({'q002'})

TerminalSet:
frozenset({'a', ' b'})

DeltaFunctions:
(frozenset({'q000'}), 'ε') : {'q002', 'q001'}
(frozenset({'q001'}), 'a') : {'q001'}
(frozenset({'q001'}), 'b') : {'q002'}
(frozenset({'q002'}), 'b') : {'q002'}

StartState: q000
FinalState: {'q002'}

-----------------------------------------------------------------------------
                         --DFA--
StateSet:
frozenset({'q002', 'q000', 'q001'})
frozenset({'q001'})

TerminalSet:
frozenset({'a', ' b'})

DeltaFunctions:
(frozenset({'q002', 'q000', 'q001'}), 'a') : frozenset({'q001'})
(frozenset({'q001'}), 'a') : frozenset({'q001'})

StartState: frozenset({'q002', 'q000', 'q001'})

FinalState:
frozenset({'q002', 'q000', 'q001'})
-----------------------------------------------------------------------------
                         --minimized_dfa--
StateSet:
(frozenset({'q002', 'q000', 'q001'}),)
(frozenset({'q001'}),)

TerminalSet:
frozenset({'a', ' b'})

DeltaFunctions:
((frozenset({'q002', 'q000', 'q001'}),), 'a') : [(frozenset({'q001'}),)]
((frozenset({'q001'}),), 'a') : [(frozenset({'q001'}),)]

StartState: (frozenset({'q002', 'q000', 'q001'}),)

FinalState:
(frozenset({'q002', 'q000', 'q001'}),)
-----------------------------------------------------------------------------
```

8) test8.txt

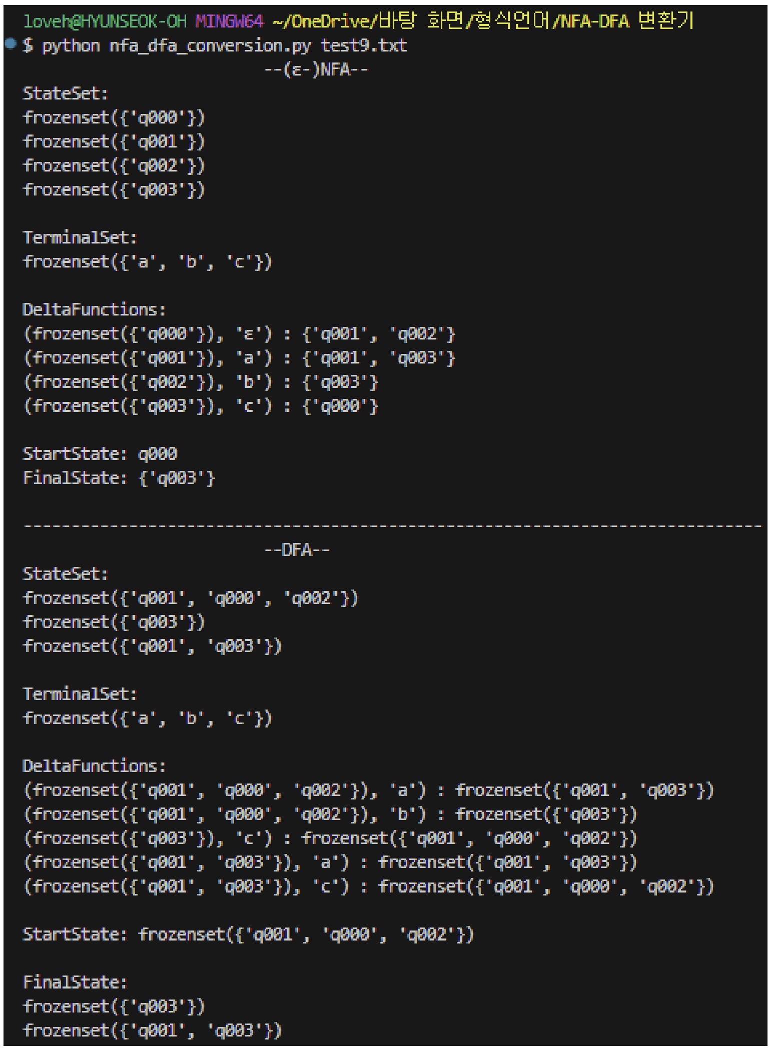


```
 loveh@HYUNSEOK-OH MINGW64 ~/OneDrive/바탕 화면/형식언어/NFA-DFA 변환기
● $ python nfa_dfa_conversion.py test8.txt
                             --(ε-)NFA--
 StateSet:
 frozenset({'q000'})
 frozenset({'q001'})
 frozenset({'q002'})
 frozenset({'q003'})

 TerminalSet:
 frozenset({'b', 'a'})

 DeltaFunctions:
 (frozenset({'q000'}), 'ε') : {'q001'}
 (frozenset({'q001'}), 'a') : {'q002', 'q001'}
 (frozenset({'q001'}), 'b') : {'q003', 'q001'}
 (frozenset({'q002'}), 'b') : {'q003'}
 (frozenset({'q003'}), 'a') : {'q003'}

 StartState: q000
 FinalState: {'q003'}

 -----------------------------------------------------------------------------
                             --DFA--
 StateSet:
 frozenset({'q001', 'q000'})
 frozenset({'q002', 'q001'})
 frozenset({'q003', 'q001'})
 frozenset({'q002', 'q003', 'q001'})

 TerminalSet:
 frozenset({'b', 'a'})

 DeltaFunctions:
 (frozenset({'q001', 'q000'}), 'b') : frozenset({'q003', 'q001'})
 (frozenset({'q001', 'q000'}), 'a') : frozenset({'q002', 'q001'})
 (frozenset({'q002', 'q001'}), 'b') : frozenset({'q003', 'q001'})
 (frozenset({'q002', 'q001'}), 'a') : frozenset({'q002', 'q001'})
 (frozenset({'q003', 'q001'}), 'b') : frozenset({'q003', 'q001'})
 (frozenset({'q003', 'q001'}), 'a') : frozenset({'q002', 'q003', 'q001'})
 (frozenset({'q002', 'q003', 'q001'}), 'b') : frozenset({'q003', 'q001'})
 (frozenset({'q002', 'q003', 'q001'}), 'a') : frozenset({'q002', 'q003', 'q001'})

 StartState: frozenset({'q001', 'q000'})

 FinalState:
 frozenset({'q003', 'q001'})
 frozenset({'q002', 'q003', 'q001'})
```

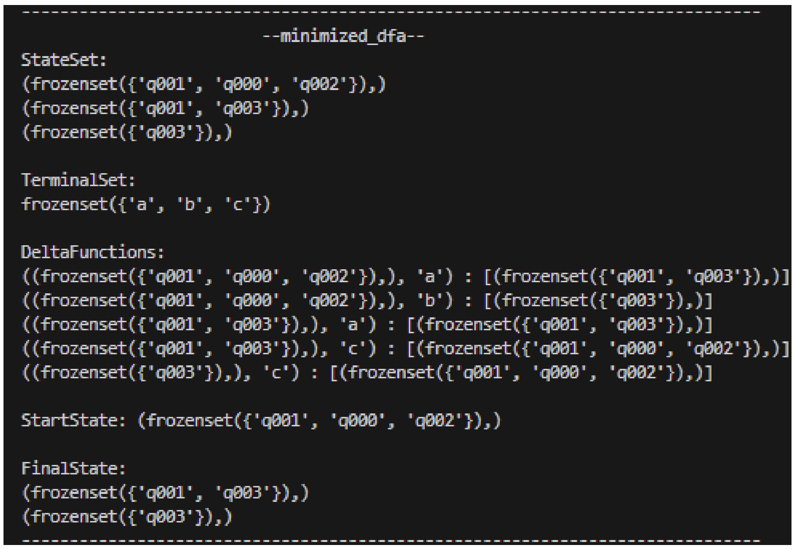


```
-----------------------------------------------------------------------------
                           --minimized_dfa--
StateSet:
(frozenset({'q003', 'q001'}), frozenset({'q002', 'q003', 'q001'}))
(frozenset({'q002', 'q001'}), frozenset({'q001', 'q000'}))

TerminalSet:
frozenset({'b', 'a'})

DeltaFunctions:
((frozenset({'q003', 'q001'}), frozenset({'q002', 'q003', 'q001'})), 'b') : [(frozenset({'q003', 'q001'}), frozenset({'q002', 'q003', 'q001'}))]
((frozenset({'q003', 'q001'}), frozenset({'q002', 'q003', 'q001'})), 'a') : [(frozenset({'q003', 'q001'}), frozenset({'q002', 'q003', 'q001'}))]
((frozenset({'q002', 'q001'}), frozenset({'q001', 'q000'})), 'b') : [(frozenset({'q003', 'q001'}), frozenset({'q002', 'q003', 'q001'}))]
((frozenset({'q002', 'q001'}), frozenset({'q001', 'q000'})), 'a') : [(frozenset({'q002', 'q001'}), frozenset({'q001', 'q000'}))]

StartState: (frozenset({'q002', 'q001'}), frozenset({'q001', 'q000'}))

FinalState:
(frozenset({'q003', 'q001'}), frozenset({'q002', 'q003', 'q001'}))
-----------------------------------------------------------------------------
```

9)test9.txt

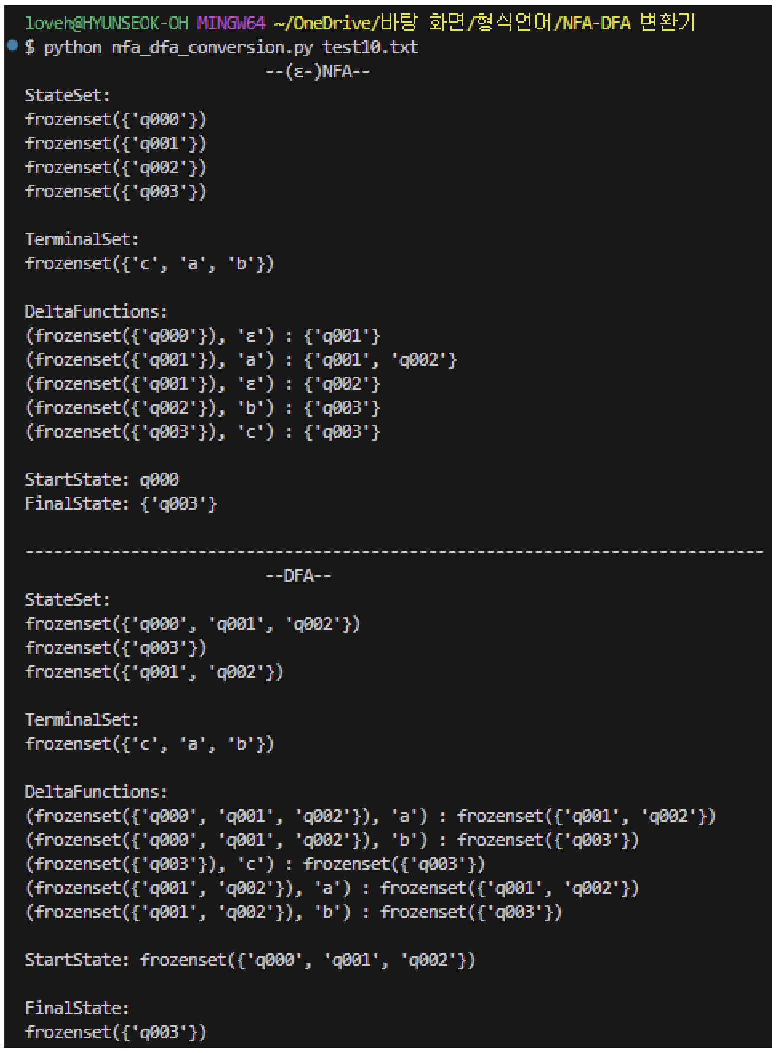


```
loveh@HYUNSEOK-OH MINGW64 ~/OneDrive/바탕 화면/형식언어/NFA-DFA 변환기
$ python nfa_dfa_conversion.py test9.txt
                         --(ε-)NFA--
StateSet:
frozenset({'q000'})
frozenset({'q001'})
frozenset({'q002'})
frozenset({'q003'})

TerminalSet:
frozenset({'a', 'b', 'c'})

DeltaFunctions:
(frozenset({'q000'}), 'ε') : {'q001', 'q002'}
(frozenset({'q001'}), 'a') : {'q001', 'q003'}
(frozenset({'q002'}), 'b') : {'q003'}
(frozenset({'q003'}), 'c') : {'q000'}

StartState: q000
FinalState: {'q003'}

-----------------------------------------------------------------------------
                         --DFA--
StateSet:
frozenset({'q001', 'q000', 'q002'})
frozenset({'q003'})
frozenset({'q001', 'q003'})

TerminalSet:
frozenset({'a', 'b', 'c'})

DeltaFunctions:
(frozenset({'q001', 'q000', 'q002'}), 'a') : frozenset({'q001', 'q003'})
(frozenset({'q001', 'q000', 'q002'}), 'b') : frozenset({'q003'})
(frozenset({'q003'}), 'c') : frozenset({'q001', 'q000', 'q002'})
(frozenset({'q001', 'q003'}), 'a') : frozenset({'q001', 'q003'})
(frozenset({'q001', 'q003'}), 'c') : frozenset({'q001', 'q000', 'q002'})

StartState: frozenset({'q001', 'q000', 'q002'})

FinalState:
frozenset({'q003'})
frozenset({'q001', 'q003'})
```

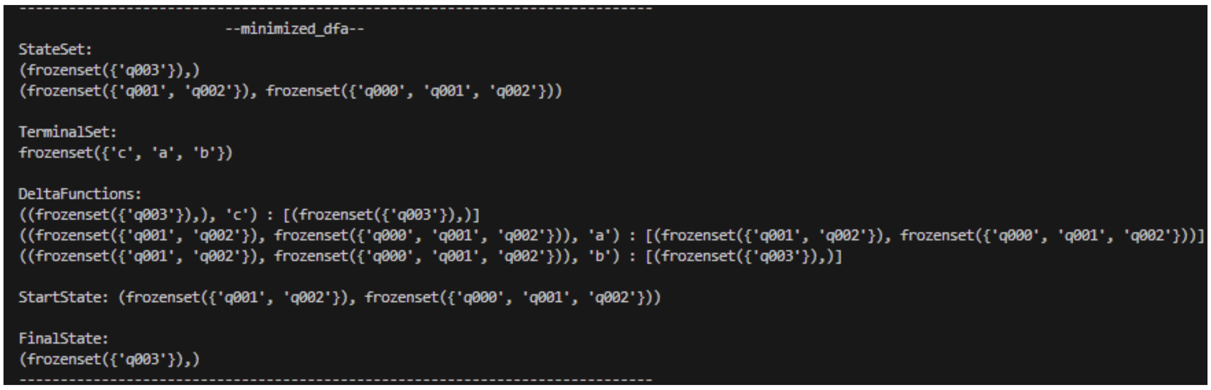


```
-----------------------------------------------------------------------------
                         --minimized_dfa--
StateSet:
(frozenset({'q001', 'q000', 'q002'}),)
(frozenset({'q001', 'q003'}),)
(frozenset({'q003'}),)

TerminalSet:
frozenset({'a', 'b', 'c'})

DeltaFunctions:
((frozenset({'q001', 'q000', 'q002'}),), 'a') : [(frozenset({'q001', 'q003'}),)]
((frozenset({'q001', 'q000', 'q002'}),), 'b') : [(frozenset({'q003'}),)]
((frozenset({'q001', 'q003'}),), 'a') : [(frozenset({'q001', 'q003'}),)]
((frozenset({'q001', 'q003'}),), 'c') : [(frozenset({'q001', 'q000', 'q002'}),)]
((frozenset({'q003'}),), 'c') : [(frozenset({'q001', 'q000', 'q002'}),)]

StartState: (frozenset({'q001', 'q000', 'q002'}),)

FinalState:
(frozenset({'q001', 'q003'}),)
(frozenset({'q003'}),)
-----------------------------------------------------------------------------
```

10) test10.txt



```
loveh@HYUNSEOK-OH MINGW64 ~/OneDrive/바탕 화면/형식언어/NFA-DFA 변환기
$ python nfa_dfa_conversion.py test10.txt
                              --(ε-)NFA--
StateSet:
frozenset({'q000'})
frozenset({'q001'})
frozenset({'q002'})
frozenset({'q003'})

TerminalSet:
frozenset({'c', 'a', 'b'})

DeltaFunctions:
(frozenset({'q000'}), 'ε') : {'q001'}
(frozenset({'q001'}), 'a') : {'q001', 'q002'}
(frozenset({'q001'}), 'ε') : {'q002'}
(frozenset({'q002'}), 'b') : {'q003'}
(frozenset({'q003'}), 'c') : {'q003'}

StartState: q000
FinalState: {'q003'}

-----------------------------------------------------------------------------
                              --DFA--
StateSet:
frozenset({'q000', 'q001', 'q002'})
frozenset({'q003'})
frozenset({'q001', 'q002'})

TerminalSet:
frozenset({'c', 'a', 'b'})

DeltaFunctions:
(frozenset({'q000', 'q001', 'q002'}), 'a') : frozenset({'q001', 'q002'})
(frozenset({'q000', 'q001', 'q002'}), 'b') : frozenset({'q003'})
(frozenset({'q003'}), 'c') : frozenset({'q003'})
(frozenset({'q001', 'q002'}), 'a') : frozenset({'q001', 'q002'})
(frozenset({'q001', 'q002'}), 'b') : frozenset({'q003'})

StartState: frozenset({'q000', 'q001', 'q002'})

FinalState:
frozenset({'q003'})
```



```
-----------------------------------------------------------------------------
                          --minimized_dfa--
StateSet:
(frozenset({'q003'}),)
(frozenset({'q001', 'q002'}), frozenset({'q000', 'q001', 'q002'}))

TerminalSet:
frozenset({'c', 'a', 'b'})

DeltaFunctions:
((frozenset({'q003'}),), 'c') : [(frozenset({'q003'}),)]
((frozenset({'q001', 'q002'}), frozenset({'q000', 'q001', 'q002'})), 'a') : [(frozenset({'q001', 'q002'}), frozenset({'q000', 'q001', 'q002'}))]
((frozenset({'q001', 'q002'}), frozenset({'q000', 'q001', 'q002'})), 'b') : [(frozenset({'q003'}),)]

StartState: (frozenset({'q001', 'q002'}), frozenset({'q000', 'q001', 'q002'}))

FinalState:
(frozenset({'q003'}),)
-----------------------------------------------------------------------------
```

5. 소감