LO's 5 & 6 from previous Semesters
Exam 2

Nondeterministic Finite Automata

Last Updated: August 11th, 2026

1 Nondeterministic Finite Automata

To prove the closure of regular languages under union, concatenation, and star, it will be of great help to use the concept of a nondeterministic finite automata (NFA). An NFA has almost the same definition as a DFA, with one exception: upon reading an input symbol, a state transition allows for more than one possible next state. More formally, the codomain of function δ is no longer Q, but rather $\mathcal{P}(Q)$, the set of all subsets of Q. This means that the output is a set of states rather than a single state.

A **nondeterministic finite automaton (NFA)** consists of a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

Q is a finite set with each member of Q called a **state**

Σ a finite alphabet that is used to represent an input word

δ a transition function that determines the set of possible next states of the automaton given the current state and the current input symbol being read. In other words,

$$\boxed{\delta : Q \times \Sigma \to \mathcal{P}(Q),}$$

is a mapping from (current) state-input pairs to a *subset* of possible next states.

$q_0 \in Q$ the **initial state**.

$F \subseteq Q$ a member of F is called an **accepting state**.

Note that every DFA is technically an NFA for which the transition function maps each state-input pair to a subset of size 1. Indeed, DFA transition function statement $\delta(q, s) = r$ can be expressed like an NFA statement by writing

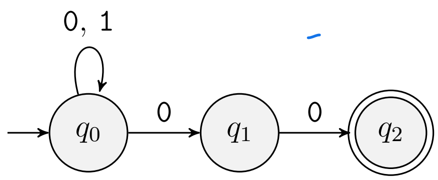
$$\delta(q, s) = \{r\} \in \mathcal{P}(Q).$$

Example 1.1. The following state diagram provides an example of an NFA N that is not a DFA, since

1. state q_0 has two different outgoing edges that are labeled with 0
2. states q_1 and q_2 are both missing outgoing edges. This is done to simplify the state diagram.

The convention is, if N is in state q, reading input symbol s, and there is no outgoing edge from state q that is labeled with s, then the default is $\boxed{\delta(q, s) = \emptyset.}$

$\delta(q_2, 0) = \emptyset \in \mathcal{P}(\{q_0, q_1, q_2\})$



The following table represents transition function δ.

$q_i \backslash s$	0	1
q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\emptyset$
q_2	$\emptyset$	$\emptyset$

1.1 NFA computation

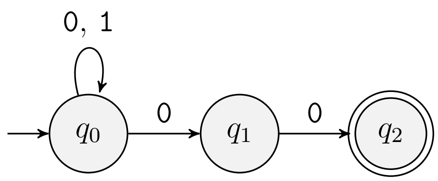
Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA, and $w = w_1 w_2 \cdots w_n$ be a word in Σ^*. Then the **computation of N on input w** is a sequence $S_0, S_1, S_2, \ldots, S_n$ of subsets of states that are recursively defined as follows.

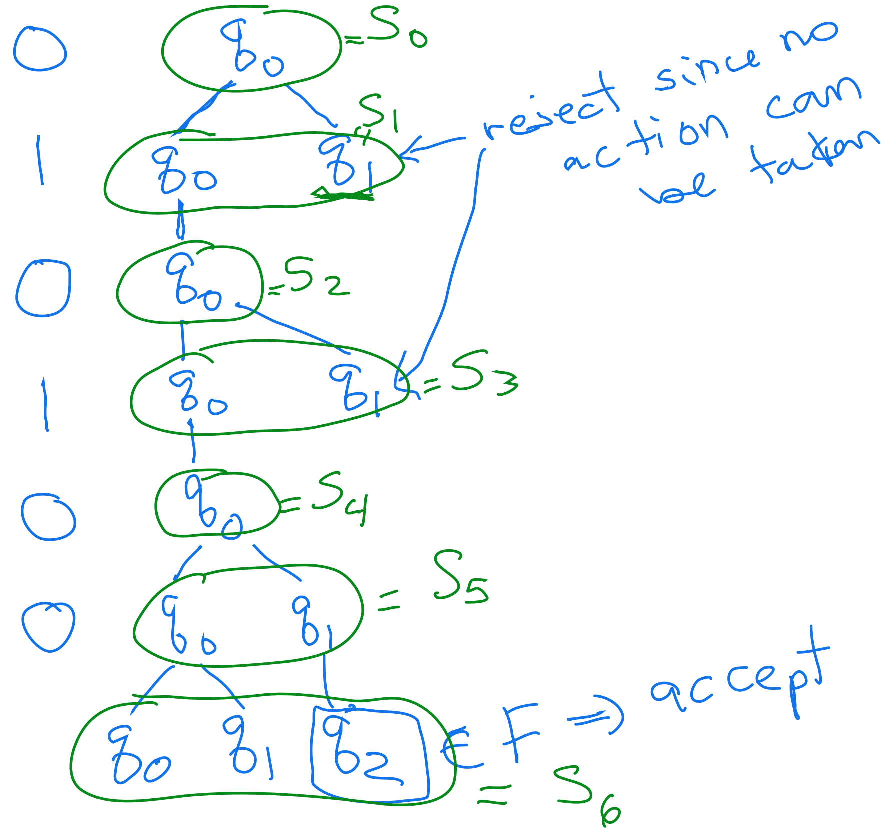
1. $S_0 = \{q_0\}$ is the singleton set consisting of the initial state q_0 of M
2. $S_i = \bigcup_{q \in S_{i-1}} \delta(q, w_i)$ for every $1 \leq i \leq n$.

In words, the computation of N on input w is the sequence of state subsets $S_0, S_1, S_2, \ldots, S_n$ that N moves through when successively reading input symbols $w_1, w_2, \ldots, w_n$. The computation of N on input w is said to be **accepting** iff $S_n \cap F \neq \emptyset$. In other words, at least one accepting state belongs in S_n. In this case we say N **accepts** w. Otherwise it is called a **rejecting** computation, and N **rejects** w.

Rather than say that an NFA is in a particular state, we instead say that it is in a particular subset S of states.

Example 1.2. For the NFA N shown below, show the computation of N on inputs 010100 and 11010. For both inputs, show the sequence of state subsets that corresponds with each of the input symbols being read. In the case of 010100, also show the computation tree that is formed by the computation.





0	1	0	1	0	0	
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$

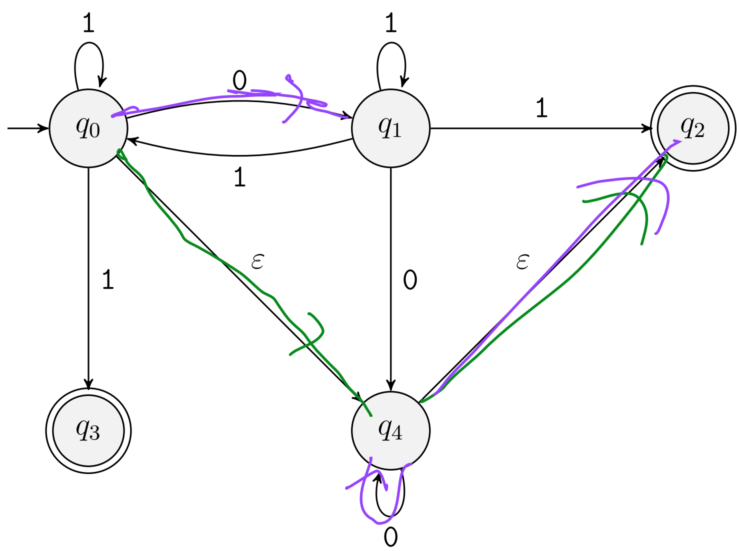
accept since $q_2 \in F$

1.2 ε-Edges

Another difference between an NFA N's state diagram and a DFA diagram is that it may have one or more **ε-edges**, i.e. an edge that is labeled with ε. Moreover, an **ε-path** is any sequence of ε-edges in the state diagram that forms a path in the diagram. The rules pertaining to ε edges are as follows.

1. An ε-edge from state q is only activated when transitioning *into* q, and *not* when transitioning from q.
2. If there is an ε-edge from q_1 to q_2 and S is a state subset in the δ-transition table, then $q_1 \in S$ implies $q_2 \in S$.
3. More generally, if there is a vertex q for which an ε-path starts at q and S is a state subset in the δ-transition table, then $q \in S$ implies that every other state in the ε-path must also be in S.
4. The initial subset state of any computation is the set consisting of intial state q_0 along with any state that is reachable from q_0 along some ε-path.

Example 1.3. Provide the table of the δ-transition function for the following NFA, and show the computation on i) $w = 010$ and ii) $w = 101$.



δ: $Q \setminus \Sigma$	0	1
q_0	$\{q_1\}$	$\{q_0, q_3, q_2, q_4\}$
q_1	$\{q_2, q_4\}$	$\{q_0, q_4, q_2, q_1\}$
q_2	$\emptyset$	$\emptyset$
q_3	$\emptyset$	$\emptyset$
q_4	$\{q_2, q_4\}$	$\emptyset$

0 1 0

$q_0 q_2 q_4$ $q_1 q_2 q_4$ $q_0 q_1 q_2 q_4$ $q_1 q_2 q_4 \Rightarrow$ accept

1.3 Equivalence between DFA's and NFA's

Since every DFA is also an NFA, it follows that every regular language L is accepted by some NFA. The converse is also true! Every language L that is accepted by an NFA is regular.

This is because an NFA is actually a DFA in disguise. The reason why an NFA is not a DFA is that it does not meet the transition-function definition for DFA's. That definition states

that

$$\delta : Q \times \Sigma \to Q$$

must map a state-input pair to a state. But an NFA N maps a state-input pair to a subset of states. Recalling that N's formal definition is

$$N = (Q, \Sigma, \delta : Q \times \Sigma \to \mathcal{P}(Q), q_0, F),$$

we can now do the following to create a DFA N'.

The state set is now $\mathcal{P}(Q)$: every state of N' is a subset of the original set of states.

Create a new transition function: instead of $\delta : \underline{Q} \times \Sigma \to \mathcal{P}(Q)$ we now have

$\delta'(\{a,b,c\}, s) = \delta(a,s) \cup \delta(b,s) \cup \delta(c,s)$

$$\delta' : \underline{\mathcal{P}(Q)} \times \Sigma \to \mathcal{P}(Q),$$

which is now a DFA transition function because it maps a state-input pair (S, a) to a state T, where S and T are subsets of states belonging to N.

$\{q_0\} \in \mathcal{P}(Q)$

Create a new initial state: the initial state is now the singleton set $\underline{\{q_0\}}$.

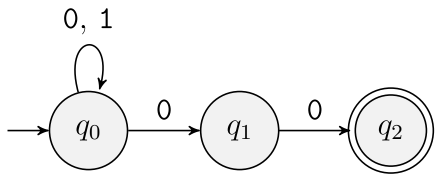
Create new accepting states: instead of F, we have F' where subset $S \in F'$ iff $S \cap F \neq \emptyset$. In other words, a subset of states is an accepting state for N' iff it contains some accepting state of N.

The new DFA N' may now be formally expressed as

$$N' = (\mathcal{P}(Q), \Sigma, \delta', \{q_0\}, F').$$

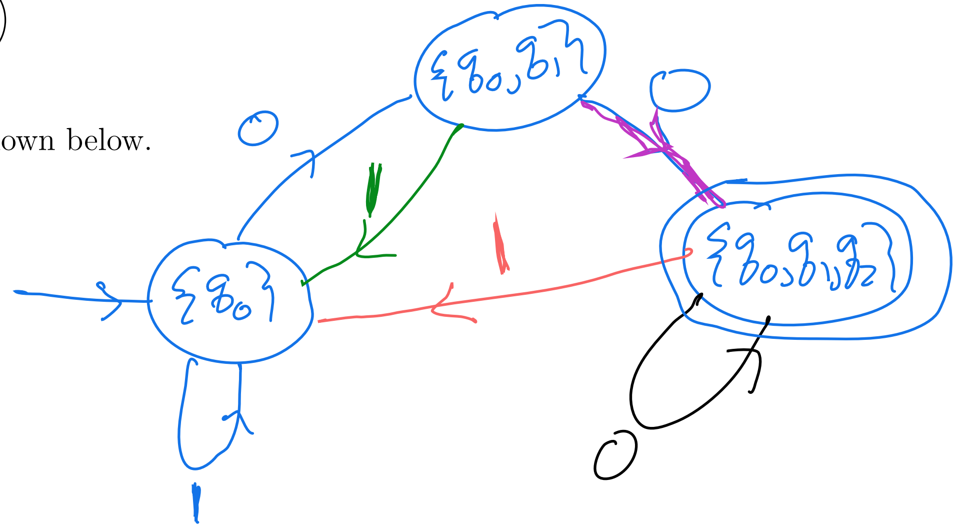
Moreover, the computation of N' on some input word w is defined *exactly* in the same way as the computation of N on w. In other words, N accepts w iff N' accepts w. To see this, as an exercise, write out the definition of what it means to be an accepting computation for a DFA, but within the context of machine N'. Then show that it is identical to the definition of what it means to be an accepting computation for NFA N. Both definitions should appear identical.

Example 1.4. Convert the following NFA to a DFA

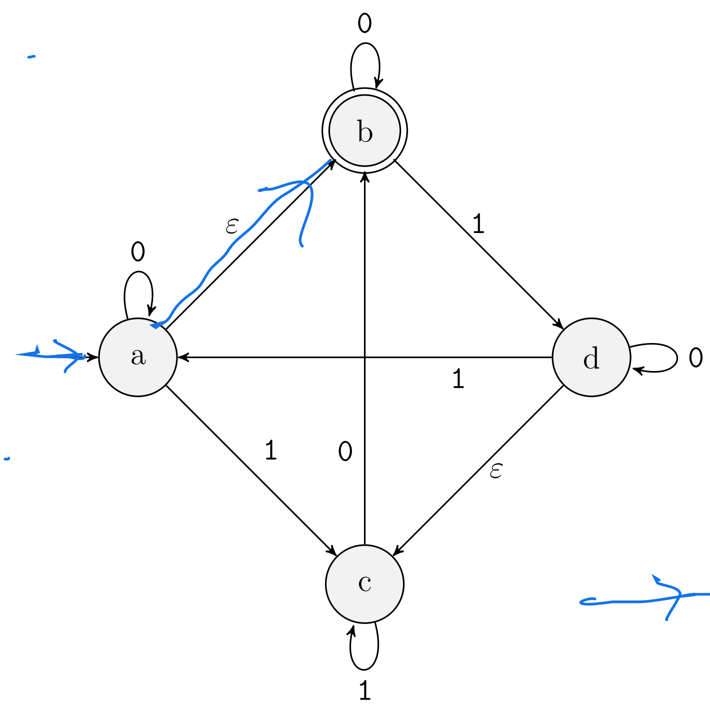


δ whose transition function δ is shown below.

$q_i \backslash s$	0	1
q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\emptyset$
q_2	$\emptyset$	$\emptyset$

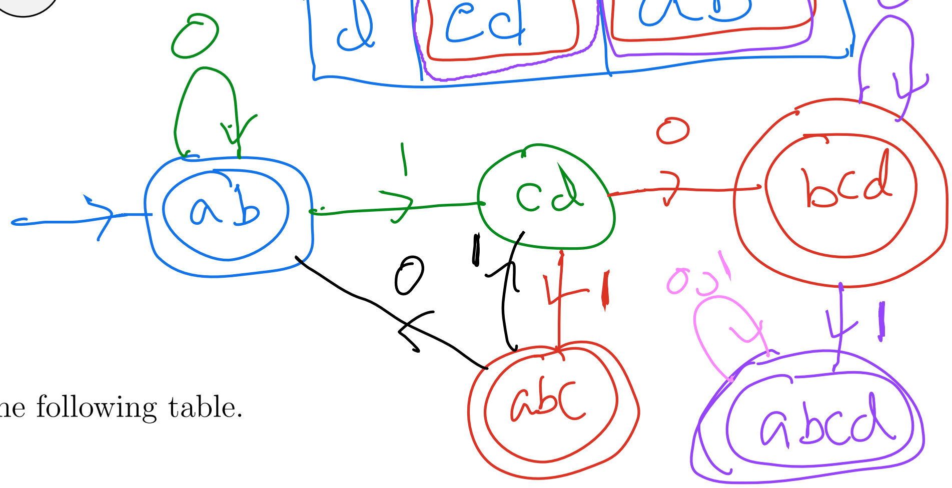


Example 1.5. Convert the following NFA to a DFA by using the method of subset states.



$Q \backslash \Sigma$	0	1
a	ab	c
b	b	cd
c	b	c
d	cd	ab

$\emptyset$



It's transition function is provided in the following table.

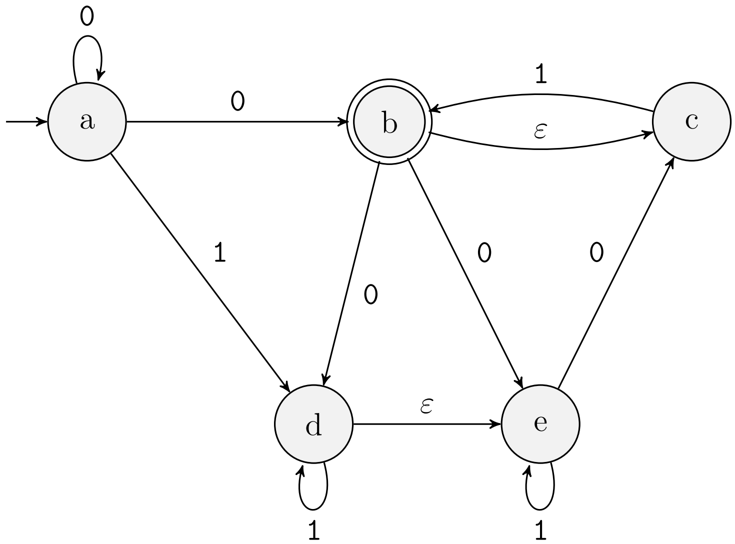
$q_i \backslash s$	0	1
a	$\{a, b\}$	$\{c\}$
b	$\{b\}$	$\{c, d\}$
c	$\{b\}$	$\{c\}$
d	$\{c, d\}$	$\{a, b\}$

w = 1 1 0 0 1

ab cd abc ab ab cd ⇒ reject

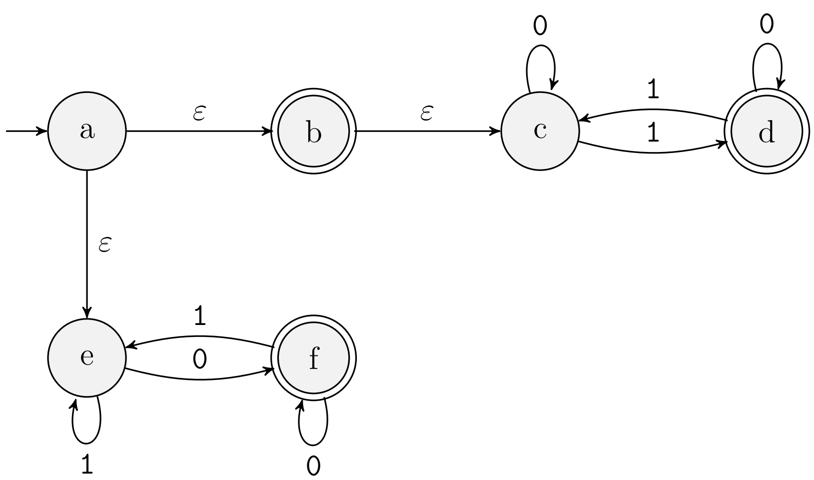
NFA Core Exercises

1. Provide the δ-transition function for the NFA whose state diagram is shown below.



2. For the NFA from the previous exercise, provide the computation on inputs 0010 and 10111. Which of the two are accepted?

3. Provide the δ-transition function for the NFA N whose state diagram is shown below.



4. For the NFA from the previous exercise, provide the computation on inputs ε, 0101 and 000. Verify that all three are accepted by N. Can you find an input that does not get accepted?

5. Provide the state diagram of an NFA that accepts the language

   (a) $\{w|w$ ends with 00$\}$ and uses only three states.

   (b) $\{w|w$ contains the substring 0101$\}$ and uses only five states.

   (c) $\{w|w$ contains an even number of zeros and exactly two ones $\}$ and uses only six states.

   (d) $\{0\}$ and uses only two states.

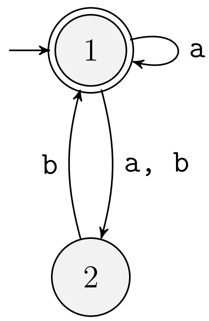
   (e) Words consisting of zero or more 0's, followed by zero or more 1's, followed by one or more 0's, and uses only three states.

(f) Words consisting of zero or more 1's, followed by zero or more words of the form "two 0's followed by one or more 1's". For example, 110010011001 is one such word. Uses only three states.

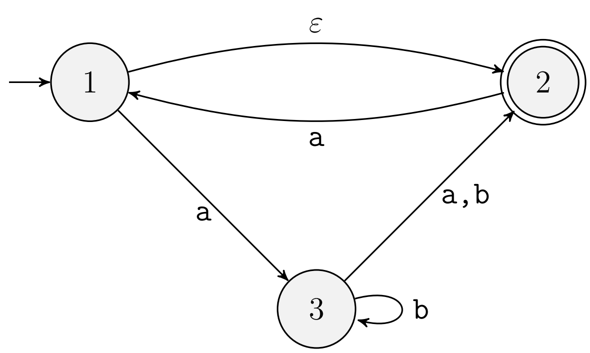
(g) ε and uses only one state.

(h) Words consisting only of zero or more 0's and uses only one state.

6. Provide a DFA M that accepts the same language as that accepted by the NFA N that is defined in the following state diagram. Do so by defining the states of M to be subsets of the states of N.



7. Provide a DFA M that accepts the same language as that accepted by the NFA N that is defined in the following state diagram. Do so by defining the states of M to be subsets of the states of N.



Solutions to NFA Core Exercises

1. The following table provides the $\delta(Q, \Sigma)$ transition function.

$Q\backslash\Sigma$	0	1
a	{a,b,c}	{d,e}
b	{d,e}	$\emptyset$
c	$\emptyset$	{b,c}
d	$\emptyset$	{d,e}
e	$\{c\}$	$\{e\}$

2. We have the following computations.

Input Symbol Read	**Current Subset State**
0	$\{a\}$
0	{a,b,c}
1	{a,b,c,d,e}
0	{b,c,d,e}
Rejecting State:	{c,d,e}

Input Symbol Read	**Current Subset State**
1	$\{a\}$
0	{d,e}
1	{c}
1	{b,c}
1	{b,c}
Accepting State:	{b,c}

3. The following table provides the $\delta(Q, \Sigma)$ transition function.

$Q\backslash\Sigma$	0	1
a	$\emptyset$	$\emptyset$
b	$\emptyset$	$\emptyset$
c	{c}	{d}
d	{d}	{c}
e	{f}	{e}
f	{f}	{e}

4. We have the following computations. For ε, the final state equals {a,b,c,e} which is accepting since b is an accepting state.

Input Symbol Read	**Current Subset State**
0	$\{a, b, c, e\}$
1	{c,f}
0	{d,e}
1	{d,f}
Rejecting State:	{c,e}

Input Symbol Read	Current Subset State
0	$\{a, b, c, e\}$
0	{c,f}
0	{c,f}
Accepting State:	{c,f}

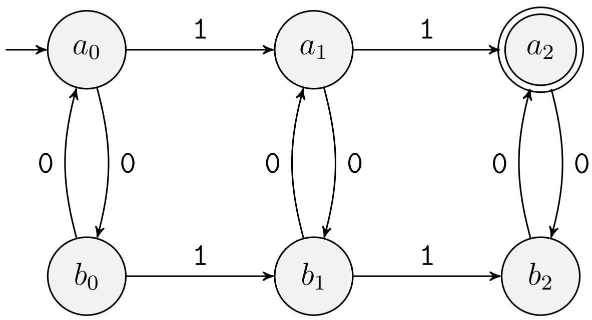
5. The following tables provide the $\delta(Q, \Sigma)$ transition functions for the desired NFA's.

   (a) This the NFA from Example 1.1.

   (b) Initial: a, $F = \{e\}$

$Q\backslash\Sigma$	0	1
a	{b}	{a}
b	{b}	$\{c\}$
c	{d}	{a}
d	$\{b\}$	{e}
e	$\{e\}$	$\{e\}$

   (c) We have the following NFA.



   (d) Initial: a, $F = \{b\}$

$Q\backslash\Sigma$	0	1
a	{b}	$\emptyset$
b	$\emptyset$	$\emptyset$

   (e) Initial: a, $F = \{a, b, c\}$.

$Q\backslash\Sigma$	0	1
a	{a}	{b}
b	{c}	{b}
c	{c}	$\emptyset$

   (f) Initial: a, $F = \{a, c\}$

$Q\backslash\Sigma$	0	1
a	{b}	{a}
b	{c}	$\emptyset$
c	{b}	{c}

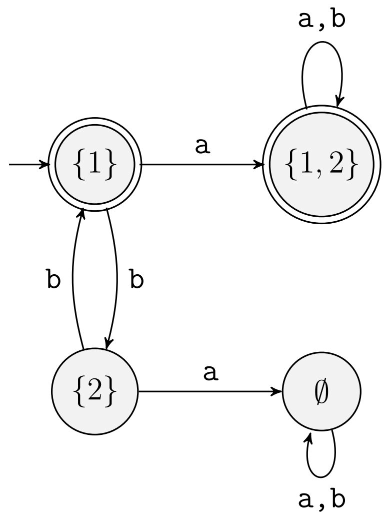
   (g) Initial: a, $F = \{a\}$

$Q\backslash\Sigma$	0	1
a	$\emptyset$	$\emptyset$

   (h) Initial: a, $F = \{a\}$

$Q\backslash\Sigma$	0	1
a	$\{a\}$	$\emptyset$

6. We have the following DFA.



7. We have the following DFA. Notice that the initial state is $\{1, 2\}$, since, becuase of the ε edge, (initially) entering state 1 triggers the entering of state 2.