Tutorial 10

COMP 5361: Discrete Structures and Formal Languages

Mohammad Reza Davari

Concordia University

Outline

Contents of the section

Finite Automata With Epsilon-Transitions

Definition

- We allow for our Finite Automato to transition with empty string i.e ϵ transition. In other words we are allowing the NFA to make a transition without receiving an input symbole.

Finite Automata With Epsilon-Transitions

Definition

- We allow for our Finite Automato to transition with empty string i.e ϵ transition. In other words we are allowing the NFA to make a transition without receiving an input symbole.
- We call NFA with this property ϵ-NFA.

Finite Automata With Epsilon-Transitions

Definition

- We allow for our Finite Automato to transition with empty string i.e ϵ transition. In other words we are allowing the NFA to make a transition without receiving an input symbole.
- We call NFA with this property ϵ-NFA.
- This feature does not make our machinery any more powerful!

Example

For the ϵ-NFA below find the final state(s) the ϵ-NFA will be at, given the following inputs:

1. 010
2. 011

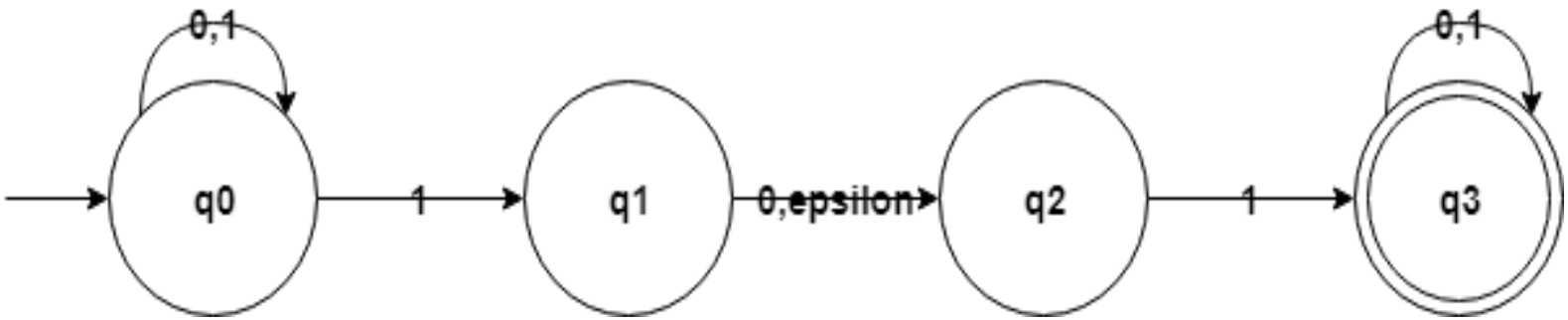

Figure: An example of ϵ-NFA

Solution: First Input

1. 010
2. Upon reading 0 we will end up at $\{q_0\}$

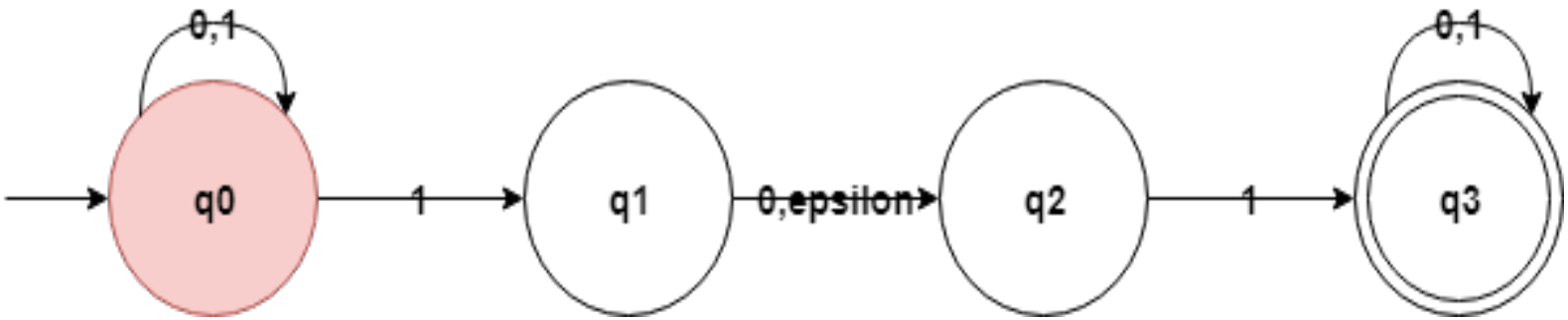

Figure: An example of ϵ-NFA

Solution: First Input

1. 010
2. Upon reading 1 from $\{q_0\}$ we will end up at $\{q_0, q_1\}$.

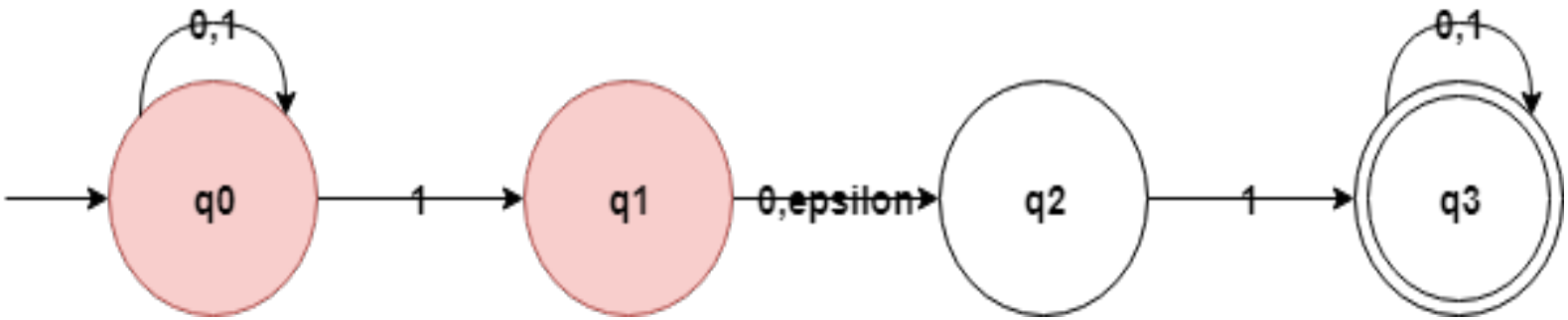

Figure: An example of ϵ-NFA

Solution: First Input

1. 010
2. Upon reading 0 from $\{q_0, q_1\}$ we will end up at $\{q_0, q_2\}$.
3. Note: When we are at q_1 and we are presented with an input we have the option of acting on that input or taking the ϵ instead.

Figure: An example of ϵ-NFA

Solution: Second Input

1. 011
2. Upon reading 0 we will end up at $\{q_0\}$

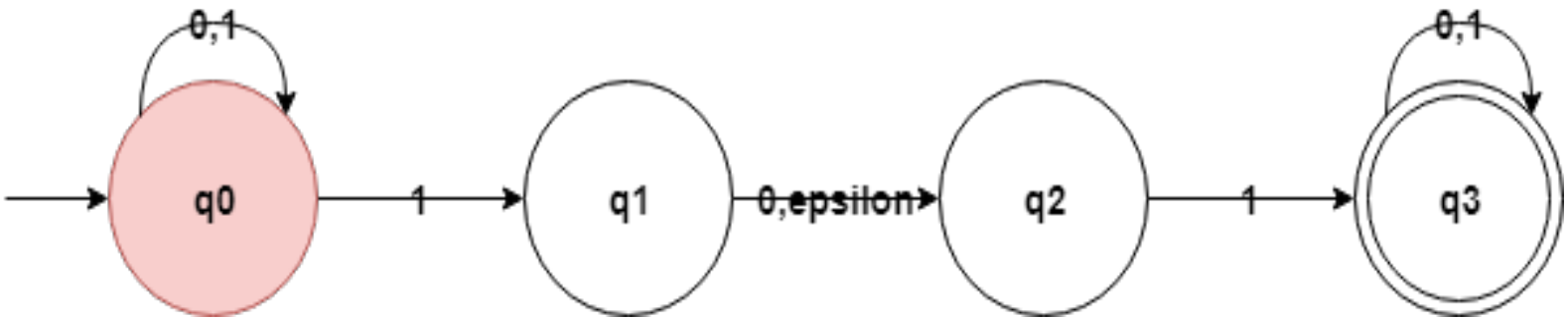

Figure: An example of ϵ-NFA

Solution: Second Input

1. 011
2. Upon reading 1 from $\{q_0\}$ we will end up at $\{q_0, q_1\}$.

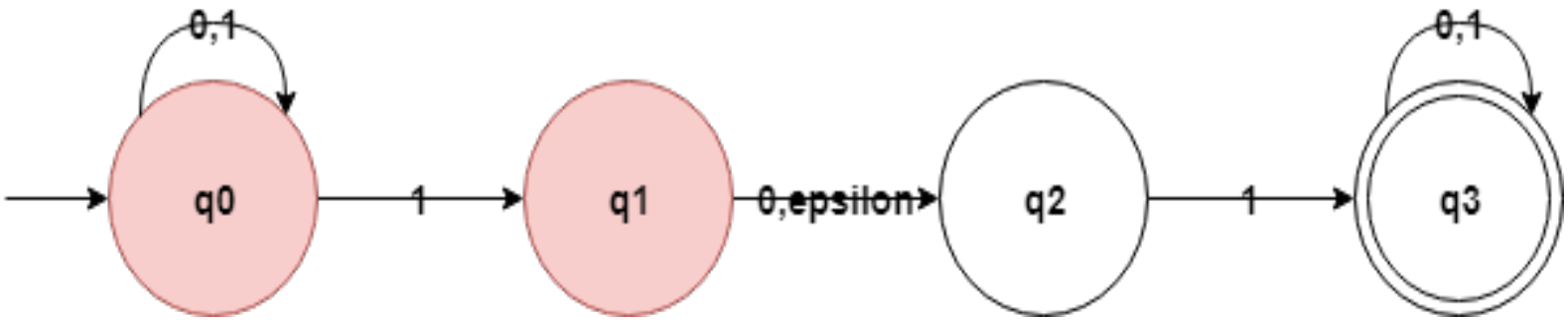

Figure: An example of ϵ-NFA

Solution: Second Input

1. 011
2. Upon reading 1 from $\{q_0, q_1\}$ we will end up at $\{q_0, q_3\}$.
3. Note: When we are at q_1 and we are presented with an input we have the option of acting on that input or taking the ϵ instead.

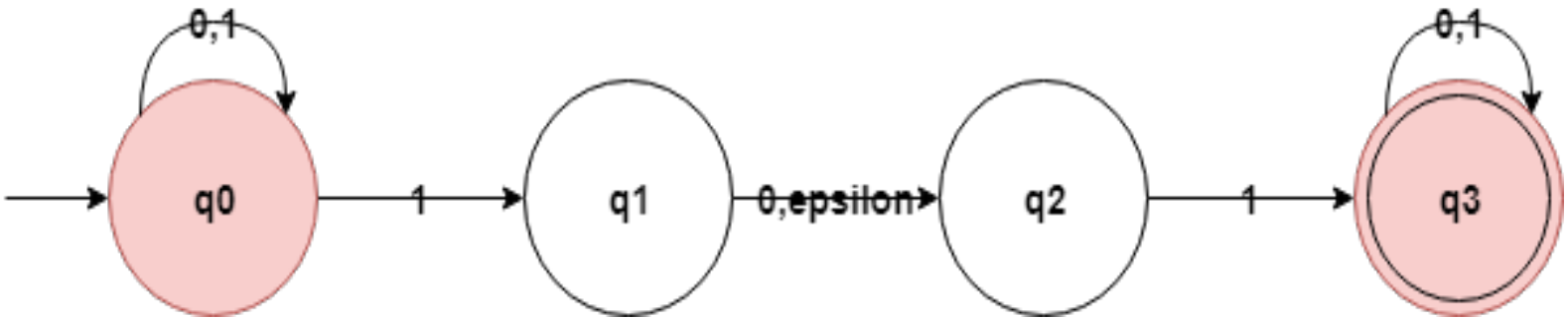

Figure: An example of ϵ-NFA

Formal Definition of ϵ-NFA

The formal definition of ϵ-NFA is exactly like NFA with a small change in the definition of the transition function. Hence ϵ-NFA is a 5-tuple $M = (Q, \sigma, \delta, q_0, F)$ where δ is defined by as a function whose inputs are:

1. A state $q \in Q$.
2. An input symbol in $\Sigma \cup \{\epsilon\}$

Finite Automata With Epsilon-Transitions

ϵ-Closure

The ϵ-Closure of a state q is given by following all transitions out of q that are labeled ϵ. However, when we get to other states by following ϵ, we follow the ϵ-transitions out of those states, and so on, eventually **finding every state that can be reached from q along any path whose arcs are all labeled ϵ.**

Example 1

Find the ϵ-closure of each states of the following ϵ-NFA

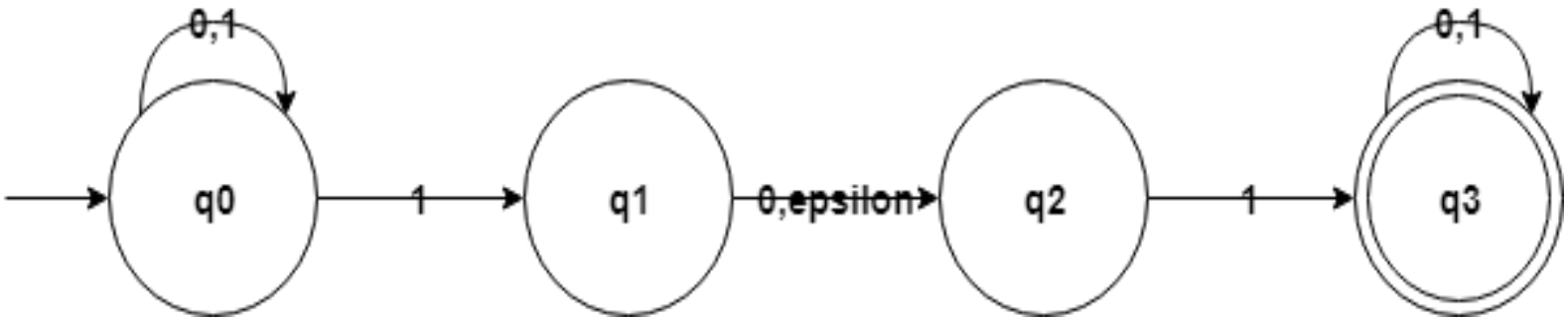

Figure: An example of ϵ-NFA

Finite Automata With Epsilon-Transitions

Solution

- ϵ-closure of q_0:

Finite Automata With Epsilon-Transitions

Solution

- ϵ-closure of q_0: $\emptyset$
- ϵ-closure of q_1:

Finite Automata With Epsilon-Transitions

Solution

- ϵ-closure of q_0: $\emptyset$
- ϵ-closure of q_1: $\{q_2\}$
- ϵ-closure of q_2:

Finite Automata With Epsilon-Transitions

Solution

- ϵ-closure of q_0: $\emptyset$
- ϵ-closure of q_1: $\{q_2\}$
- ϵ-closure of q_2: $\emptyset$
- ϵ-closure of q_3:

Finite Automata With Epsilon-Transitions

Solution

- ϵ-closure of q_0: $\emptyset$
- ϵ-closure of q_1: $\{q_2\}$
- ϵ-closure of q_2: $\emptyset$
- ϵ-closure of q_3: $\emptyset$

Example 2

Find the ϵ-closure of each states of the following ϵ-NFA

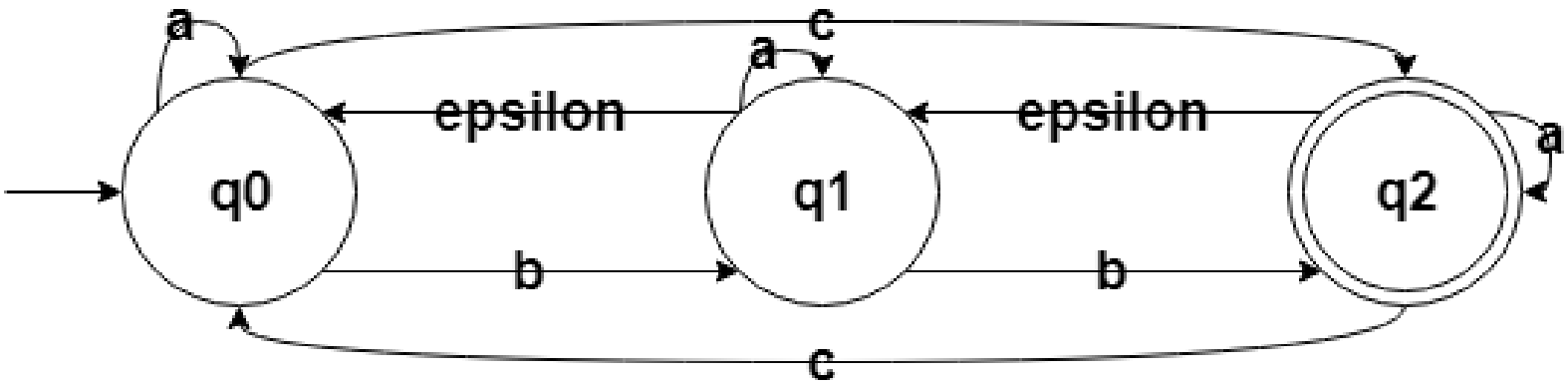

Figure: An example of ϵ-NFA

Finite Automata With Epsilon-Transitions

Solution

- ϵ-closure of q_0:

Finite Automata With Epsilon-Transitions

Solution

- ϵ-closure of q_0: $\emptyset$
- ϵ-closure of q_1:

Finite Automata With Epsilon-Transitions

Solution

- ϵ-closure of q_0: $\emptyset$
- ϵ-closure of q_1: $\{q_0\}$
- ϵ-closure of q_2:

Finite Automata With Epsilon-Transitions

Solution

- ϵ-closure of q_0: $\emptyset$
- ϵ-closure of q_1: $\{q_0\}$
- ϵ-closure of q_2: $\{q_1, q_0\}$

Example 3

Find the ϵ-closure of each states of the following ϵ-NFA

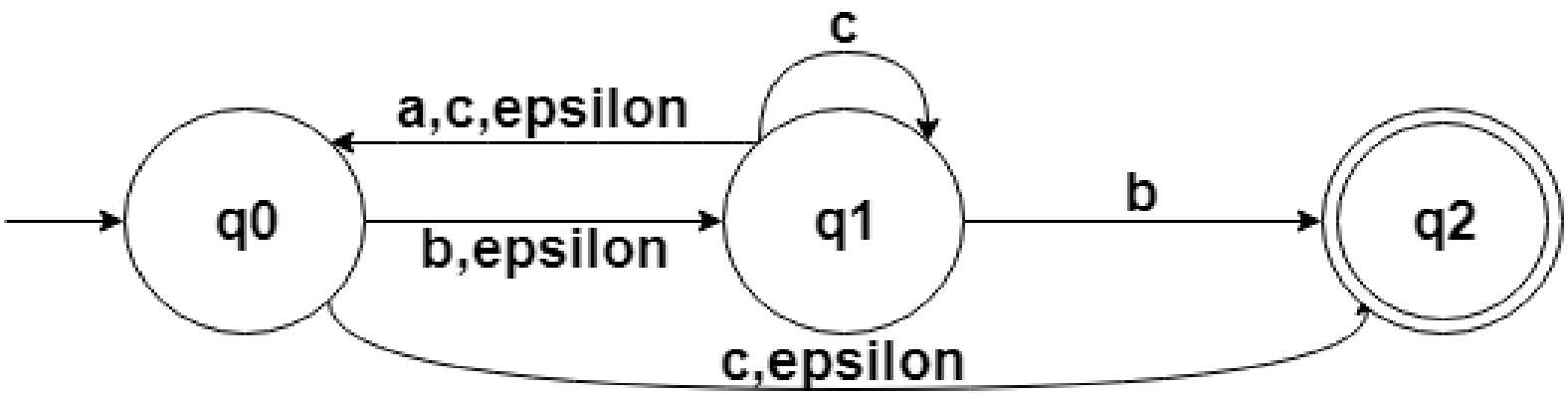

Figure: An example of ϵ-NFA

Finite Automata With Epsilon-Transitions

Solution

- ϵ-closure of q_0:

Finite Automata With Epsilon-Transitions

Solution

- ϵ-closure of q_0: $\{q_1, q_2, q_0\}$
- ϵ-closure of q_1:

Finite Automata With Epsilon-Transitions

Solution

- ϵ-closure of q_0: $\{q_1, q_2, q_0\}$
- ϵ-closure of q_1: $\{q_0, q_1, q_2\}$
- ϵ-closure of q_2:

Finite Automata With Epsilon-Transitions

Solution

- ϵ-closure of q_0: $\{q_1, q_2, q_0\}$
- ϵ-closure of q_1: $\{q_0, q_1, q_2\}$
- ϵ-closure of q_2: $\emptyset$

Language of ϵ-NFA

The language of ϵ-NFA consists of all the strings in Σ^* such that **at least one** of their state expansions lead to **an accept state**. Note that when the input is traveling through the machine, for each state that has ϵ transition, we branch out by considering:

1. What if the machine reads a symbol and ignores the ϵ.
2. What if the machine does not read a symbol and instead takes advantage of ϵ transition.

Finite Automata With Epsilon-Transitions

Theorem

The followings are all equivalent:

1. DFA
2. NFA
3. ϵ-NFA

Finite Automata With Epsilon-Transitions

Theorem

The followings are all equivalent:

1. DFA
2. NFA
3. ϵ-NFA

Note

- By definition a DFA is an NFA and also an ϵ-NFA.
- We have already shown an algorithm for converting NFA to DFA, hence if we show that we can also convert ϵ-NFA to DFA the theorem is proved.

Converting ϵ-NFA to DFA

The algorithm we use for this conversion is exactly the same as for the algorithm we used for the conversion of NFA to DFA. The only difference is in filling the transition table of ϵ-NFA, where we also need to consider the ϵ-closure of each state.

Example 1

Convert the following ϵ-NFA to DFA.

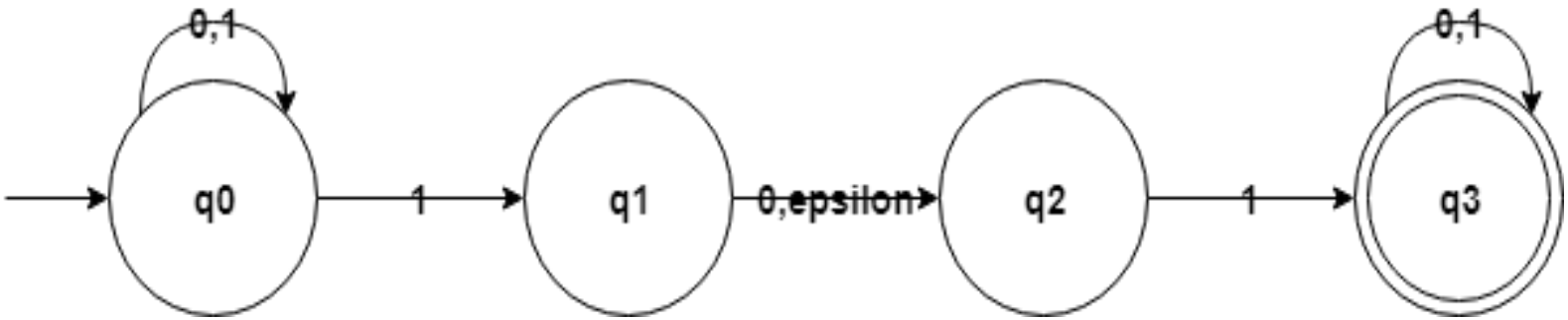

Figure: An example of ϵ-NFA

Solution: Step 1 - Transition Table for ϵ-NFA

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$

Solution: Step 1 - Transition Table for ϵ-NFA

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
q_1	q_2	q_3

Solution: Step 1 - Transition Table for ϵ-NFA

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
q_1	q_2	q_3
q_2	$\emptyset$	q_3

Solution: Step 1 - Transition Table for ϵ-NFA

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
q_1	q_2	q_3
q_2	$\emptyset$	q_3
$\star\ q_3$	q_3	q_3

Solution: Step 2 - Transition Table for DFA

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$

Solution: Step 2 - Transition Table for DFA

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_3\}$

Solution: Step 2 - Transition Table for DFA

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_2\}$	q_0	$\{q_0, q_1, q_3\}$

Solution: Step 2 - Transition Table for DFA

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_2\}$	q_0	$\{q_0, q_1, q_3\}$
$\star \{q_0, q_1, q_3\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_3\}$

Solution: Step 2 - Transition Table for DFA

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_2\}$	q_0	$\{q_0, q_1, q_3\}$
$\star \{q_0, q_1, q_3\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$\star \{q_0, q_2, q_3\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_3\}$

Solution: Step 2 - Transition Table for DFA

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_2\}$	q_0	$\{q_0, q_1, q_3\}$
$\star \{q_0, q_1, q_3\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$\star \{q_0, q_2, q_3\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_3\}$
$\star \{q_0, q_3\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_3\}$

Solution: Step 3 - Transition Diagram for DFA

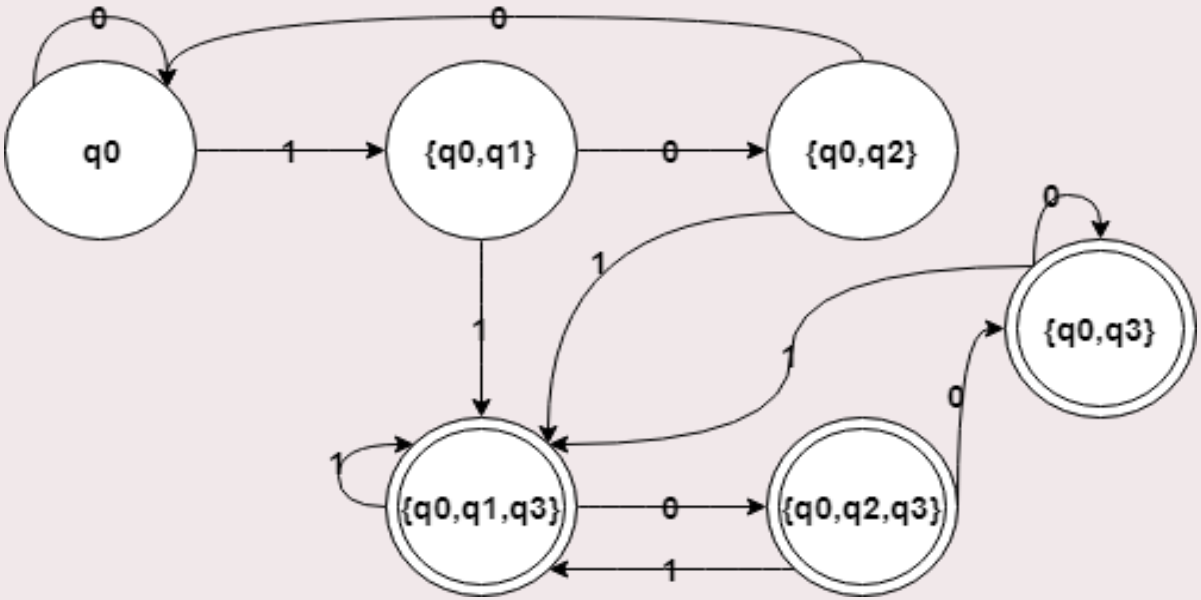

Figure: DFA transition diagram.

Example 2

Convert the following ϵ-NFA to DFA.

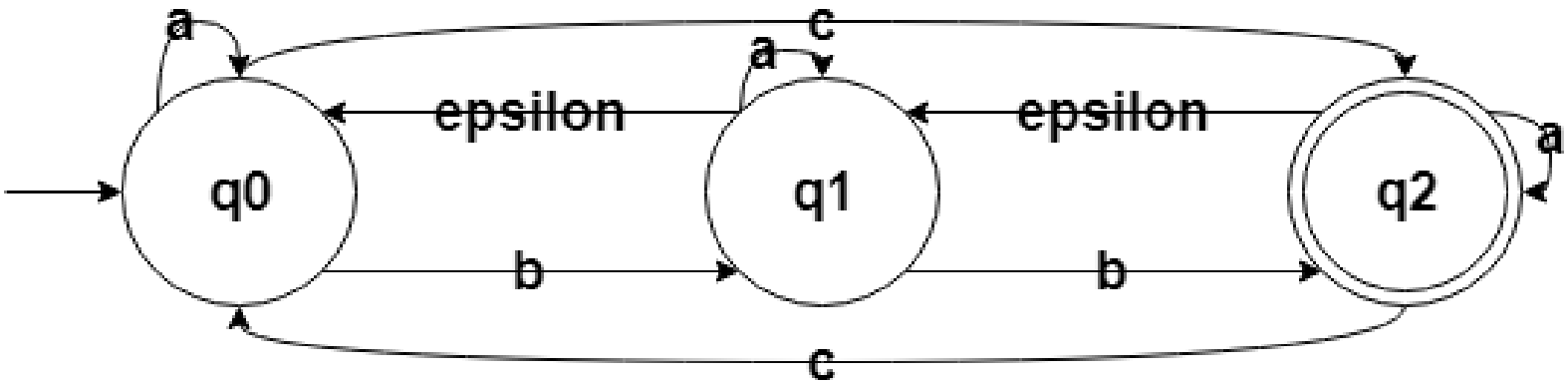

Figure: An example of ϵ-NFA

Solution: Step 1 - Transition Table for ϵ-NFA

δ	a	b	c
$\rightarrow q_0$	q_0	$\{q_1, q_0\}$	$\{q_2, q_1, q_0\}$

Solution: Step 1 - Transition Table for ϵ-NFA

δ	a	b	c
$\rightarrow q_0$	q_0	$\{q_1, q_0\}$	$\{q_2, q_1, q_0\}$
q_1	$\{q_1, q_0\}$	$\{q_2, q_1, q_0\}$	$\{q_2, q_1, q_0\}$

Solution: Step 1 - Transition Table for ϵ-NFA

δ	a	b	c
$\rightarrow q_0$	q_0	$\{q_1, q_0\}$	$\{q_2, q_1, q_0\}$
q_1	$\{q_1, q_0\}$	$\{q_2, q_1, q_0\}$	$\{q_2, q_1, q_0\}$
$\star q_2$	$\{q_2, q_1, q_0\}$	$\{q_2, q_1\}$	$\{q_0, q_2\}$

Solution: Step 2 - Transition Table for DFA

δ	a	b	c
$\rightarrow q_0$	q_0	$\{q_1, q_0\}$	$\{q_2, q_1, q_0\}$

Solution: Step 2 - Transition Table for DFA

δ	a	b	c
$\rightarrow q_0$	q_0	$\{q_1, q_0\}$	$\{q_2, q_1, q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$

Solution: Step 2 - Transition Table for DFA

δ	a	b	c
$\rightarrow q_0$	q_0	$\{q_1, q_0\}$	$\{q_2, q_1, q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$
$\star \{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$

Solution: Step 3 - Transition Diagram for DFA

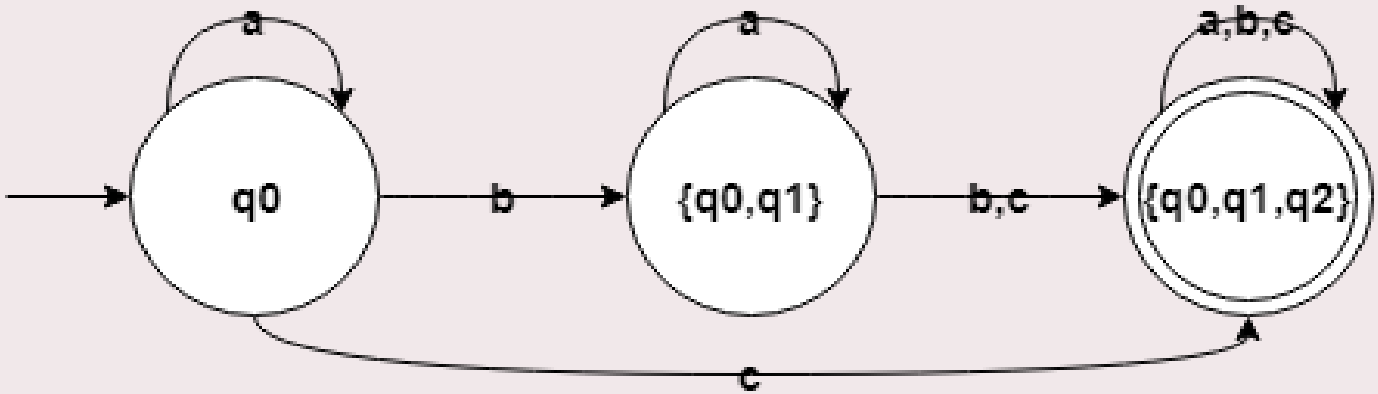

Figure: DFA transition diagram.

Example 3

Convert the following ϵ-NFA to DFA.

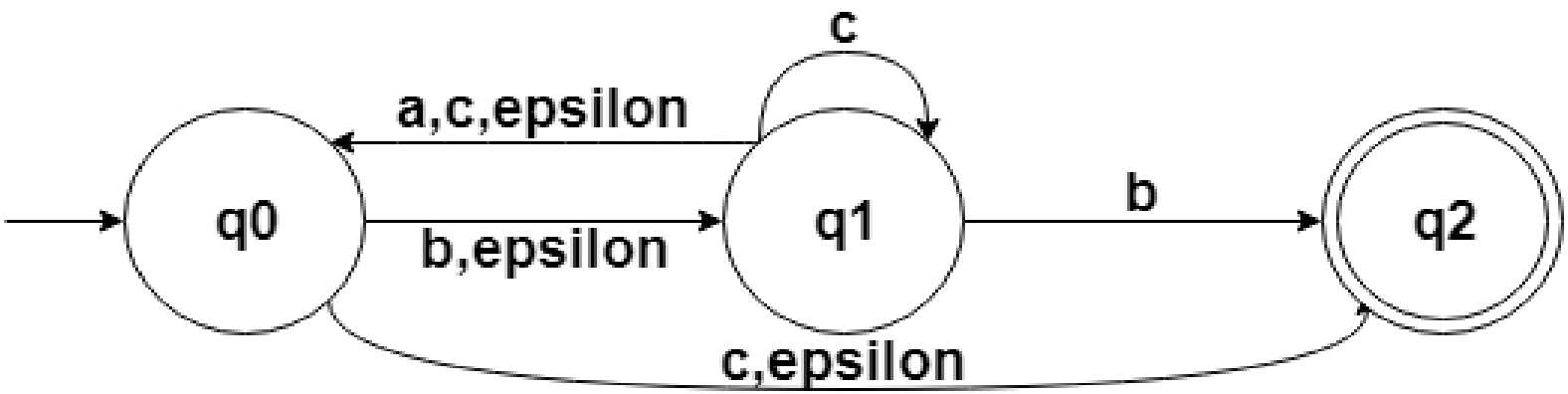

Figure: An example of ϵ-NFA

Solution: Step 1 - Transition Table for ϵ-NFA

δ	a	b	c
$\rightarrow q_0$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$

Solution: Step 1 - Transition Table for ϵ-NFA

δ	a	b	c
$\rightarrow q_0$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$
q_1	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$

Solution: Step 1 - Transition Table for ϵ-NFA

δ	a	b	c
$\rightarrow q_0$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$
q_1	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$
$\star\ q_2$	$\emptyset$	$\emptyset$	$\emptyset$

Solution: Step 2 - Transition Table for DFA

δ	a	b	c
$\rightarrow q_0$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$

Solution: Step 2 - Transition Table for DFA

δ	a	b	c
$\rightarrow q_0$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$
$\star \{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$

Solution: Step 3 - Transition Diagram for DFA

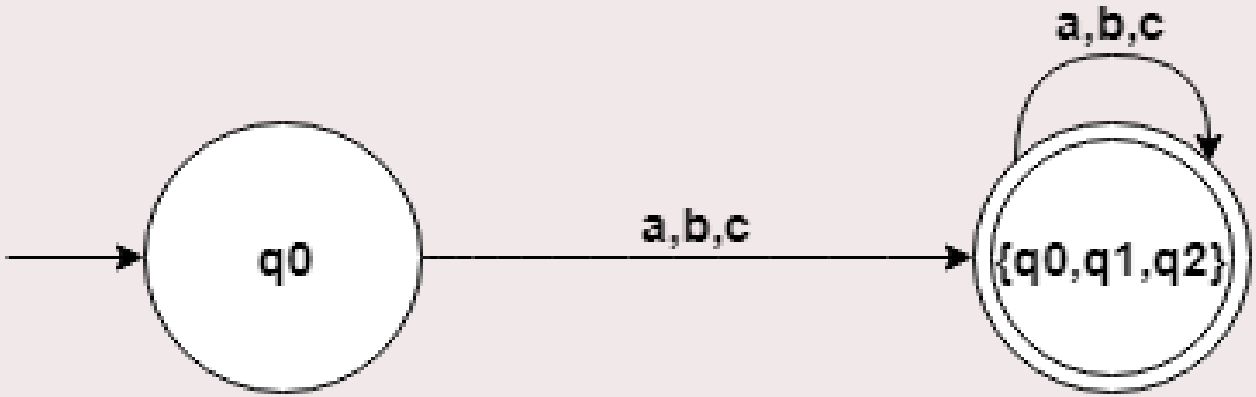

Figure: DFA transition diagram.

Contents of the section

Regular Expressions

Introduction

- We have worked so far with Finite Automata to represent regular languages (a regular language is the language accepted by a FA).

Regular Expressions

Introduction

- We have worked so far with Finite Automata to represent regular languages (a regular language is the language accepted by a FA).
- Regular Expressions are the algebraic way of denoting regular languages.

Formal Definition of Regular Expressions

We describe the regular expressions recursively as follows:

- **Basis**:
  1. The constants ϵ and $\emptyset$ are regular expressions, denoting the languages $\{\epsilon\}$ and $\emptyset$ respectively.
  2. If a is any symbol then a is a regular expression.
  3. A variable, usually capitalized and italic such as L is a variable, representing any language.

Formal Definition of Regular Expressions

- **Induction**:
  1. If E and F are regular expressions, then $E + F$ is a regular expression denoting the union ($\cup$) of $L(E)$ and $L(F)$.
  2. If E and F are regular expressions,then EF is a regular expression denoting the concatenation of $L(E)$ and $L(F)$.
  3. If E is a regular expression, then E^* is a regular expression, denoting the Kleene closure of $L(E)$.
  4. If E is a regular expression, then (E) a parenthesized E, is also a regular expression denoting the same language as E.

Regular Expressions

Example

Express the following languages using regular expressions.

1. L is the language that consists of alternating 0's and 1's.
2. L is the language that consists of either 1's or 0's in multiples of 3.
3. L is the language consists of words of length 5 using English alphabet that start with ax.

Regular Expressions

Example

Express the following languages using regular expressions.

1. L is the language that consists of alternating 0's and 1's.
2. L is the language that consists of either 1's or 0's in multiples of 3.
3. L is the language consists of words of length 5 using English alphabet that start with ax.

Solution

1. $L = (01)^*$
2. $L = (000)^* + (111)^*$
3. $L = ax(\{\epsilon\} + \Sigma + \Sigma^2 + \Sigma^3)$ where Σ is English alphabet.

Regular Expressions: Order of Operation

1. The star operator has the highest priority! That is it applies only to the smallest sequence of symbols to its left that is a well formed regular expression.
2. Concatenation has the second highest order of operation.
3. Union $(+)$ has the lowest order of operation.

Contents of the section

Reminder

We have already shown that the followings are equivalent and we covered an algorithm for their conversion.

- DFA
- NFA
- ϵ-NFA

Now we want to add one more item to the list: **Regular Expressions**.

Reminder

We have already shown that the followings are equivalent and we covered an algorithm for their conversion.

- DFA
- NFA
- ϵ-NFA

Now we want to add one more item to the list: **Regular Expressions**.
Next we will cover the conversion algorithms.

Finite Automata to Regular Expression

Our conversion starts with first converting the FA to a Generalized NFA (GNFA). GNFA has the following characteristics/conditions:

1. The start state has transition arrows going to every other state but **no arrows** coming in from any other states.
2. There is only a **single accept** state, and it has arrows coming in from every other state but no arrows going to any other state. Furthermore, the accept state is not the same as the start state.
3. Except for the start and accept states, one arrows go from every state to every other state and also from each state to itself.

GNFA Construction

- If condition 1 is not already satisfied then create a new start state and connect it to the old start state with ϵ connection.
- If condition 2 is not satisfied then create a new accept state and connect the old accept state(s) by ϵ connection to it. Turn the old accept states to non-accept states.
- If condition 3 is not satisfied for some state q create *dummy* connections with $\emptyset$ transitions.

Example

Turn the following NFA to a GNFA.

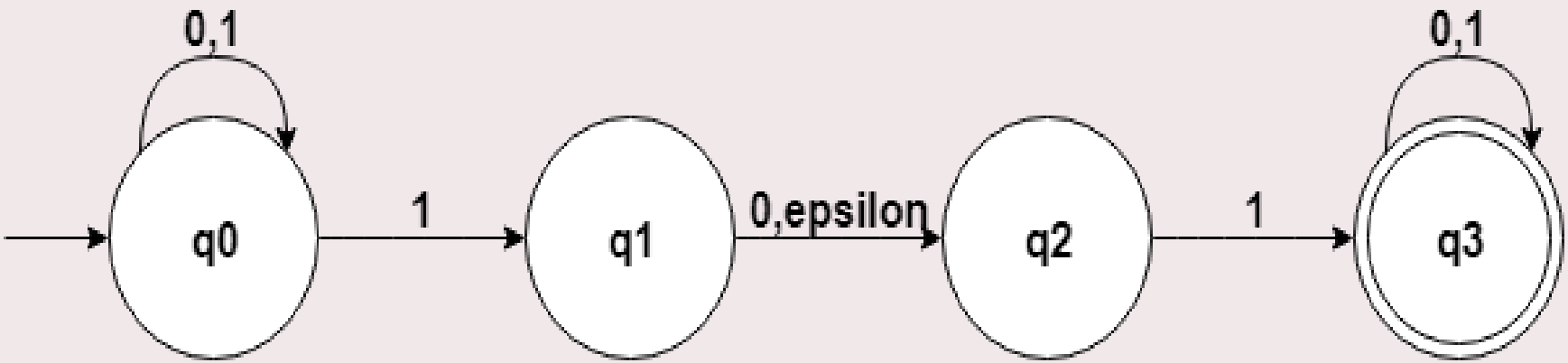

Figure: NFA diagram.

Solution: Part 1

Condition 1 is not met, we fix it as instructed and the result will be:

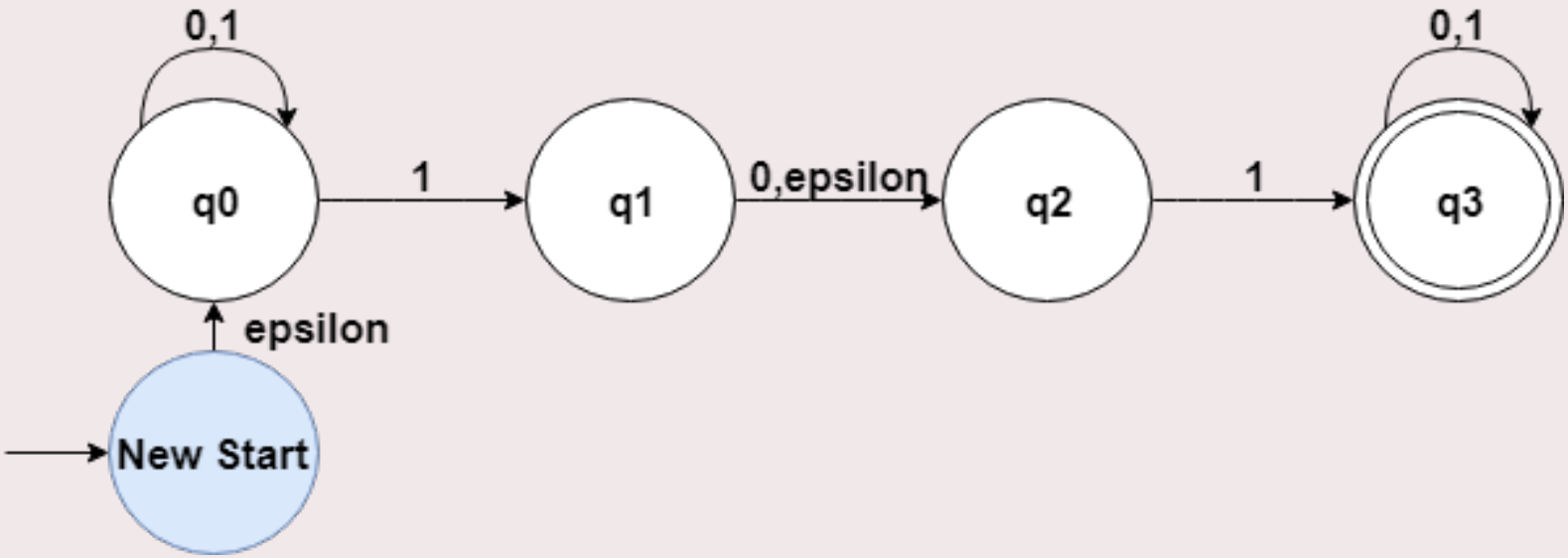

Figure: NFA diagram.

Solution: Part 2

Condition 2 is not met, we fix it as instructed and the result will be:

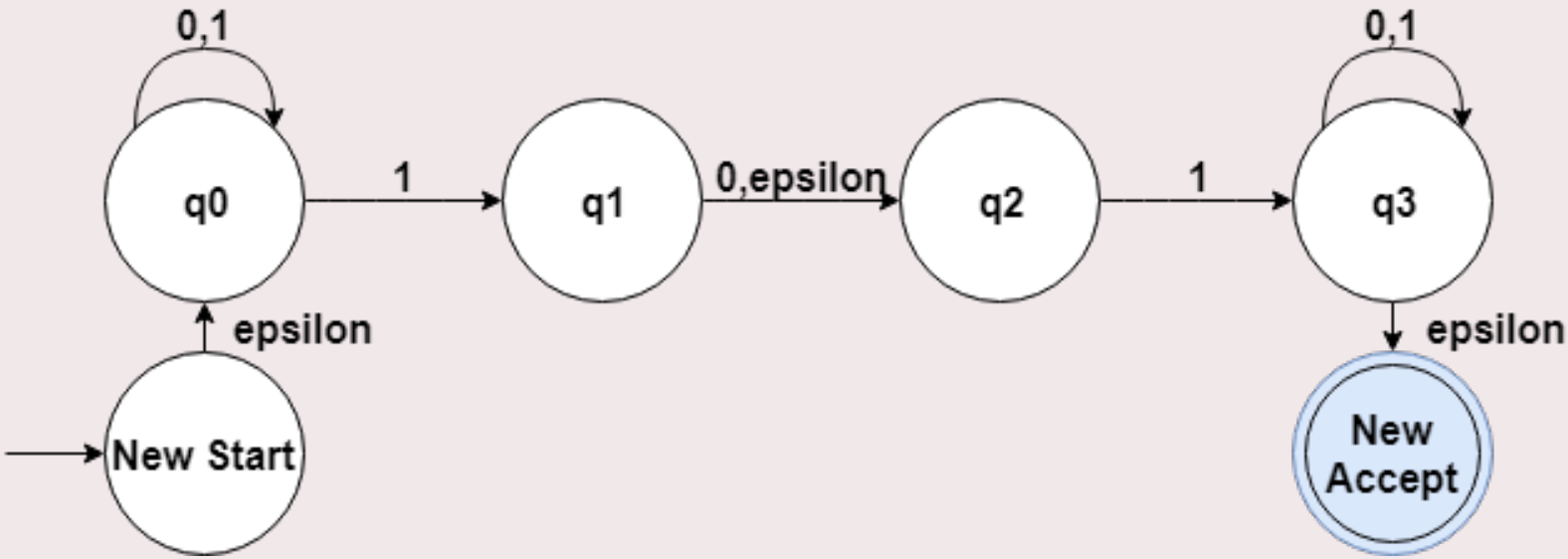

Figure: NFA diagram.

Solution: Part 3

Condition 3 is not met, we fix it as instructed and the result will be the following: (Please not that all red arrows are $\emptyset$ transitions)

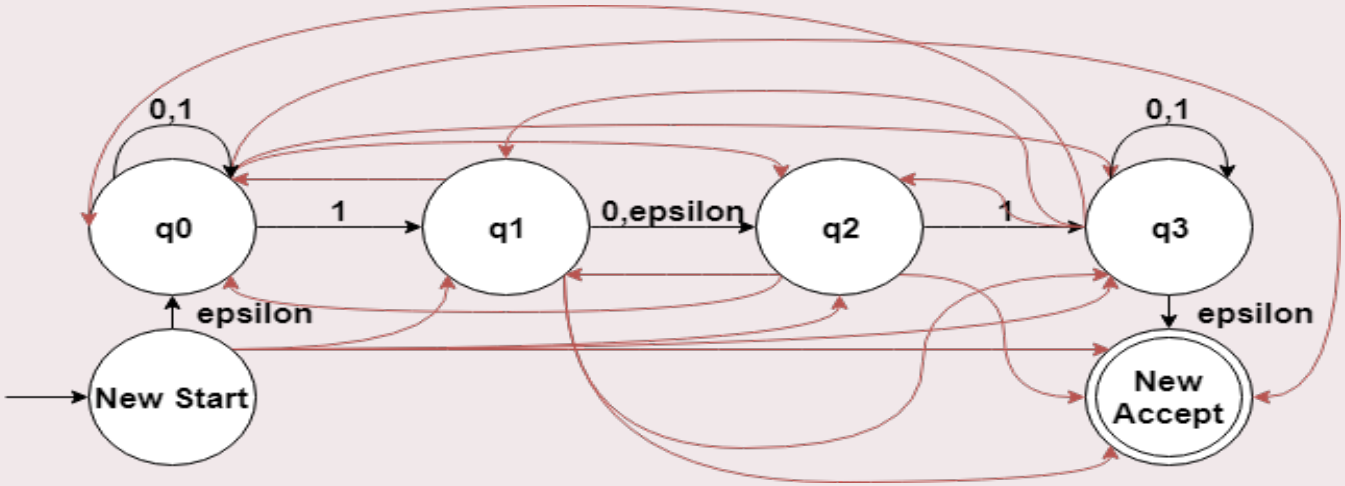

Figure: NFA diagram.

Finite Automata to Regular Expression

Once we obtained the GNFA form the FA we perform the following to obtain the regular expression.

- Let k be the number of states of GNFA.
- Replace all "," with $+$.
- If $k = 2$ then the GNFA consists of a start state, an accept state, and a single arrow connecting them and labeled with regular expression R. Return R.

Finite Automata to Regular Expression

- If $k > 2$ we select any state $q_{del} \in Q$ different from q_{start} and q_{accept} and delete it while adding the following transitions:

$$\delta(q_i, q_j) = (R_1)(R_2)^*(R_3) + (R_4)$$

for $R_1 = \delta(q_i, q_{del})$, $R_2 = \delta(q_{del}, q_{del})$, $R_3 = \delta(q_{del}, q_j)$ and $R_4 = \delta(q_i, q_j)$.

- Repeat the above until you are done.

Example

Find the regular expression of the following GNFA over $\Sigma = \{a, b\}$:

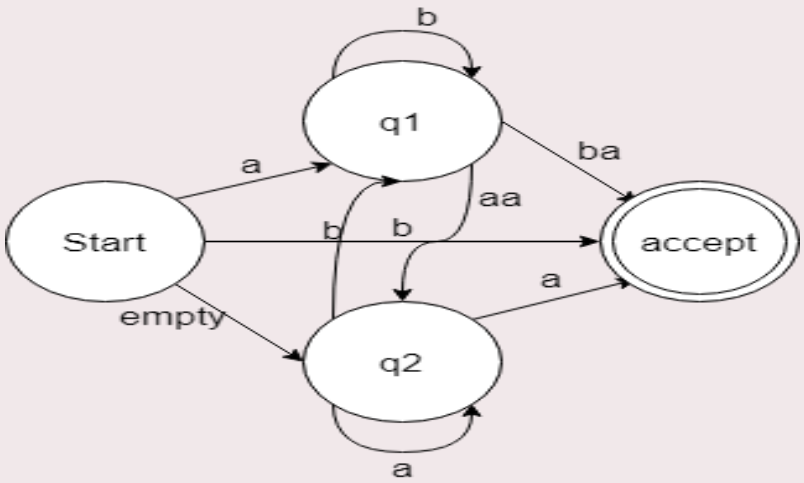

Figure: GNFA to RegEx.

Solution: Part 1

In this step we will remove q_1 and perform the algorithm to obtain the new transitions.

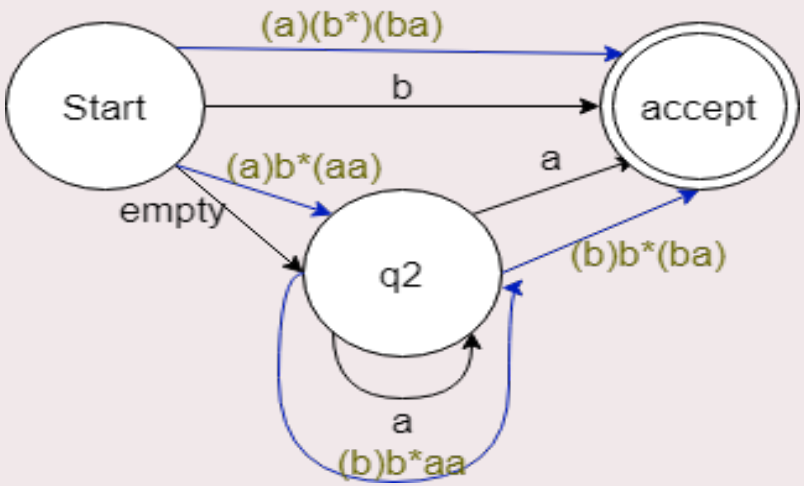

Figure: GNFA to RegEx.

Solution: Part 2

In this step we will just simplify the previous diagram and we have:

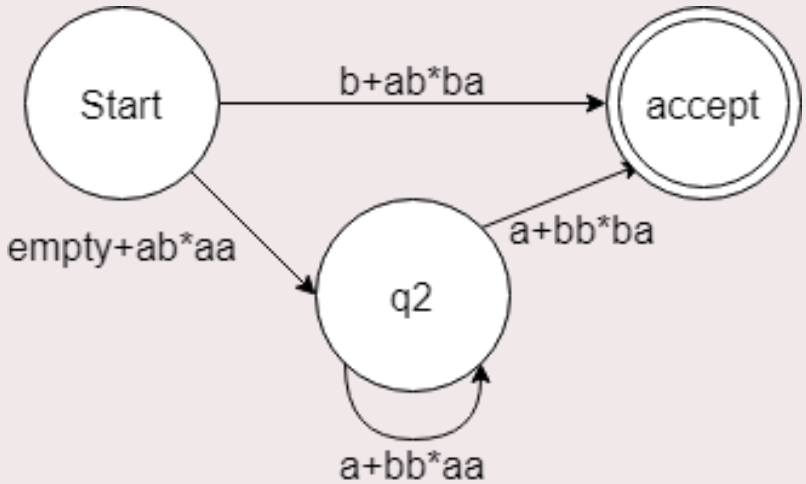

Figure: GNFA to RegEx.

Solution: Part 3

In this step we will remove q_2 and perform the algorithm to obtain the new transitions.

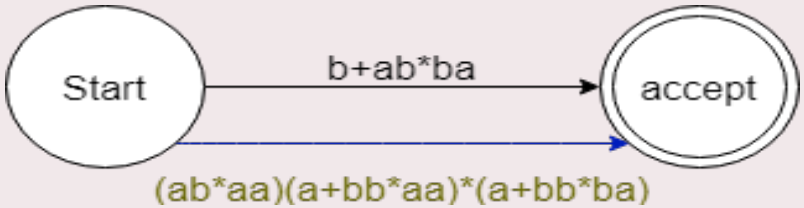

Figure: GNFA to RegEx.

Solution: Part 4

In this step we will just simplify the previous diagram and we have:

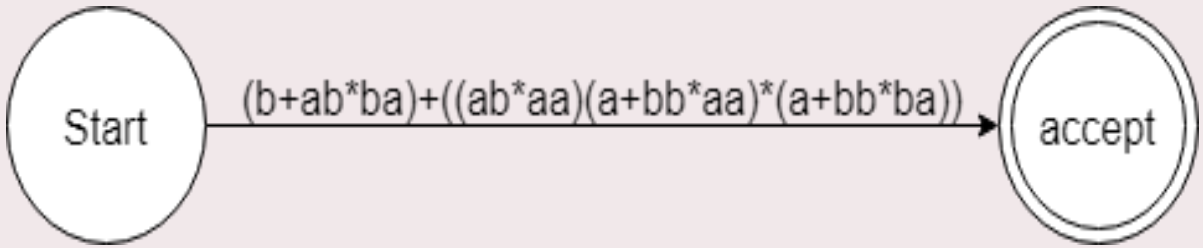

Figure: GNFA to RegEx.

Note that our algorithm is finished at this point and the regular expression is given on the transition above.

Regular Expression to ϵ-NFA

The idea here is to perform the reverse of what we had for the previous conversion. We will introduce a few *building blocks* that you will use a combination of to *translate* your regular expression into an ϵ-NFA.

Building Block: Part 1

The following Automata is our block for ϵ.

Figure: Building block.

Building Block: Part 2

The following Automata is our block for $\emptyset$.

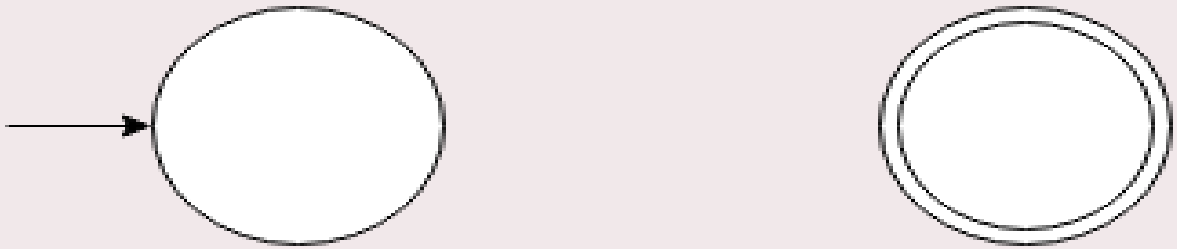

Figure: Building block.

Building Block: Part 3

The following Automata is our block for any symbol a in Σ.

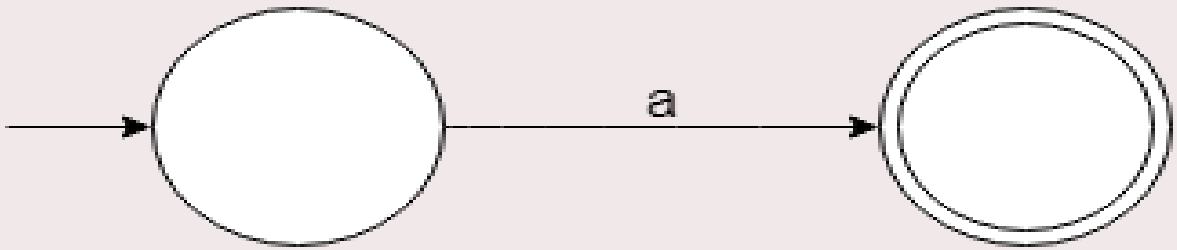

Figure: Building block.

Building Block: Part 4

For any 2 regular expressions R and E, we build $R + E$ as follows:

1. We first create a new start state and connect it to the start state of R and E with ϵ connections.
2. We then create a new accept state and connect the accept state(s) of R and E to it with ϵ connections.

Building Block: Part 4

Below we will demonstrate a pictorial representation of the block mentioned in the previous slide.

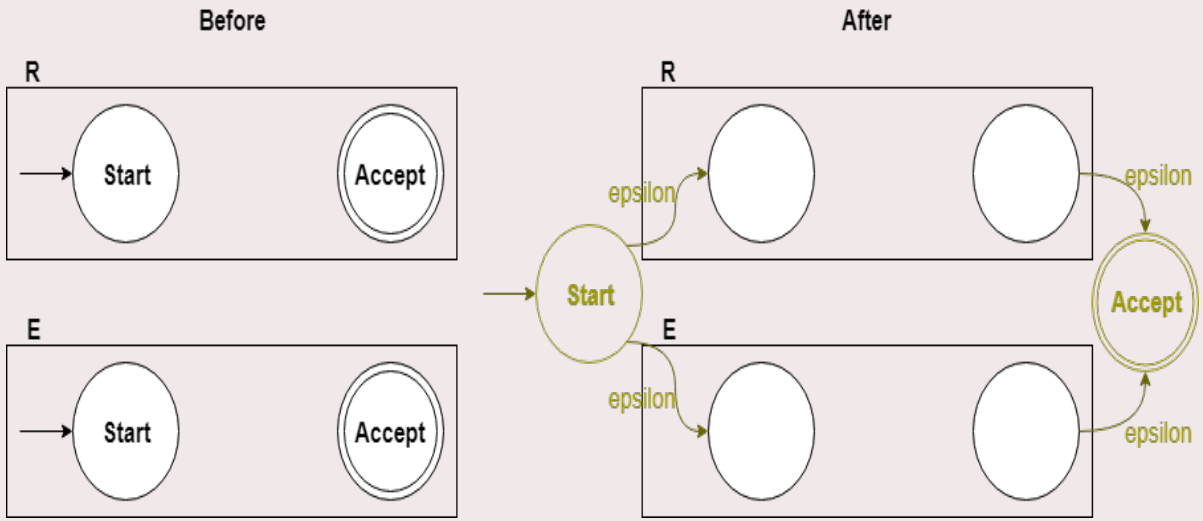

Figure: Building block.

Building Block: Part 5

For any 2 regular expressions R and E, we build RE as follows:

1. We first use ϵ connections to connect the accept state(s) of R to the start state of E.
2. We then turn the accept state(s) of R to non-accept state(s).

Building Block: Part 5

Below we will demonstrate a pictorial representation of the block mentioned in the previous slide.

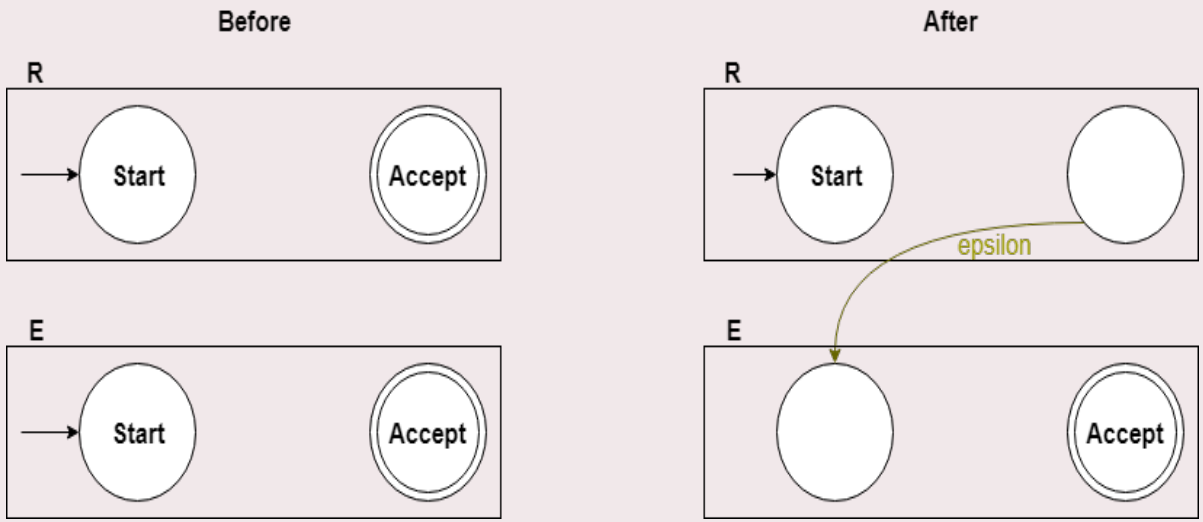

Figure: Building block.

Building Block: Part 6

For any regular expressions R, we build R^* as follows:

1. We use ϵ connections to connect the accept state(s) of R to the start state of R and vice versa.

Building Block: Part 6

Below we will demonstrate a pictorial representation of the block mentioned in the previous slide.

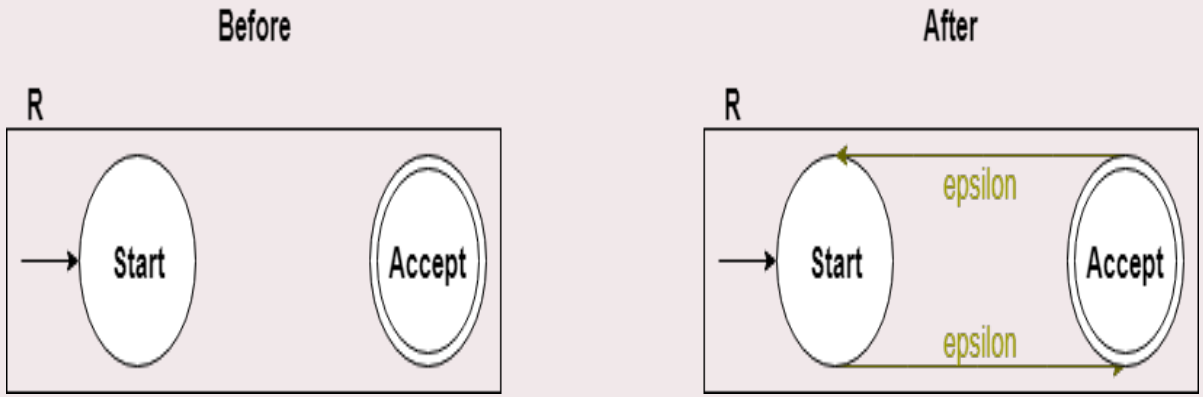

Figure: Building block.

Example

Build an NFA for the following regular expression:

$$R = (00 + 11)^*1$$