Weighted Finite State Transducers

CSE 447 / 517

February 10, 2022 (Week 6)

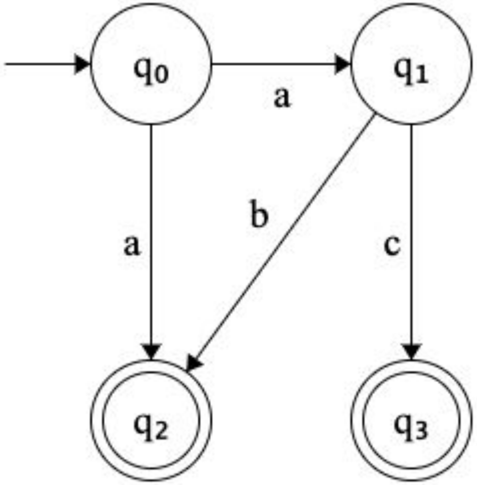
Agenda

- Finite State Automata
- Weighted Finite-State Transducer
- Quiz 5 Solutions
- Q & A

Finite State Automata

Defined by:

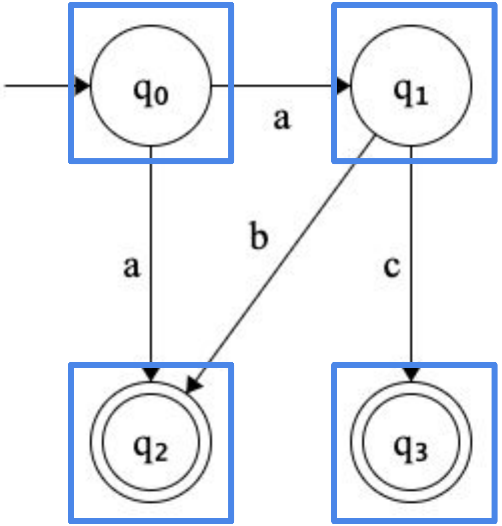
- a finite set of states, Q
  - a start state, $q_0 \in Q$
  - a set of final states, $F \subseteq Q$
- a finite alphabet of input symbols, Σ
- a transition function that maps a state and a symbol (or an empty string, denoted ε) to a set of states, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$



Finite State Automata

Defined by:

- **a finite set of states, Q**
  - a start state, $q_0 \in Q$
  - a set of final states, $F \subseteq Q$
- a finite alphabet of input symbols, Σ
- a transition function that maps a state and a symbol (or an empty string, denoted ε) to a set of states, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$



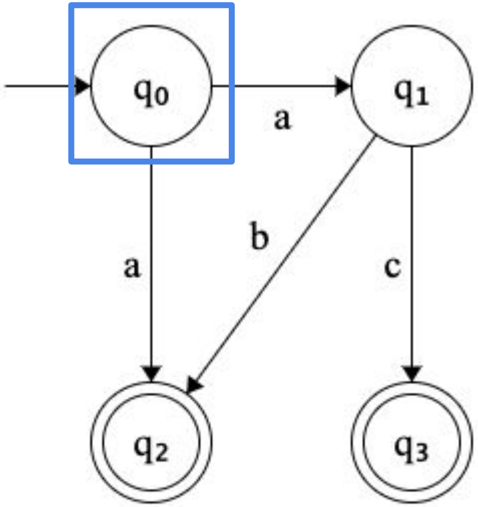
In this example: **$\{q_0, q_1, q_2, q_3\}$**

Finite State Automata

Defined by:

- a finite set of states, Q
  - **a start state, $q_0 \in Q$**
  - a set of final states, $F \subseteq Q$
- a finite alphabet of input symbols, Σ
- a transition function that maps a state and a symbol (or an empty string, denoted ε) to a set of states, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$

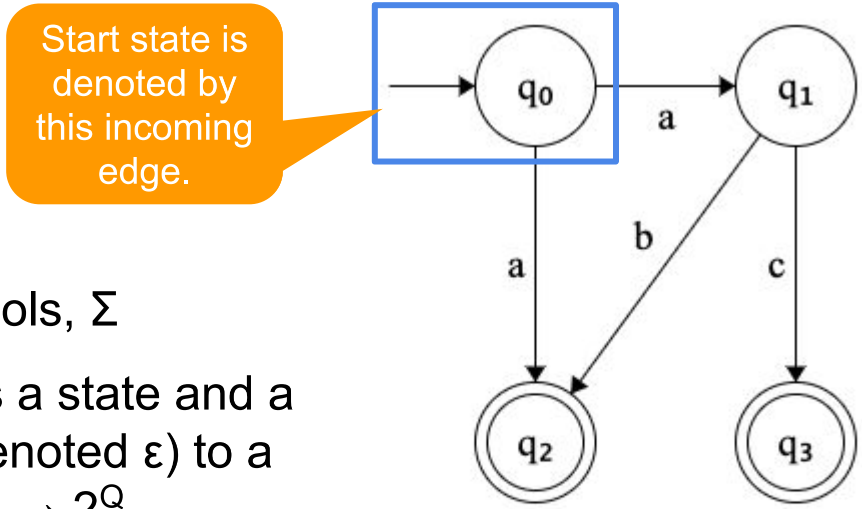
In this example: **q_0**



Finite State Automata

Defined by:

- a finite set of states, Q
  - **a start state, $q_0 \in Q$**
  - a set of final states, $F \subseteq Q$
- a finite alphabet of input symbols, Σ
- a transition function that maps a state and a symbol (or an empty string, denoted ε) to a set of states, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$

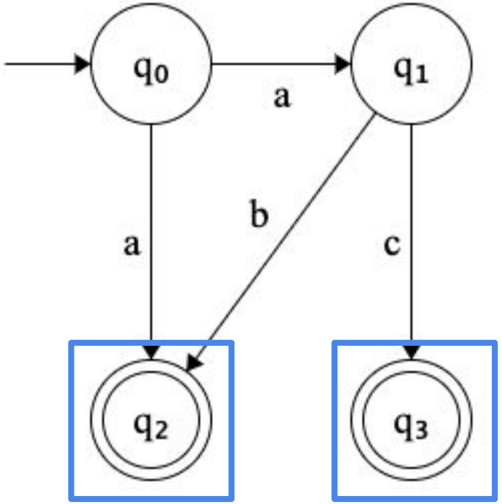


In this example: **q_0**

Finite State Automata

Defined by:

- a finite set of states, Q
  - a start state, $q_0 \in Q$
  - **a set of final states, $F \subseteq Q$**
- a finite alphabet of input symbols, Σ
- a transition function that maps a state and a symbol (or an empty string, denoted ε) to a set of states, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$



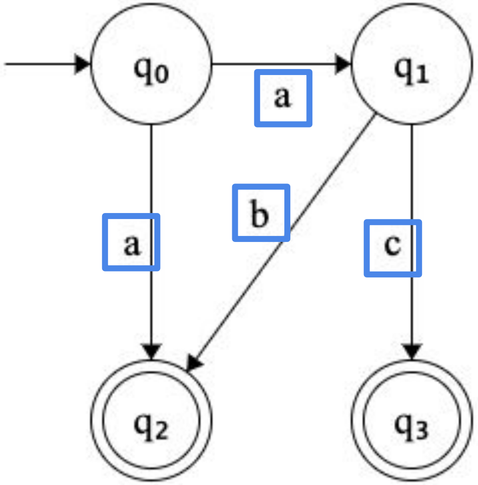
In this example: **$\{q_2, q_3\}$**

Finite State Automata

Defined by:

- a finite set of states, Q
    - a start state, $q_0 \in Q$
    - a set of final states, $F \subseteq Q$
- **a finite alphabet of input symbols, Σ**
- a transition function that maps a state and a symbol (or an empty string, denoted ε) to a set of states, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$

In this example: **{a, b, c}**



Finite State Automata

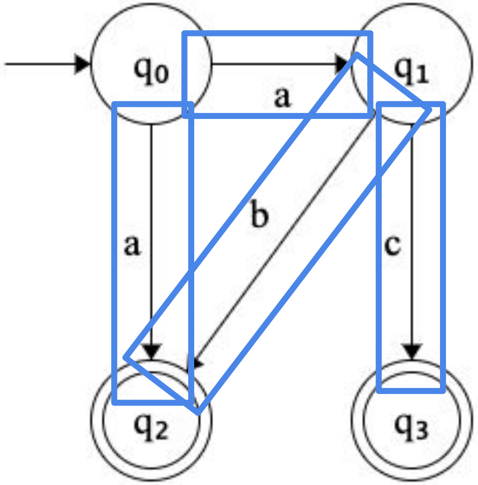
Defined by:

- a finite set of states, Q
  - a start state, $q_0 \in Q$
  - a set of final states, $F \subseteq Q$
- a finite alphabet of input symbols, Σ
- **a transition function that maps a state and a symbol (or an empty string, denoted ε) to a set of states, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$**

In this example: **{**

$(q_0, a) \rightarrow \{q_1, q_2\}, (q_0, b) \rightarrow \varnothing$, ...

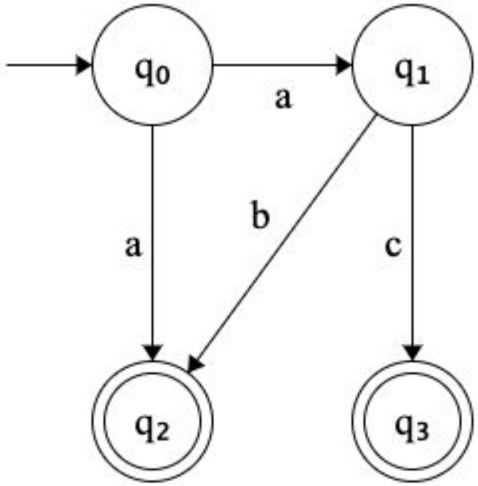
}



Finite State Automata

An FSA, F, defines a *language*, L(F), by accepting the strings that belong to the language, and reject strings that do not.

What is accepting a string?

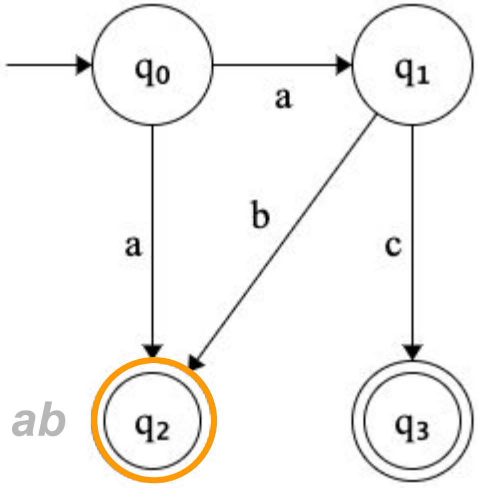


Finite State Automata

An FSA, F, defines a *language*, L(F), by accepting the strings that belong to the language, and reject strings that do not.

What is accepting a string?

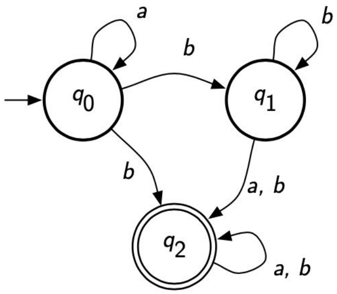
As long as we could have (1) landed a final state (2) after we consume our entire input, then the FSA accept the string!



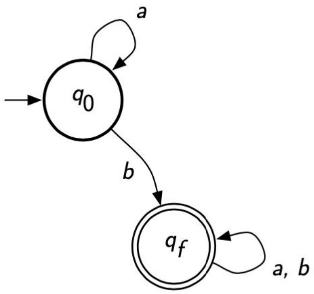
Finite State Automata

Deterministic v.s. Non-deterministic FSA:

- An FSA is deterministic (a “deterministic finite automata”) if there is exactly one path per string in L(F).



NFA



DFA

Finite State Automata

Deterministic v.s. Non-deterministic FSA:

- An FSA is deterministic (a “deterministic finite automata”) if there is exactly one path per string in L(F).

- Any NFA can be mechanically transformed into a DFA one with the same language, but the number of states may explode.

Finite State Automata

Weighted FSA:

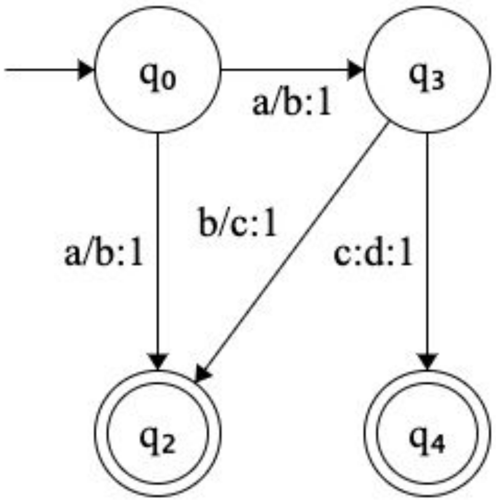
- Associate each transition (edge) with a weight
- Associate the start state with a weight
- Associate each final state with a weight
- To score a path:

$$\lambda(q_0) + \left(\sum_{i=1}^{n} \delta(q_{i-1}, x_i, q_i) \right) + \rho(q_n)$$

Weighted Finite State Transducer

Defined by:

- a finite set of states, Q
  - a start state, $q_0 \in Q$
  - a set of final states, $F \subseteq Q$
- a finite alphabet of input symbols, Σ
- a finite alphabet of output symbols, Ω
- a transition function that maps a state pair and a pair of symbols (or ε) to weights, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Omega \cup \{\varepsilon\}) \times Q \rightarrow R$
- an initial weight function, $\lambda: Q \rightarrow R$
- a final weight function, $\rho: Q \rightarrow R$



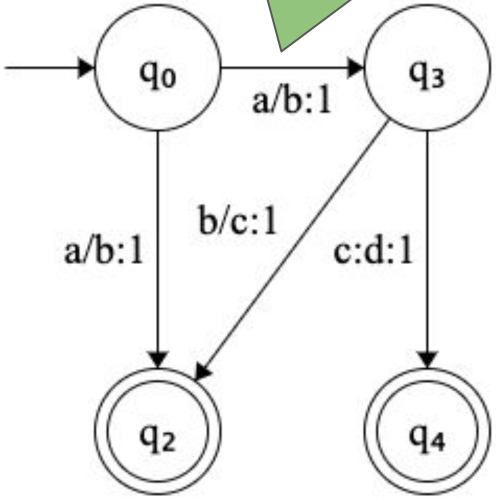
Weighted Finite State Transducer

Defined by:

- a finite set of states, Q
  - a start state, $q_0 \in Q$
  - a set of final states, $F \subseteq Q$
- a finite alphabet of input symbols, Σ
- a finite alphabet of output symbols, Ω
- a transition function that maps a state pair and a pair of symbols (or ε) to weights, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Omega \cup \{\varepsilon\}) \times Q \rightarrow R$
- an initial weight function, $\lambda: Q \rightarrow R$
- a final weight function, $\rho: Q \rightarrow R$

These are from the "weighted" part.

Notation: input symbol / output symbol / weight



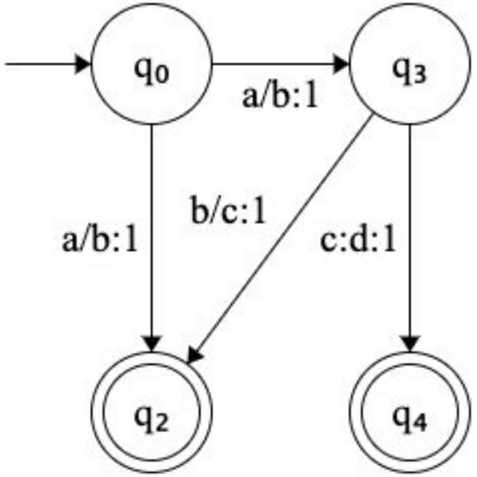
Weighted Finite State Transducer

Key: it is still a weighted FSA, but also “emit” symbols along the way!

Example:

Input: ***ab***

Output:



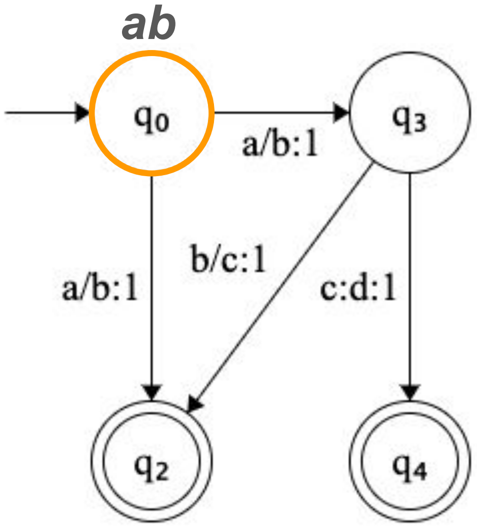
Weighted Finite State Transducer

Key: it is still a weighted FSA, but also "emit" symbols along the way!

Example:

Input: ***ab***

Output:



Start as usual.

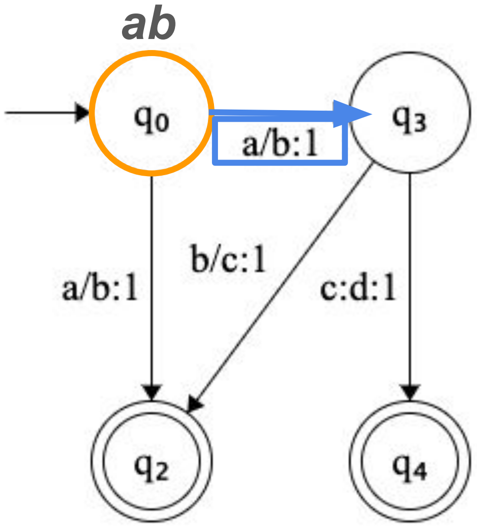
Weighted Finite State Transducer

Key: it is still a weighted FSA, but also “emit” symbols along the way!

Example:

Input: ***ab***

Output:



By taking this transition, it emits a symbol “b”.

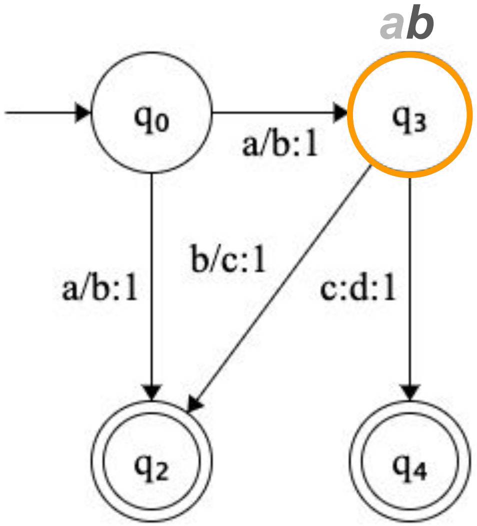
Weighted Finite State Transducer

Key: it is still a weighted FSA, but also “emit” symbols along the way!

Example:

Input: ***ab***

Output: ***b***



By taking this transition, it emits a symbol “b”.

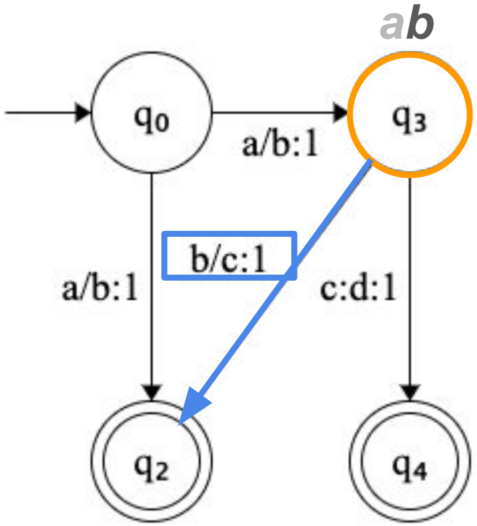
Weighted Finite State Transducer

Key: it is still a weighted FSA, but also "emit" symbols along the way!

Example:

Input: ***ab***

Output: ***b***



By taking this transition, it emits a symbol "c".

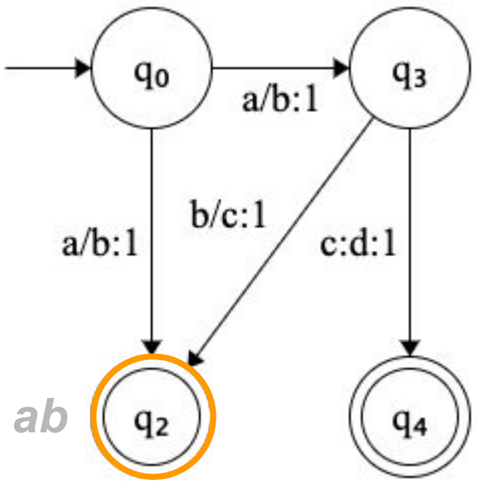
Weighted Finite State Transducer

Key: it is still a weighted FSA, but also “emit” symbols along the way!

Example:

Input: ***ab***

Output: ***bc***



By taking this transition, it emits a symbol “c”.

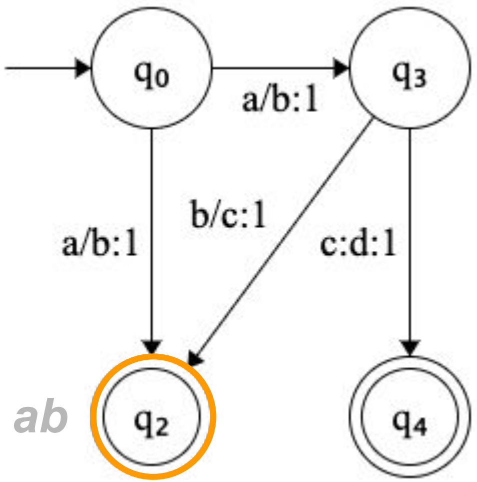
Weighted Finite State Transducer

Key: it is still a weighted FSA, but also “emit” symbols along the way!

Example:

Input: ***ab***

Output: ***bc***

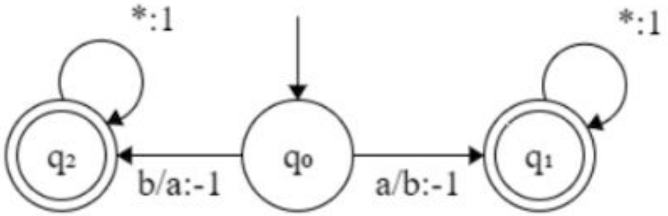


At this point, we consumed all of the input symbols and landed on a final state!

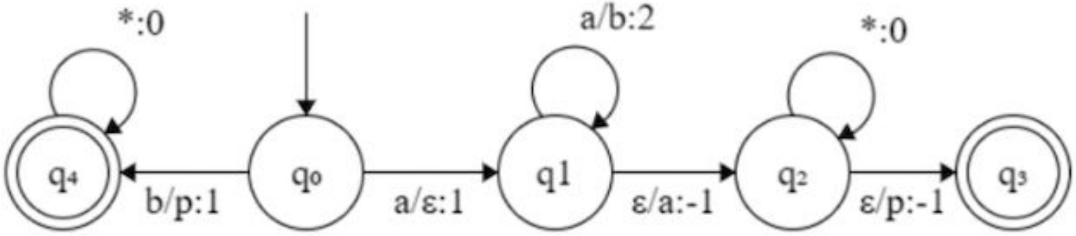
Quiz 5 - Problem 1 Setup

Fill in the output given the input after applying each WFST.

G

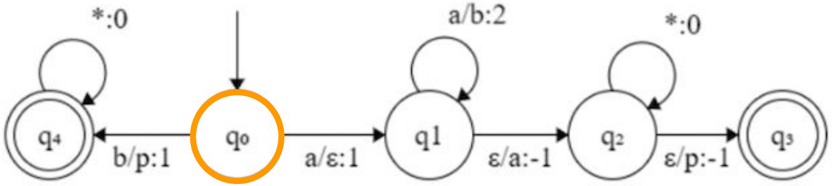


F



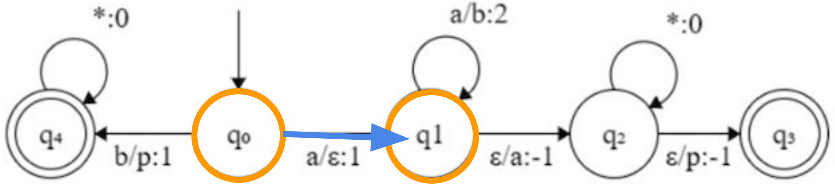
Quiz 5 - Problem 1 - F(abc)

Input	Output	State
abc	ε	q_0



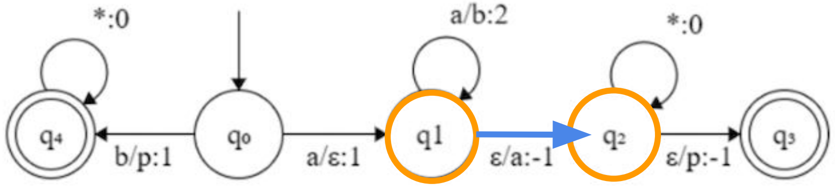
Quiz 5 - Problem 1 - F(abc)

Input	Output	State
abc	ε	q_0
bc	ε	q_1



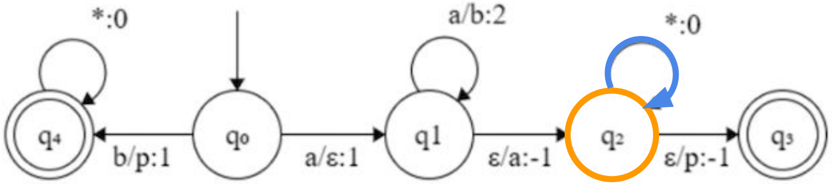
Quiz 5 - Problem 1 - F(abc)

Input	Output	State
abc	ε	q_0
bc	ε	q_1
bc	a	q_2



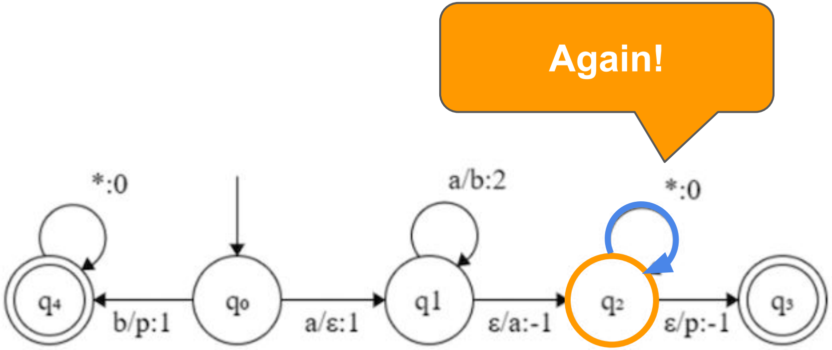
Quiz 5 - Problem 1 - F(abc)

Input	Output	State
abc	ε	q_0
bc	ε	q_1
bc	a	q_2
c	ab	q_2



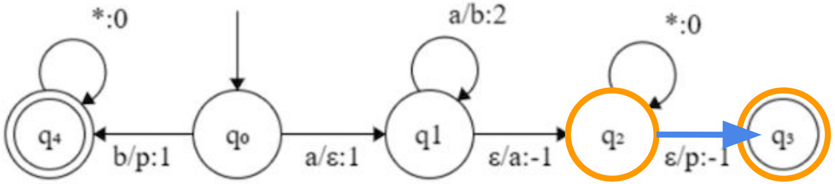
Quiz 5 - Problem 1 - F(abc)

Input	Output	State
abc	ε	q_0
bc	ε	q_1
bc	a	q_2
c	ab	q_2
ε	abc	q_2



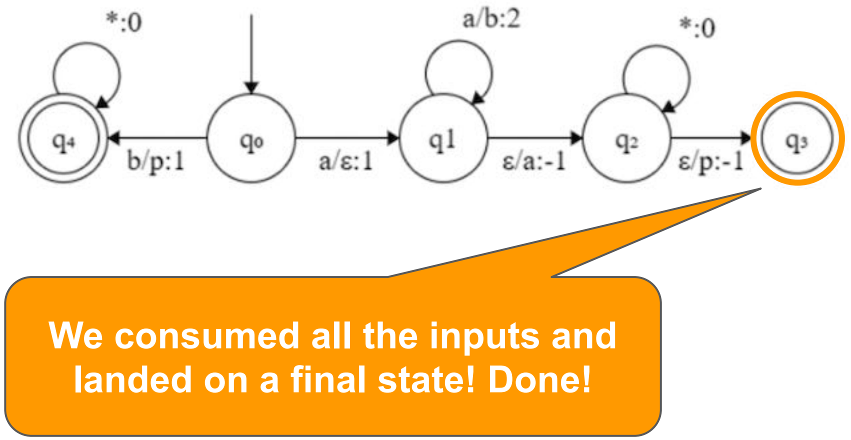
Quiz 5 - Problem 1 - F(abc)

Input	Output	State
abc	ε	q_0
bc	ε	q_1
bc	a	q_2
c	ab	q_2
ε	abc	q_2
ε	abcp	q_3



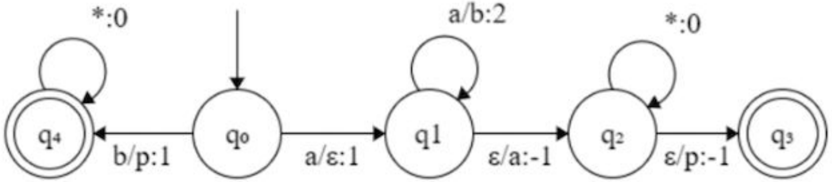
Quiz 5 - Problem 1 - F(abc)

Input	Output	State
abc	ε	q_0
bc	ε	q_1
bc	a	q_2
c	ab	q_2
ε	abc	q_2
ε	abcp	q_3

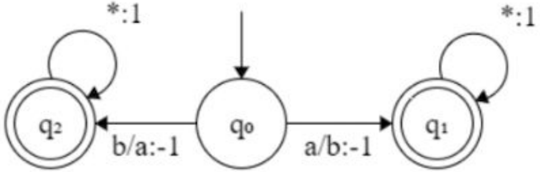


Quiz 5 - Problem 1 - G ○ F(abbc)

Input **Output** **State**



F

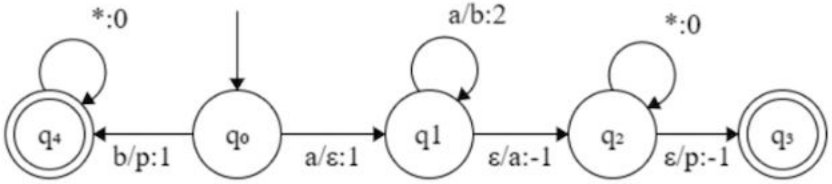


G

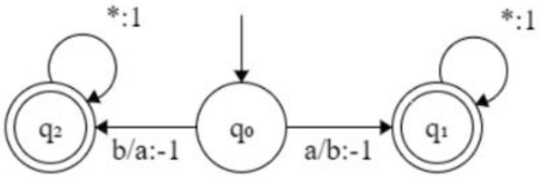
Quiz 5 - Problem 1 - G ○ F(abbc)

This means we pass "abbc" through F first, then pass its output through G (note the ordering).

Input **Output** **State**



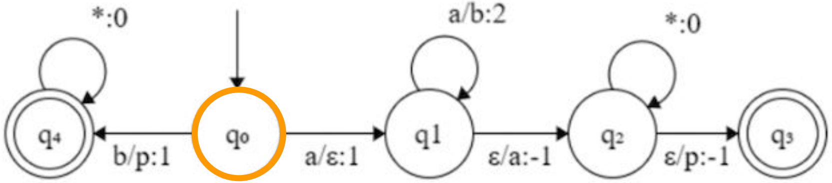
F



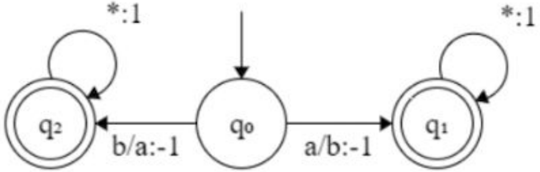
G

Quiz 5 - Problem 1 - G ○ F(abbc)

Input	Output	State
abbc	ε	q_0



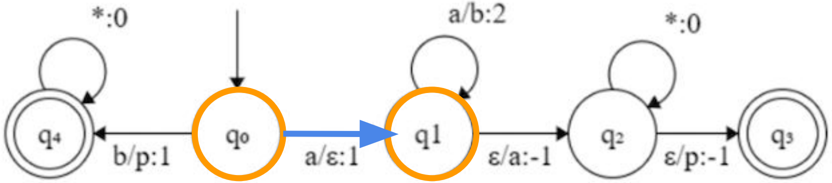
F



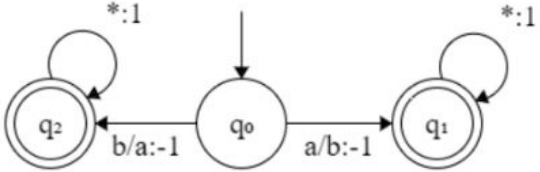
G

Quiz 5 - Problem 1 - G ○ F(abbc)

Input	Output	State
abbc	ε	q_0
bbc	ε	q_1



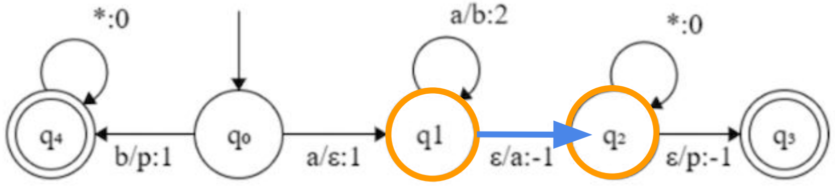
F



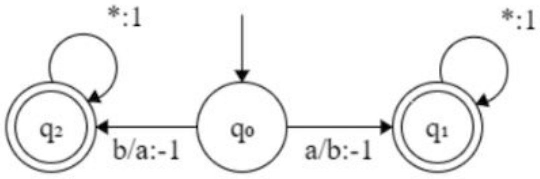
G

Quiz 5 - Problem 1 - G ○ F(abbc)

Input	Output	State
abbc	ε	q_0
bbc	ε	q_1
bbc	a	q_2



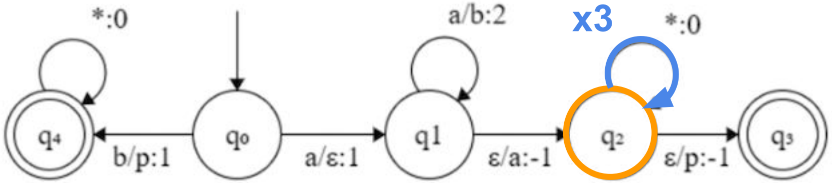
F



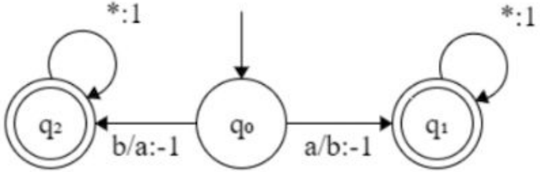
G

Quiz 5 - Problem 1 - G ○ F(abbc)

Input	Output	State
abbc	ε	q_0
bbc	ε	q_1
bbc	a	q_2
bc	ab	q_2
c	abb	q_2
ε	abbc	q_2



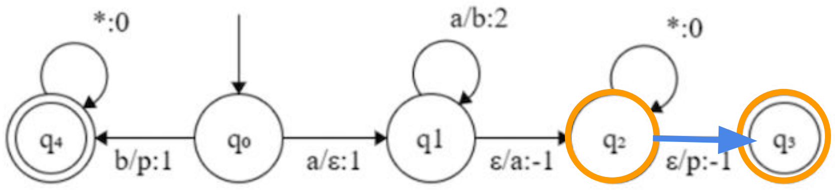
F



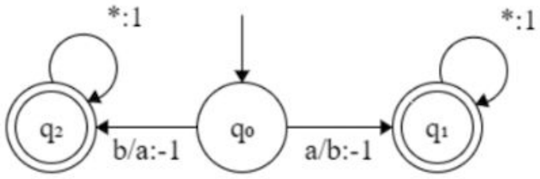
G

Quiz 5 - Problem 1 - G ○ F(abbc)

Input	Output	State
abbc	ε	q_0
bbc	ε	q_1
bbc	a	q_2
bc	ab	q_2
c	abb	q_2
ε	abbc	q_2
ε	abbcp	q_3



F

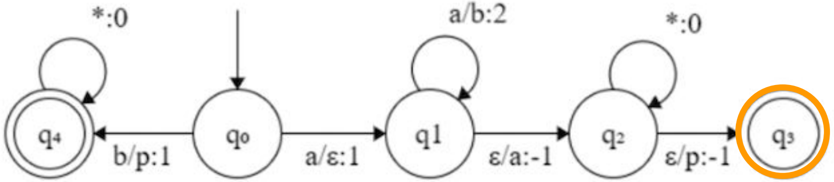


G

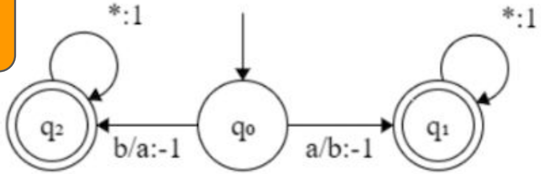
Quiz 5 - Problem 1 - G ○ F(abbc)

Input	**Output**	**State**
abbc	ε	q_0
bbc	ε	q_1
bbc	a	q_2
bc	ab	q
c	ab	2
ε	abbc	q_2
ε	abbcp	q_3

Now treat this as the input to G.



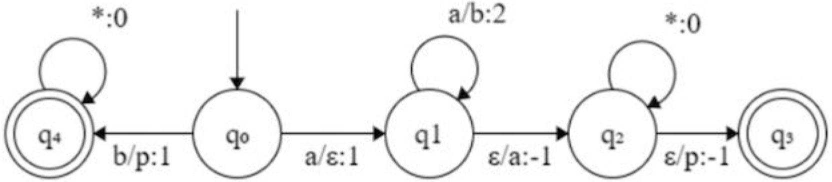
F



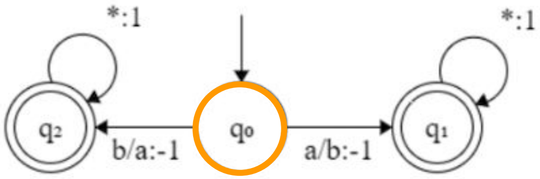
G

Quiz 5 - Problem 1 - G ○ F(abbc)

Input	**Output**	**State**
abbcp	ε	q_0



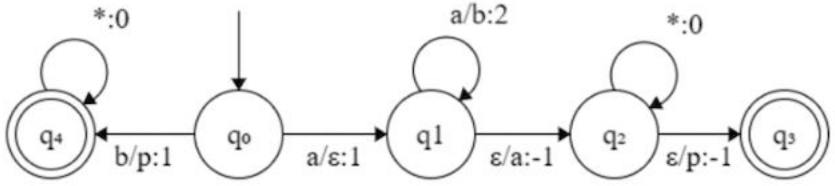
F



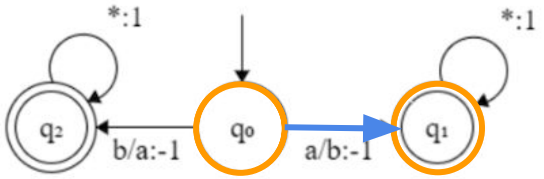
G

Quiz 5 - Problem 1 - G ○ F(abbc)

Input	Output	State
abbcp	ε	q_0
bbcp	b	q_1



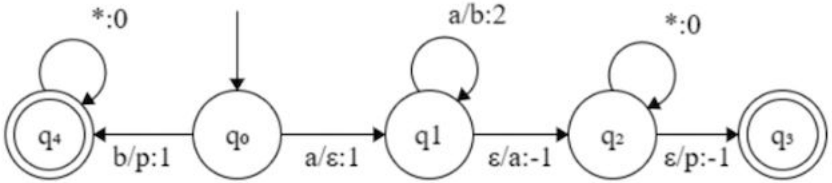
F



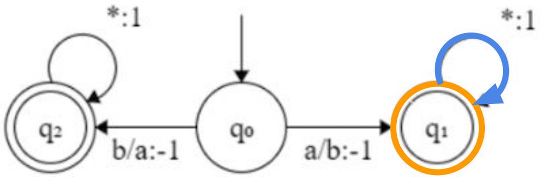
G

Quiz 5 - Problem 1 - G ○ F(abbc)

Input	Output	State
abbcp	ε	q_0
bbcp	b	q_1
bcp	bb	q_1



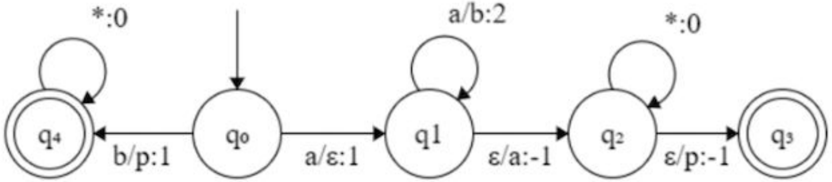
F



G

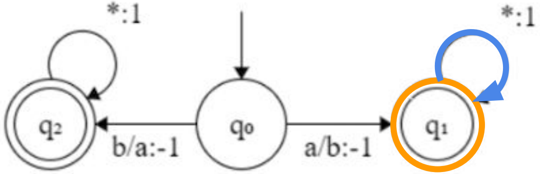
Quiz 5 - Problem 1 - G ○ F(abbc)

Input	Output	State
abbcp	ε	q_0
bbcp	b	q_1
bcp	bb	q_1



F

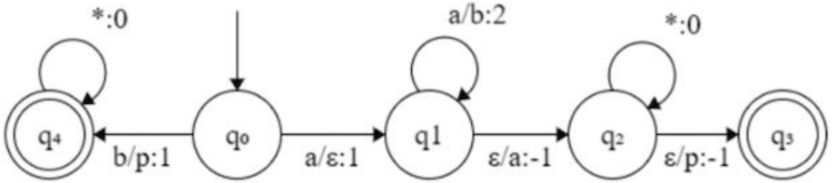
3 steps later



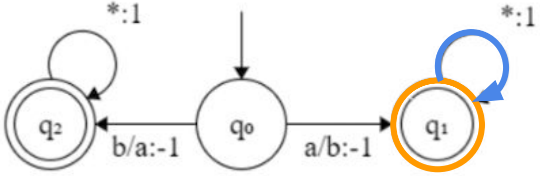
G

Quiz 5 - Problem 1 - G ○ F(abbc)

Input	Output	State
abbcp	ε	q_0
bbcp	b	q_1
bcp	bb	q_1
	...	
ε	bbbcp	q_1



F

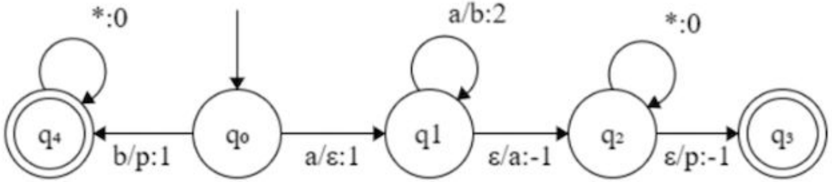


G

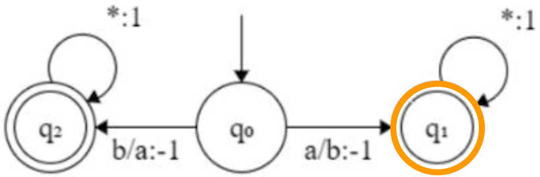
Quiz 5 - Problem 1 - G ○ F(abbc)

Input	Output	State
abbcp	ε	q_0
bbcp	b	q_1
bcp	bb	q_1
	...	
ε	bbbcp	q_1

This is the output of G ○ F(abbc).



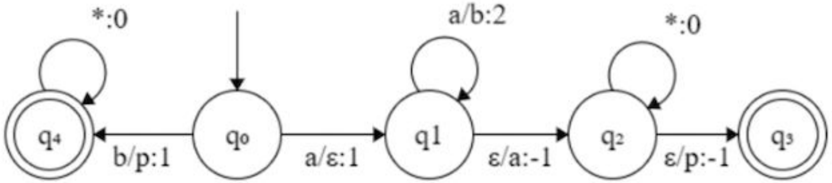
F



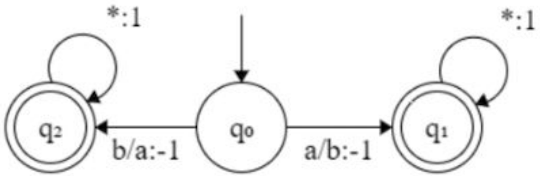
G

Quiz 5 - Problem 1 - F ○ G(abbc)

Input	**Output**	**State**
abbc	ε	q_0



F

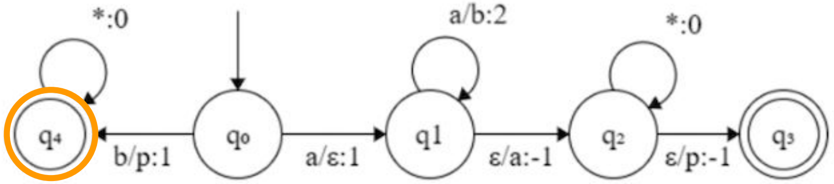


G

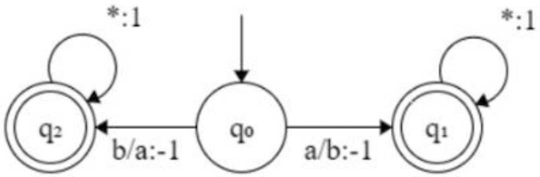
Quiz 5 - Problem 1 - F ○ G(abbc)

Input	Output	State
abbc	ε	q_0
	...	
ε	pbbc	q_4

You get the gist of it!



F



G

Quiz 5 - Problem 2 - Setup

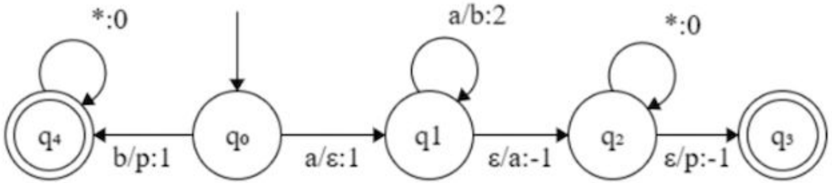
Given the two WFST, what is the score of the path of applying F(abc)?

What about G(aabc)?

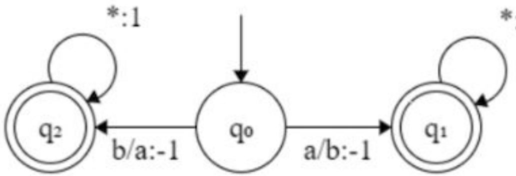
Also given for both WFSTs:

$$\lambda(q_0) = 1$$

$$\rho(q_n) = n$$



F



G

Quiz 5 - Problem 2 - Setup

Given the two WFST, what is the score of the path of applying F(abc)?

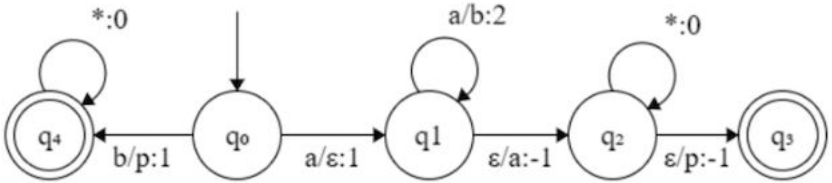
What about G(aabc)?

Also given for both WFSTs:

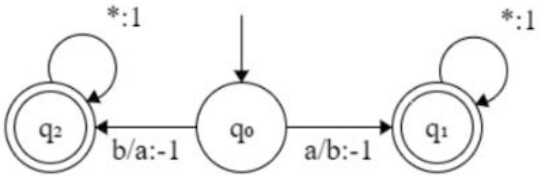
$$\lambda\left(q_0\right) = 1$$

What is the cost of *starting* at q_0?

$$\rho\left(q_n\right) = n$$



F



G

Quiz 5 - Problem 2 - Setup

Given the two WFST, what is the score of the path of applying F(abc)?

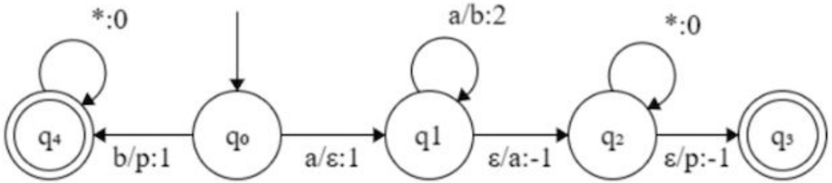
What about G(aabc)?

Also given for both WFSTs:

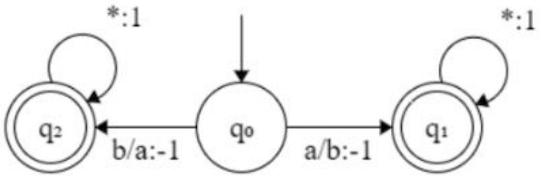
$$\lambda(q_0) = 1$$

$$\rho(q_n) = n$$

What is the cost of *ending* at q_n?



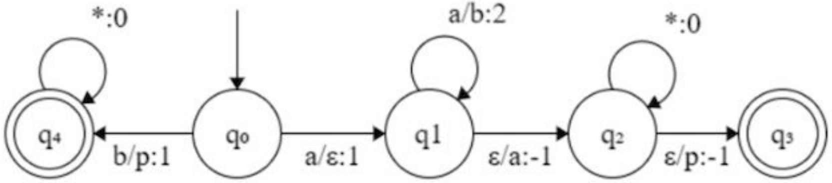
F



G

Quiz 5 - Problem 2 - F(abc)

Input **Output** **State**

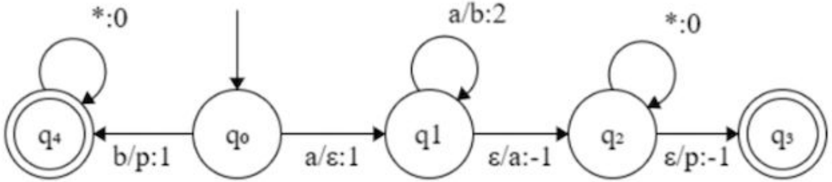


F

Cost:

Quiz 5 - Problem 2 - F(abc)

Input **Output** **State**



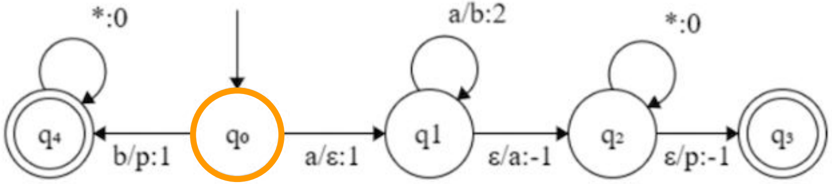
F

Cost: 0

We did this one already -- except this time we also keep track of the cost.

Quiz 5 - Problem 2 - F(abc)

Input	Output	State
abc	ε	q_0



F

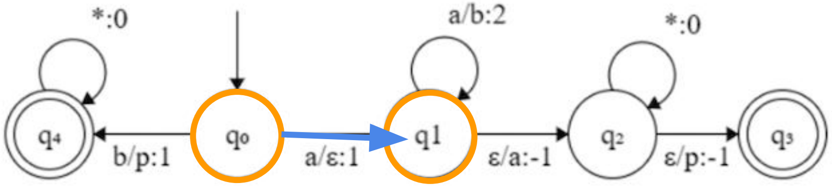
Cost: 0+1=1

We add 1 to cost because

$$\lambda(q_0) = 1$$

Quiz 5 - Problem 2 - F(abc)

Input	Output	State
abc	ε	q_0
bc	ε	q_1

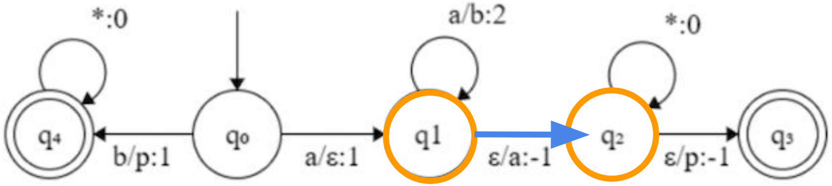


F

Cost: 1+1=2

Quiz 5 - Problem 2 - F(abc)

Input	Output	State
abc	ε	q_0
bc	ε	q_1
bc	a	q_2

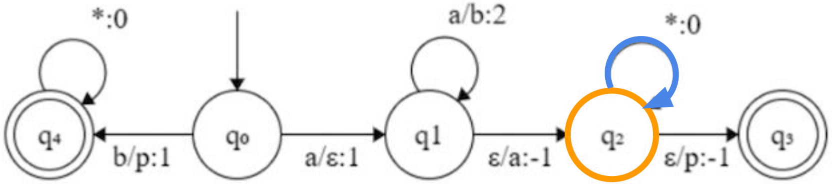


F

Cost: 2+(-1)=1

Quiz 5 - Problem 2 - F(abc)

Input	Output	State
abc	ε	q_0
bc	ε	q_1
bc	a	q_2
c	ab	q_2

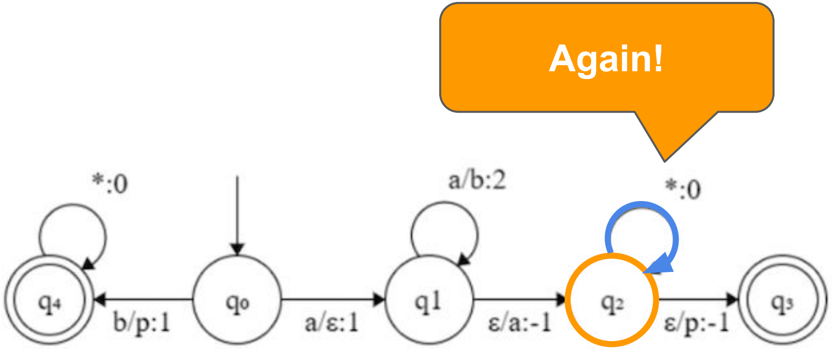


F

Cost: 1+0=1

Quiz 5 - Problem 2 - F(abc)

Input	Output	State
abc	ε	q_0
bc	ε	q_1
bc	a	q_2
c	ab	q_2
ε	abc	q_2

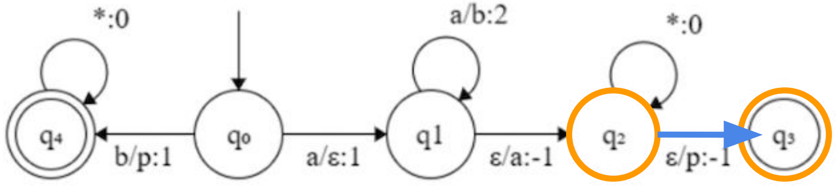


F

Cost: 1+0=1

Quiz 5 - Problem 2 - F(abc)

Input	Output	State
abc	ε	q_0
bc	ε	q_1
bc	a	q_2
c	ab	q_2
ε	abc	q_2
ε	abcp	q_3

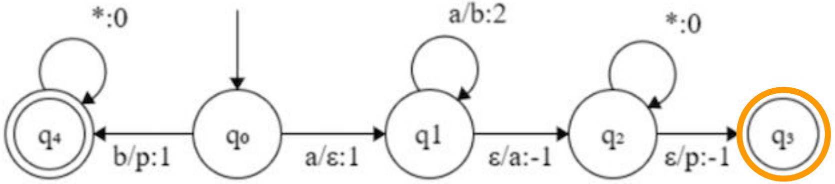


F

Cost: 1+(-1)=0

Quiz 5 - Problem 2 - F(abc)

Input	Output	State
abc	ε	q_0
bc	ε	q_1
bc	a	q_2
c	ab	q_2
ε	abc	q_2
ε	abcp	q_3



F

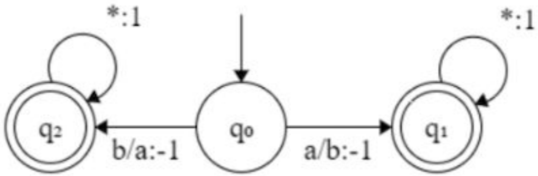
Cost: 0+3=3

We add 3 to cost because

$\rho(q_3) = 3$

Quiz 5 - Problem 2 - G(aabc)

Input **Output** **State**

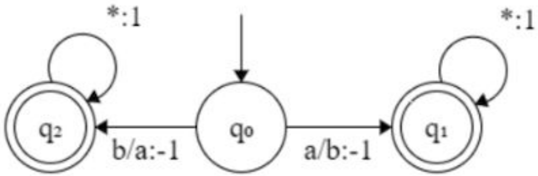


G

Cost: 0

Quiz 5 - Problem 2 - G(aabc)

Input	Output	State
	...	
ε	babc	q_1



G

Cost: 4

Q & A