San José State University
Department of Computer Science

Ahmad Yazdankhah
ahmad.yazdankhah@sjsu.edu
www.cs.sjsu.edu/~yazdankhah

Deterministic Finite Automata

(Part 3)

Lecture 08

Day 08/31

CS 154

Formal Languages and Computability

Spring 2018


A MILLION DOLLAR QUESTION!!
This is me!
Dec 6. 2017

Agenda of Day 08

- Solution and Feedback of HW 1
- Solution and Feedback of Quiz 2
- ~~Summary of Lecture 07~~
- Lecture 08: Teaching …
  - Deterministic Finite Automata (Part 3)

Solution and Feedback of HW 1 (Out of 30)

Metrics	Section 1	Section 2	Section 3
Average	22	25	25
High Score	28	30	30
Low Score	14	18	10

Solution and Feedback of Quiz 2 (Out of 27)

Metrics	Section 1	Section 2	Section 3
Average	22	21	21
High Score	25	26	25
Low Score	17	16	14

Definitions

Formal Definition of DFAs

- Here is the formal (mathematical) definition of DFAs:
- A DFA M is defined by the quintuple (5-tuple):

$$M = (Q, \Sigma, \delta, q_0, F)$$

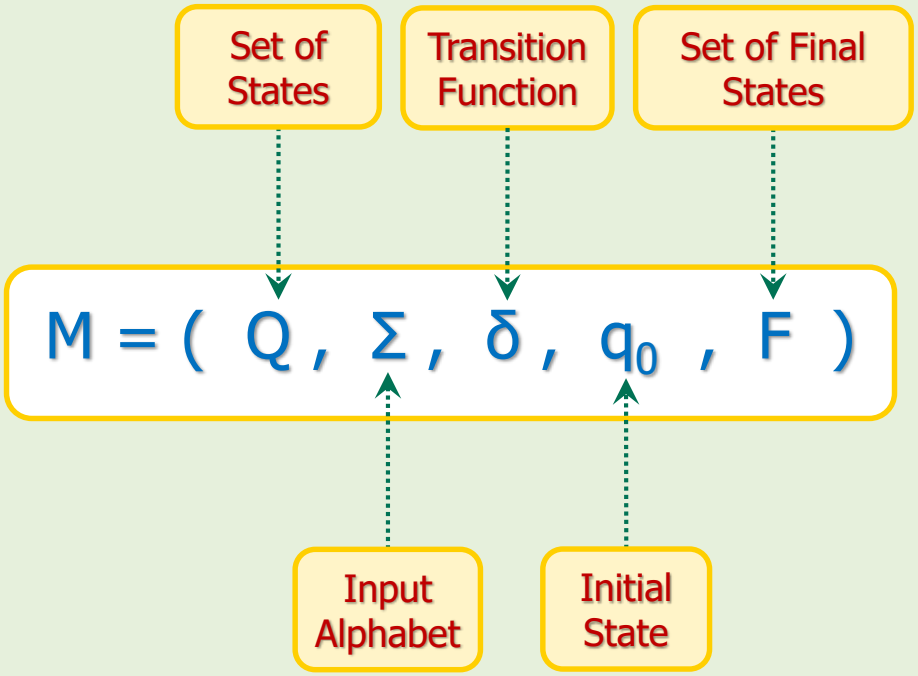
- Where:
  - Q is a finite and nonempty set of states of the transition graph.
  - Σ is a finite and nonempty set of symbols called input alphabet.
  - δ is called transition function and is defined as:

$$\delta: Q \times \Sigma \rightarrow Q$$

δ must be total function. Why?

  - $q_0 \in Q$ is the initial state of the transition graph.
  - $F \subseteq Q$ is the set of accepting states of the transition graph.

Formal Definition of DFAs



Transition Function $\delta : Q \times \Sigma \rightarrow Q$

Example 22

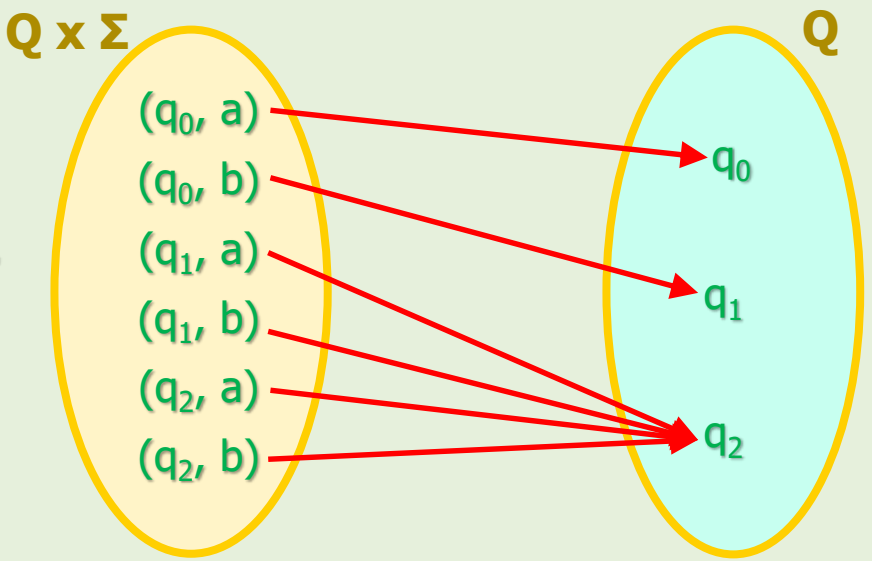
- Let $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a , b\}$; Domain = ? , Range = ?

- Domain: $Q \times \Sigma = \{(q_0, a), (q_0, b), (q_1, a), (q_1, b), (q_2, a), (q_2, b)\}$
  - Note that the domain contains all possible combination of states and alphabet.

- Range: $Q = \{q_0, q_1, q_2\}$

- Rule: Let's assume that this figure is the rule of the function.

- Is this a "total function" or "partial function"? Why?

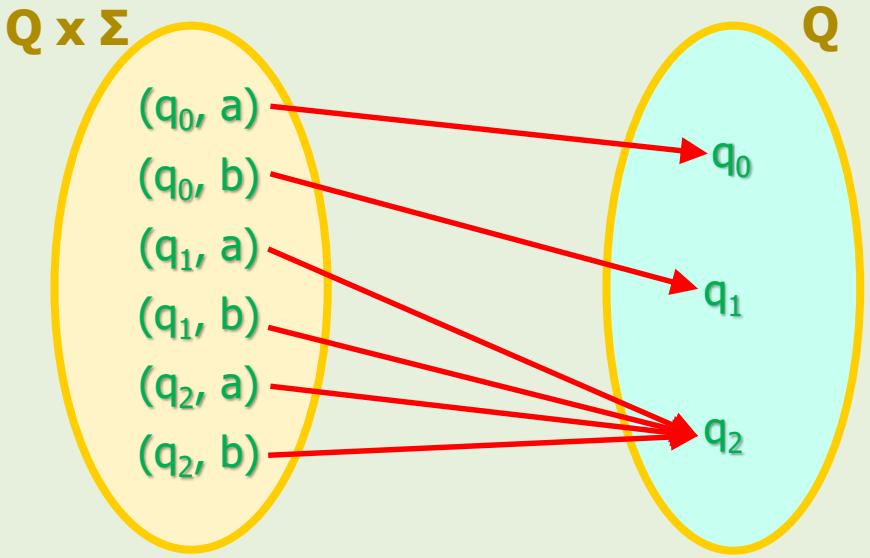


Transition Function $\delta : Q \times \Sigma \rightarrow Q$

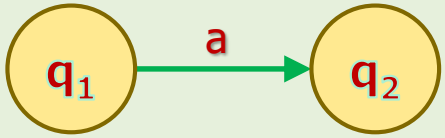
Example 22 (cont'd)

- Write the rule of δ in algebraic notation.

$$\delta: \begin{cases} \delta(q_0, a) = q_0 \\ \delta(q_0, b) = q_1 \\ \delta(q_1, a) = q_2 \\ \delta(q_1, b) = q_2 \\ \delta(q_2, a) = q_2 \\ \delta(q_2, b) = q_2 \end{cases}$$



- How do we show a sub-rule like $\delta(q_1, a) = q_2$ in transition graph?

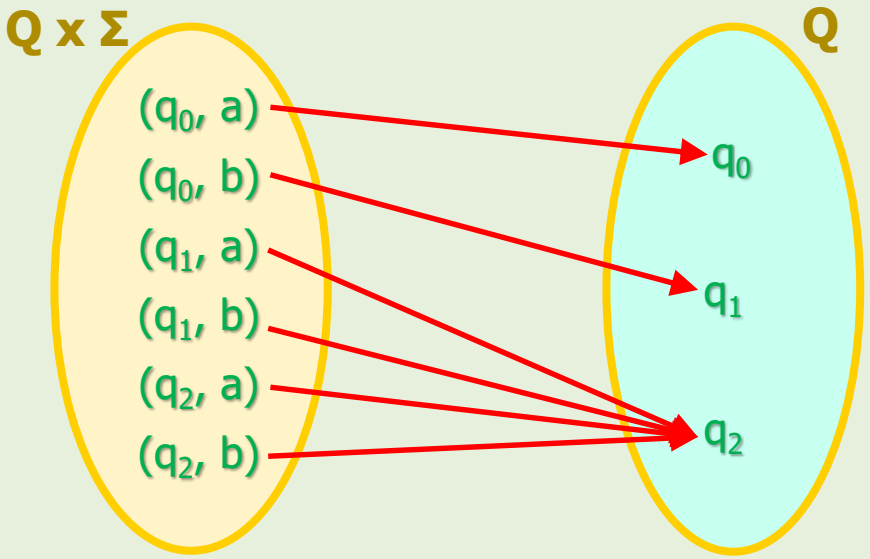


- So, every sub-rule is a transition in transition graph.

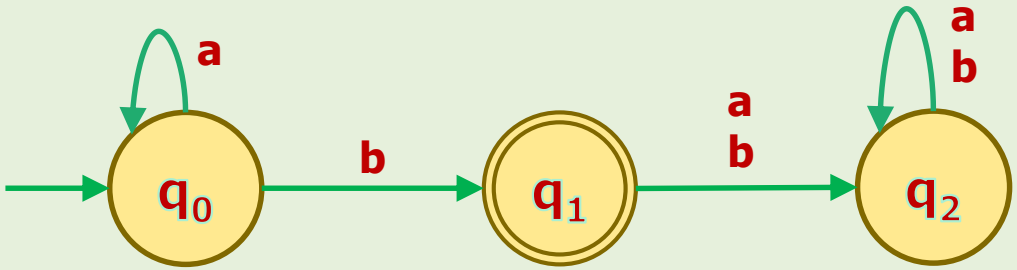
Draw the Transition Graph from DFA Definition

Example 22 (cont'd)

$$\delta: \begin{cases} \delta(q_0, a) = q_0 \\ \delta(q_0, b) = q_1 \\ \delta(q_1, a) = q_2 \\ \delta(q_1, b) = q_2 \\ \delta(q_2, a) = q_2 \\ \delta(q_2, b) = q_2 \end{cases} , \quad \Sigma = \{a , b\}$$



- Initial state = q_0
  Final state = $\{q_1\}$
- Draw the transition graph.



Formal Definition and Transition Graph

Homework

- Draw a transition graph for the DFA M defined as:
- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{a , b\}$
- δ:

$$\begin{cases} \delta(q_0, a) = q_1 \\ \delta(q_0, b) = q_3 \\ \delta(q_1, a) = q_3 \\ \delta(q_1, b) = q_2 \\ \delta(q_2, a) = q_2 \\ \delta(q_2, b) = q_2 \\ \delta(q_3, a) = q_3 \\ \delta(q_3, b) = q_3 \end{cases}$$

- Initial state = q_0
- $F = \{q_2\}$

Homework

- Draw a transition graph for

$M = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_1\})$

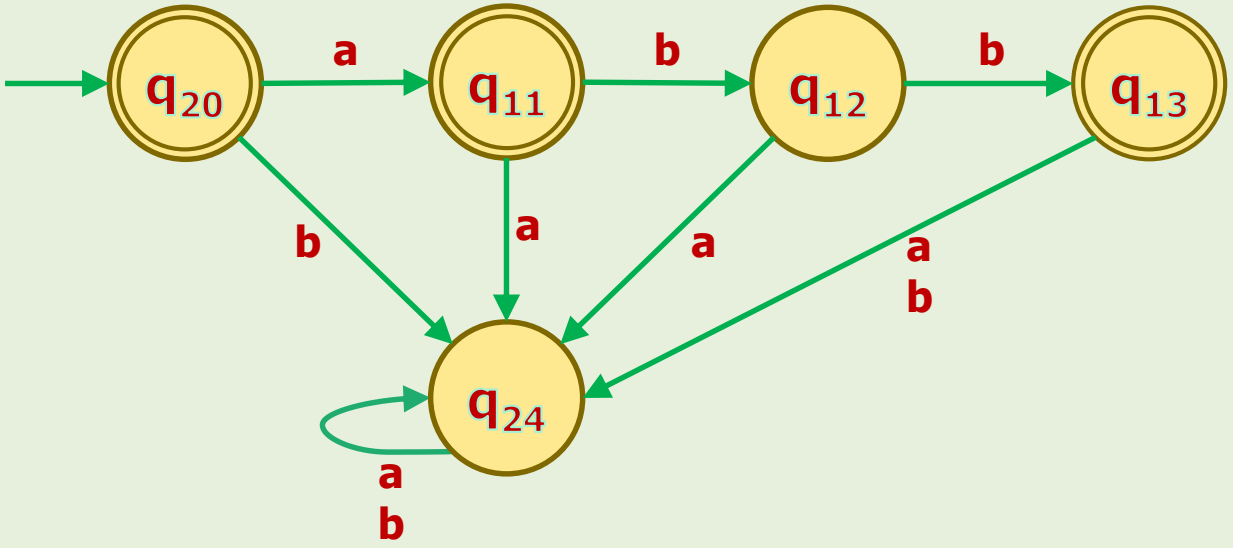
$$\delta: \begin{cases} \delta(q_0, 0) = q_0 \\ \delta(q_1, 0) = q_0 \\ \delta(q_2, 0) = q_2 \\ \delta(q_0, 1) = q_1 \\ \delta(q_1, 1) = q_2 \\ \delta(q_2, 1) = q_1 \end{cases}$$

- Which strings from the following set are accepted?

{01, 00, 101, 0111, 11001, 100, 1100}

Homework

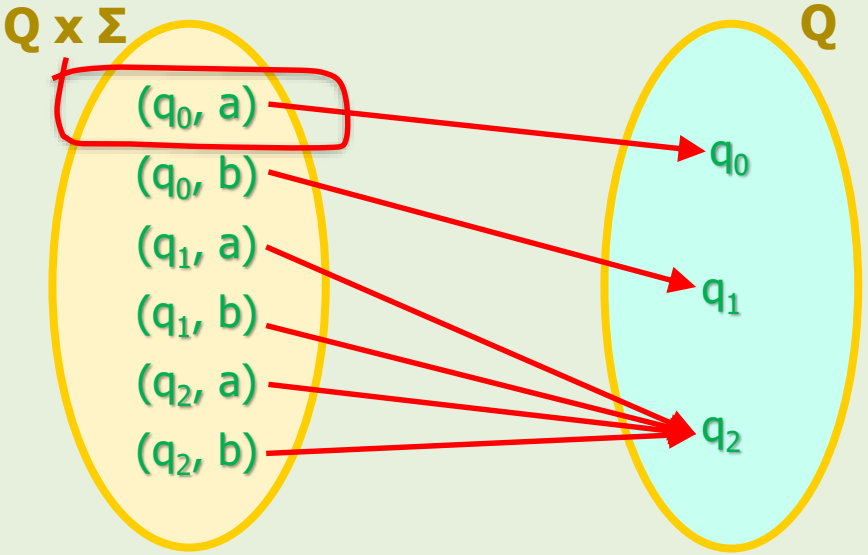
- Write all elements of the following transition graph.
- Q = ?
- Σ = ?
- δ = ?
- q_0 = ?
- F = ?

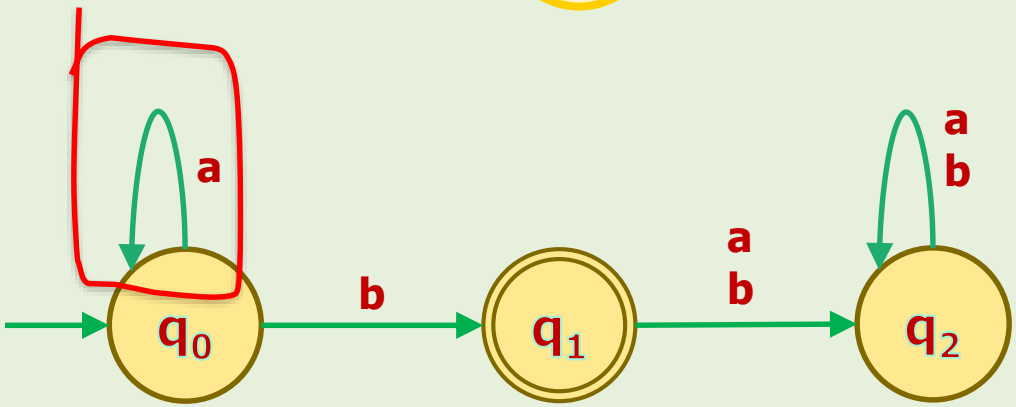


Why Total Function

Example 22 (cont'd)

- What would happen if δ is NOT total function?
- Then at least one member of domain is undefined!
- To see the effect, let's modify δ by making (q_0, a) undefined.

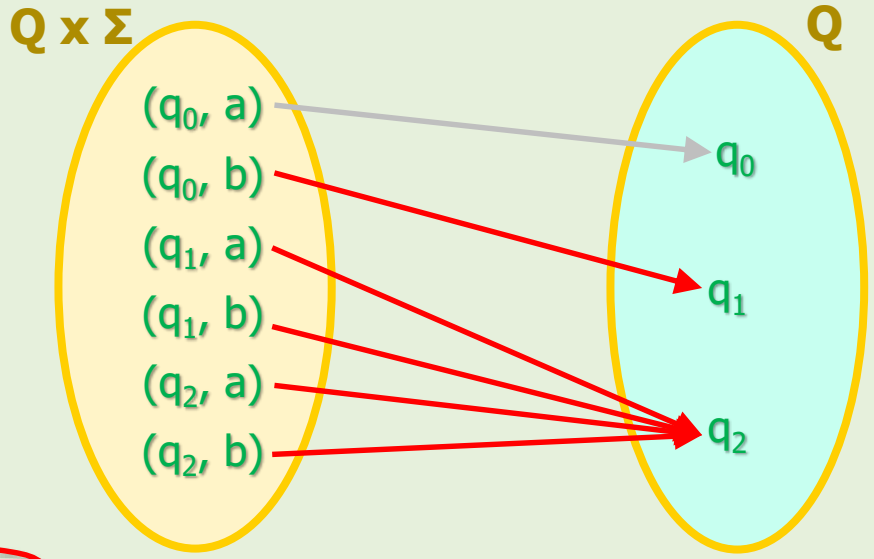




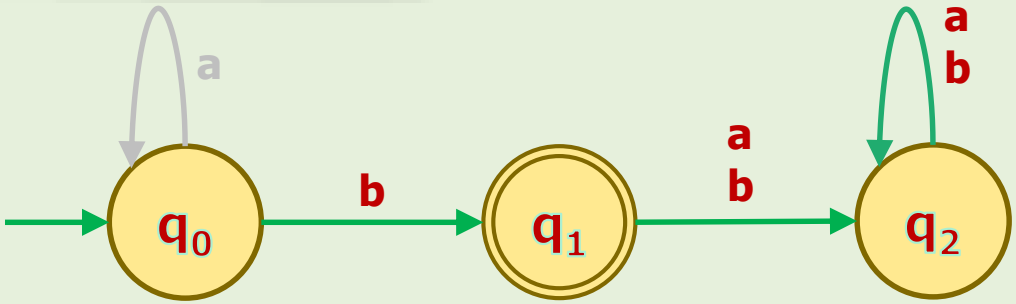
Why Total Function

Example 22 (cont'd)

- What is the effect of δ being partial function?
- If the DFA is in q_0 and the input is a, it does not know where to go!
- It "hangs"!
- So, δ must be total function.



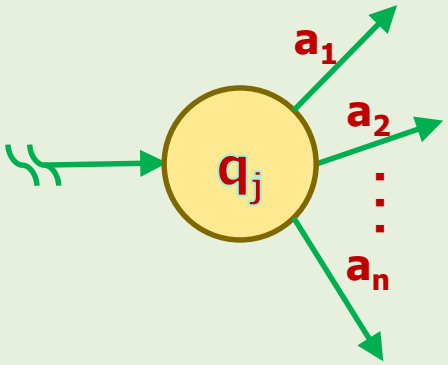
The gray items don't exist!



Total Function from Different Angle

- So, DFAs must know where to go at any moment of their operation.

- This means, in general:
  - If $\Sigma = \{a_1 , a_2 , \dots , a_n\}$ is the alphabet of a DFA, then ...
  - ... every state q_j, must have an outgoing transition a_k for k = 1, 2, ..., n.



Conclusion

- δ being total function is "DFAs' constraint".

Transition Table

- To represent a transition function, we can also use a table called "transition table".

Example 23

- Represent the following transition function by using a transition table.

- δ: $\begin{cases} \delta(q_0, a) = q_0 \\ \delta(q_1, a) = q_0 \\ \delta(q_2, a) = q_2 \\ \delta(q_0, b) = q_1 \\ \delta(q_1, b) = q_2 \\ \delta(q_2, b) = q_1 \end{cases}$

Alphabet

δ	a	b
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_2	q_1

States

Range

Associated Language to DFAs

- Every DFA accepts a set of strings.
  - Of course the set can be empty!
- A set of string is called a language.
- Therefore, every DFA accepts a language.

Definition

- The associated language to a DFA is the set of all strings that it accepts.
- The associated language L to the machine M is denoted by L(M).

- Note that this definition can be extended to all types of automata.

Equivalency of Machines

- When are two machines M_1 and M_2 equivalent?

Definition

- Machine M_1 is equivalent to machine M_2 iff $L(M_1) = L(M_2)$.
  - M_1 and M_2 are equivalent iff their associated languages are equal.

- Note that this is also a general definition for all types of automata.

- What is wrong with the following definition?

  Two machines are equivalent iff both accept the same language.

What is Computation?

- A machine during its operation transits (aka moves) from one configuration to another.
- Ultimately, when the machine halts, it accepts or rejects a string.
- This sequence of configurations is called "computation".

Definition of Computation

- "Computation" is the sequence of configurations from when the machine starts until it halts.

What is Determinism?

Etymology

- Merriam-Webster dictionary defines "determinism" as:
  - "the belief that all events are caused by things that happened before them and that people have no real ability to make choices or control what happens"

- This is a philosophical definition.
- If something is deterministic, then it will happen (with 100% certainty) and there won't be any other choices.

- Let's see what does it mean in computer science world.

What is Determinism?

Is DFAs' behavior predictable?

- You (an observer) are given a known DFA's configuration at timeframe n.

- Can you predict its configuration at timeframe n+1?
  - Yes, we (an observer) can predict its behavior with 100% certainty.
  - Because during any timeframe, there is one and only one transition.

What is Determinism?

Definition

- A machine is called deterministic iff during any timeframe, there is NO MORE THAN ONE transition.

- This means, the number of transitions can be zero or one.
- In DFAs, it is always one but in other machines, it can be zero as well.
  - Will be covered later.

- How can this definition be violated?

References

1. Linz, Peter, "An Introduction to Formal Languages and Automata, 5th ed.," Jones & Bartlett Learning, LLC, Canada, 2012
2. Kenneth H. Rosen, "Discrete Mathematics and Its Applications, 7th ed.," McGraw Hill, New York, United States, 2012
3. Michael Sipser, "Introduction to the Theory of Computation, 3rd ed.," CENGAGE Learning, United States, 2013
   ISBN-13: 978-1133187790