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 2019

Agenda of Day 08

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

Solution and Feedback of HW 2 (Out of 15)

Section	Average	High Score	Low Score
01 (TR 3:00 PM)	10.5	15	0
02 (TR 4:30 PM)	7	15	0
03 (TR 6:00 PM)	12.81	15	0

Summary of Lecture 07: We learned ...

DFAs

- DFAs' configuration is a snapshot of the machine's status in one timeframe.
- It is the combination of the following data:
  1. Input string + Position of the read-head
  2. Current state of the transition graph
  3. Timeframe number on the clock
- During a timeframe, the machine "transits" (aka "moves") from one configuration to another.

Design Notes

1. The machine should be designed in such a way that:

   It accepts all strings of **L**.

   AND

   It rejects all strings of $\overline{\mathbf{L}}$.

2. To test your machine, all accepted strings and rejected strings should be picked from Σ^*.
   - We are not allowed to input strings from outside of Σ^*.

Any question?

Summary of Lecture 07: We learned ...

When DFAs halt

- When all input symbols are consumed.

$$h \leftrightarrow c$$

How DFAs accept the string w

- Three conditions should be satisfied:
  - The DFA halts. ≡ **h**
  - All symbols of w are consumed. ≡ **c**
  - The DFA is in an accepting state. ≡ **f**

$$(h \wedge c \wedge f) \leftrightarrow a$$

- For DFAs, h and c are equivalent.
- So, the logical statement of accepting a string is:

$$(c \wedge f) \leftrightarrow a$$

How DFAs reject the string w

- We need to negate the previous statement:

$$\sim (c \wedge f) \leftrightarrow \sim a$$

$$\equiv (\sim c \vee \sim f) \leftrightarrow \sim a$$

- Translation:
  - At least one symbol is NOT consumed.

  OR

  - The DFA is NOT in an accepting state.

Any question?

Design Examples

Example 14

- Let L be the set of strings that contains even number of 1's over Σ = {1}.

  a. Write a set-builder for L.

  b. Design a DFA to accept L.

Design Examples

Example 15

- Let L be the set of strings that contains even unary numbers over Σ = {1}.
  a. Write a set-builder for L.
  b. Design a DFA to accept L.

Design Examples: DFA over Σ = {a , b}

Example 16: Empty Language

- L = { }

Example 17: All Strings

- L = Σ*

Example 18

- L = {λ}

Homework

- L = Σ+

Homework: DFA Design

- Design a DFA for each of the following languages over Σ = {a , b}:

1. All strings that every odd positions is 'a'.
2. All strings that no two consecutive a's occur.
3. All strings that does not end with ab.
4. All strings in which every a is followed by bb.

Homework: DFA Design

- Design a DFA over Σ = {a, b} for each of the following languages:

1. All strings with exactly one 'a' and exactly two 'b's
2. All strings with at least one 'a' and exactly two 'b's
3. All strings with exactly two 'a's and more than two 'b's

Definitions

Formal Definition of DFAs

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

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

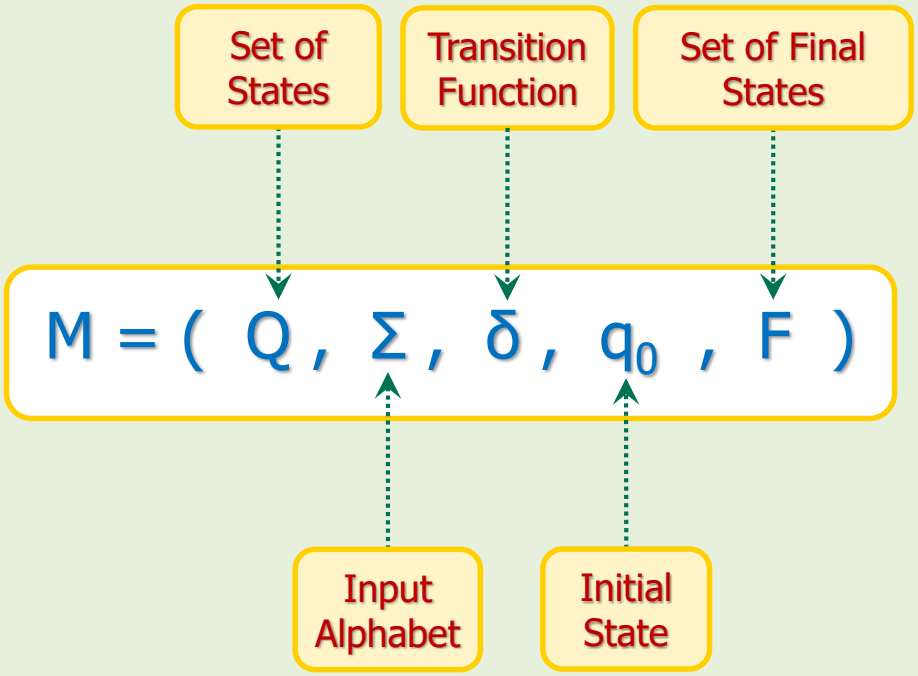
- Where:
  - Q is a finite and nonempty set of states of the transition graph.
  - Σ is a finite and nonempty set of symbols called input alphabet.
  - δ is called transition function (aka delta function) and is defined as:

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

δ is total function.

  - $q_0 \in Q$ 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 19

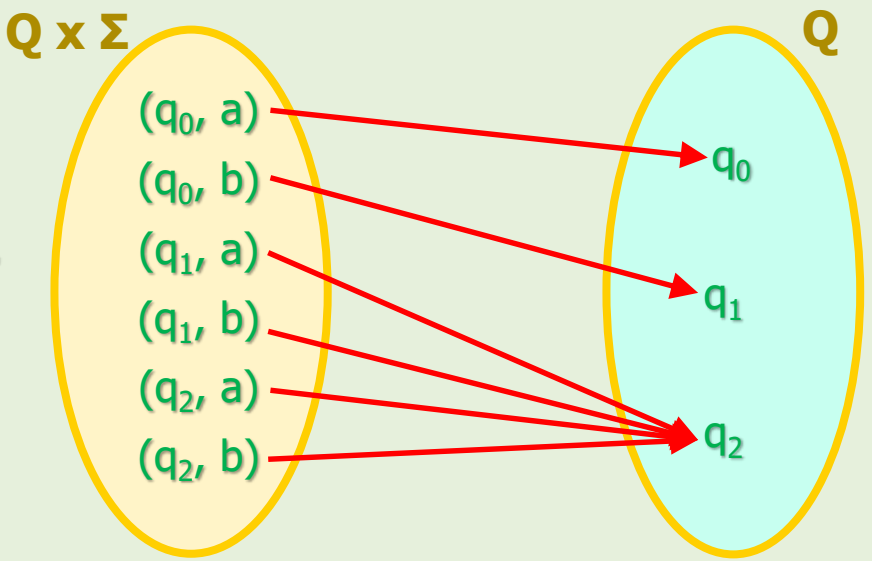
- Let $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a , b\}$; Domain = ? , Range = ?

- 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?

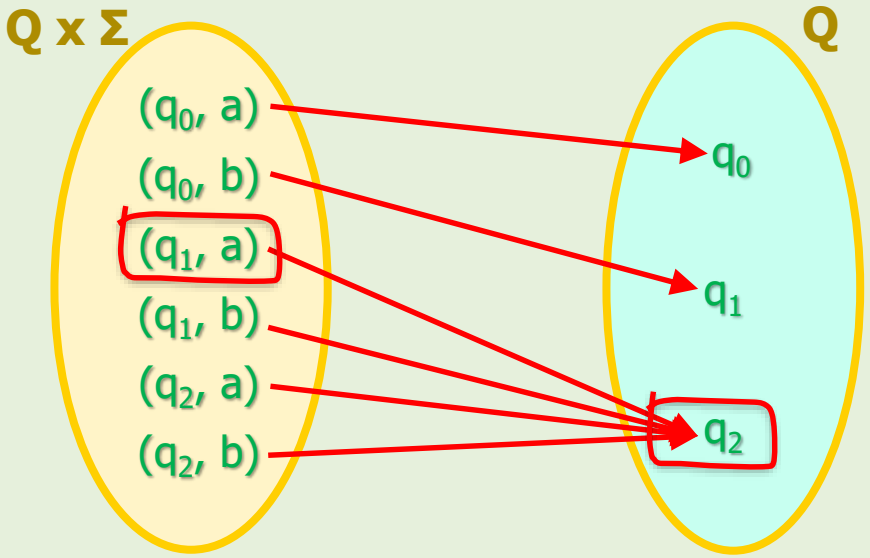


Transition Function δ : Q x Σ → Q

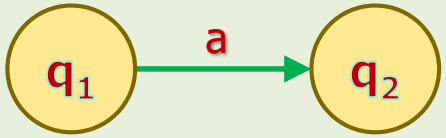
Example 19 (cont'd)

- 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?

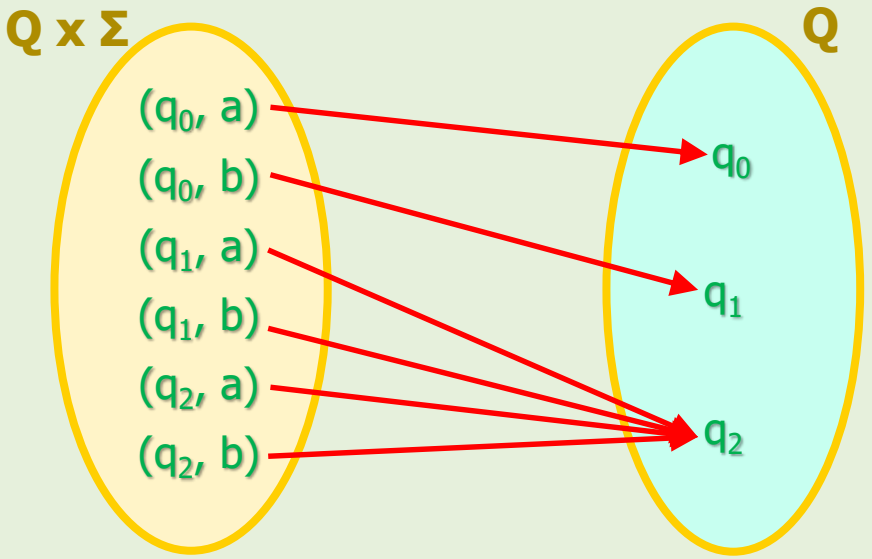


- So, every sub-rule of δ is a transition in transition graph.

Draw the Transition Graph from DFA Definition

Example 19 (cont'd)

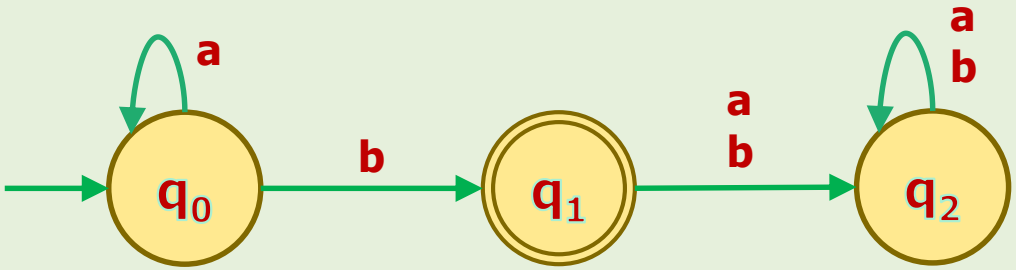
$$\delta: \begin{cases} \delta(q_0, a) = q_0 \\ \delta(q_0, b) = q_1 \\ \delta(q_1, a) = q_2 \\ \delta(q_1, b) = q_2 \\ \delta(q_2, a) = q_2 \\ \delta(q_2, b) = q_2 \end{cases} , \quad \Sigma = \{a , b\}$$



- Let's assume:

  Initial state = q_0
  Final states set = $\{q_1\}$

- Draw the transition graph.



Formal Definition and Transition Graph

Homework

- Draw a transition graph for the DFA M defined as:

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{a , b\}$

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

Initial state = q_0

$F = \{q_2\}$

Homework

- Draw a transition graph for

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

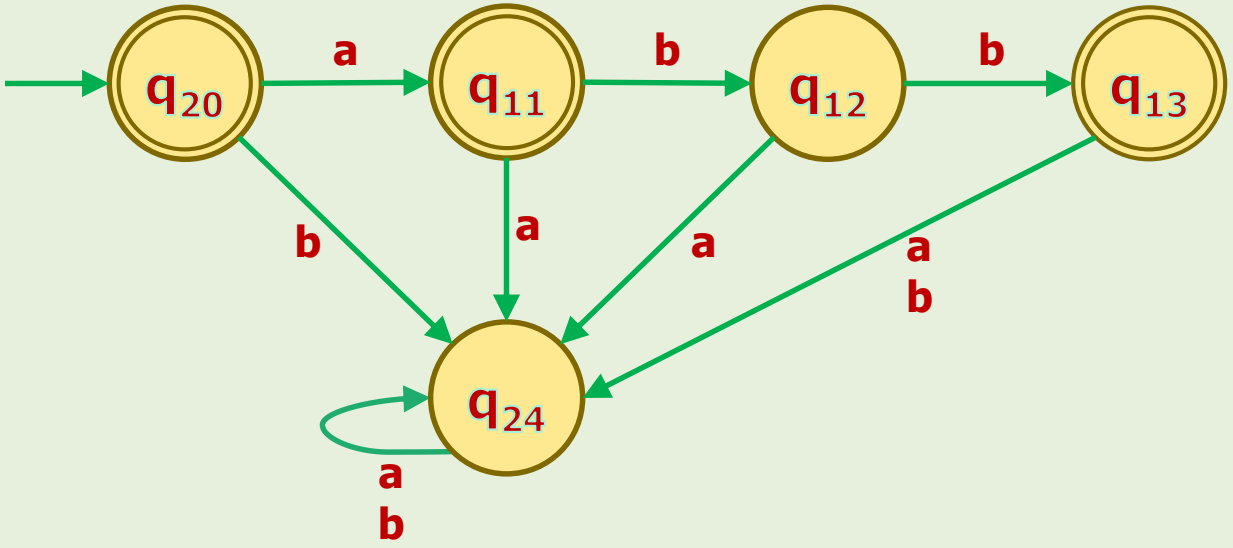
$$\delta: \begin{cases} \delta(q_0, 0) = q_0 \\ \delta(q_1, 0) = q_0 \\ \delta(q_2, 0) = q_2 \\ \delta(q_0, 1) = q_1 \\ \delta(q_1, 1) = q_2 \\ \delta(q_2, 1) = q_1 \end{cases}$$

- Which strings from the following set are accepted?

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

Homework

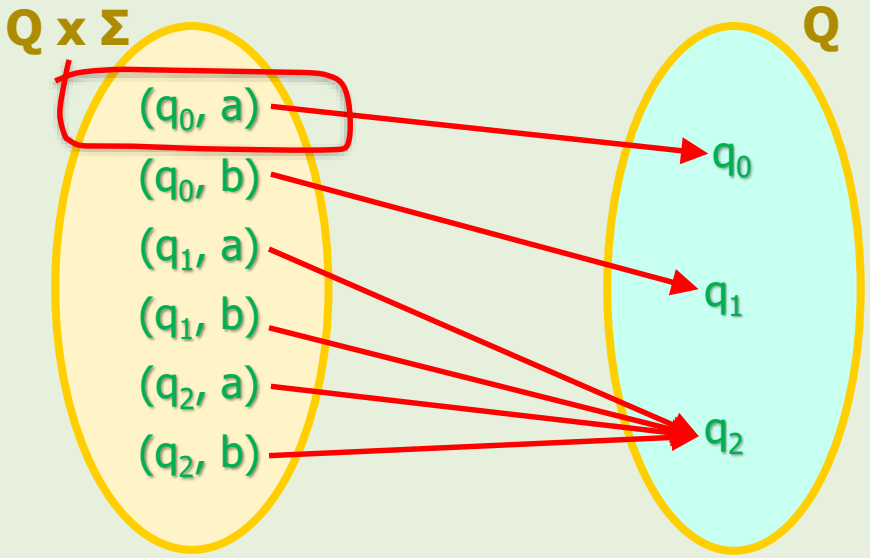
- Write all elements of the following transition graph.
- Q = ?
- Σ = ?
- δ = ?
- q_0 = ?
- F = ?

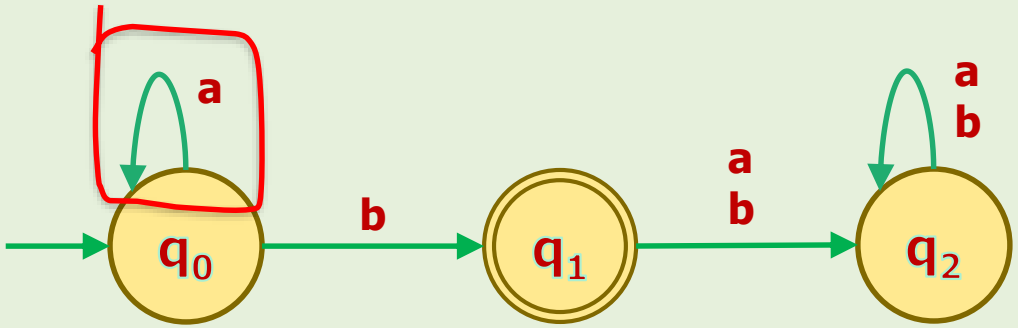


Why Total Function

Example 19 (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.

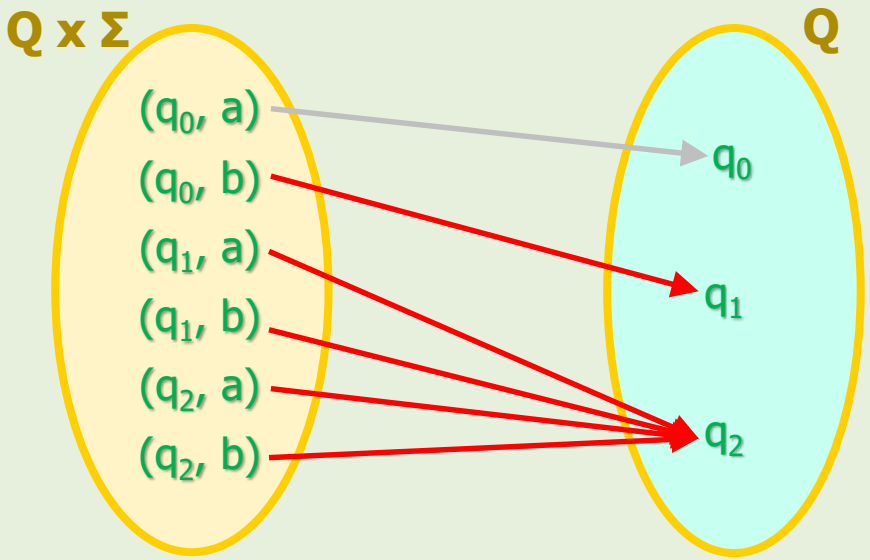


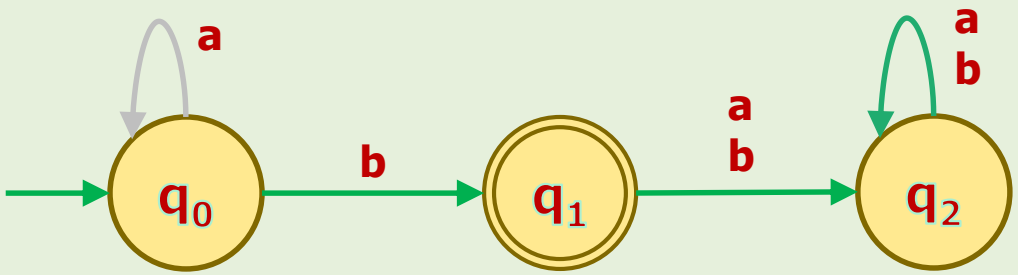


Why Total Function

Example 19 (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 what to do?
- It "hangs"!!
- So, δ must be total function!

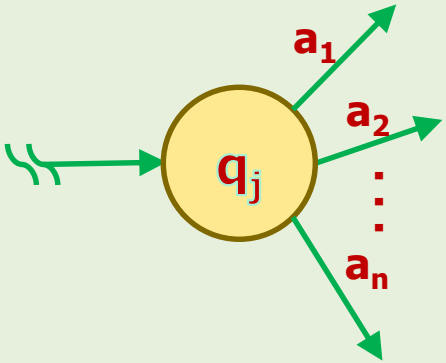




Total Function from Different Angle

- DFAs must know where to go at any timeframe.

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



Conclusion

- δ being total function is "DFAs' constraint".
- It means that for DFAs:
  At any timeframe, there must be one and only one transition.

Transition Table

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

Example 20

- Represent the following transition function by a transition table.

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

Alphabet

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

Current States

Range: Next State

Associated Language to DFAs

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

Definition

- The associated language to a DFA M is the set of ALL STRINGS that M 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 $L(M_1) = L(M_2)$ over Σ.
  - M_1 and M_2 are equivalent iff their associated languages are equal.

- 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

- "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, as AN OBERSERVER, are given a known DFA's configuration at timeframe n.
- Can you predict its configuration at timeframe n+1?
- Yes, we, as AN OBERSERVER, can predict its behavior with 100% certainty.
- Because at any timeframe, there is one and only one transition.
- In other words, there is no randomness in DFAs behavior!

What is Determinism?

Definition

- A machine is called deterministic iff at any timeframe, there is NO MORE THAN ONE transition.
  - This definition is satisfied if the number of transitions is zero or one.

- Note that for DFAs, it is one and only one but for other deterministic machines, it could be zero or one. (Will be covered later.)

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