San José State University
Department of Computer Science

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

Deterministic Finite Automata

(Part 2)

Lecture 07

Day 07/31

CS 154

Formal Languages and Computability

Spring 2018

Agenda of Day 07

- Summary of Lecture 06
- Quiz 2
- Lecture 07: Teaching ...
  - Deterministic Finite Automata (Part 2)
- Prepare Your Development Environment

Summary of Lecture 06: We learned ...

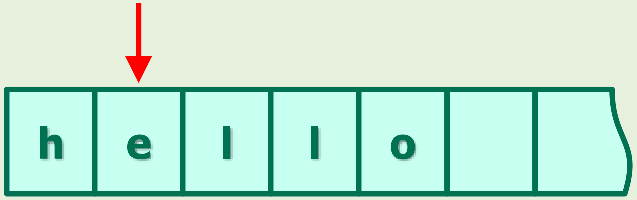
Deterministic Finite Automata

- We started constructing machines that understand formal languages.
- We'll construct several classes of machines in this course.
- Each class has different power.
- DFAs are the simplest ones.

Building blocks of DFAs:

- Input tape, Control unit, Output

Input Tape



- The input tape is read-only.
- The read-head moves from left-to-right.
  - We cannot move the head back.
- Reading, consuming and scanning have the same meaning.

Any question?

Summary of Lecture 06: We learned ...

Control Unit

- Represented by transition graph.
- The number of states is finite.
- There is only one initial state.
- There can be zero or more accepting state (aka final state).

Output

- The output has two messages:
  - Accept (aka: understood, recognized, Yes)
  - Reject (aka: not understood, not recognized, No)

- DFAs configuration (aka snapshot) is the combination of the following data:
  - Timeframe of the clock
  - Input string
  - Position of the read-head
  - Current state of the transition graph

- Starting configuration is ...

Any question?

Summary of Lecture 06: We learned ...

When DFAs halt

- When all input symbols are consumed.

$$h \leftrightarrow c$$

How DFAs accept a 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, accepting a string condition is:

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

How DFAs reject a 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?

NAME	Alan M. Turing		
SUBJECT	CS 154	TEST NO.	2
DATE	02/15/2018	PERIOD	1 , 2 , 3

TEST RECORD	
PART 1	123
PART 2	
TOTAL	

Your **list #** goes here!

Quiz 2

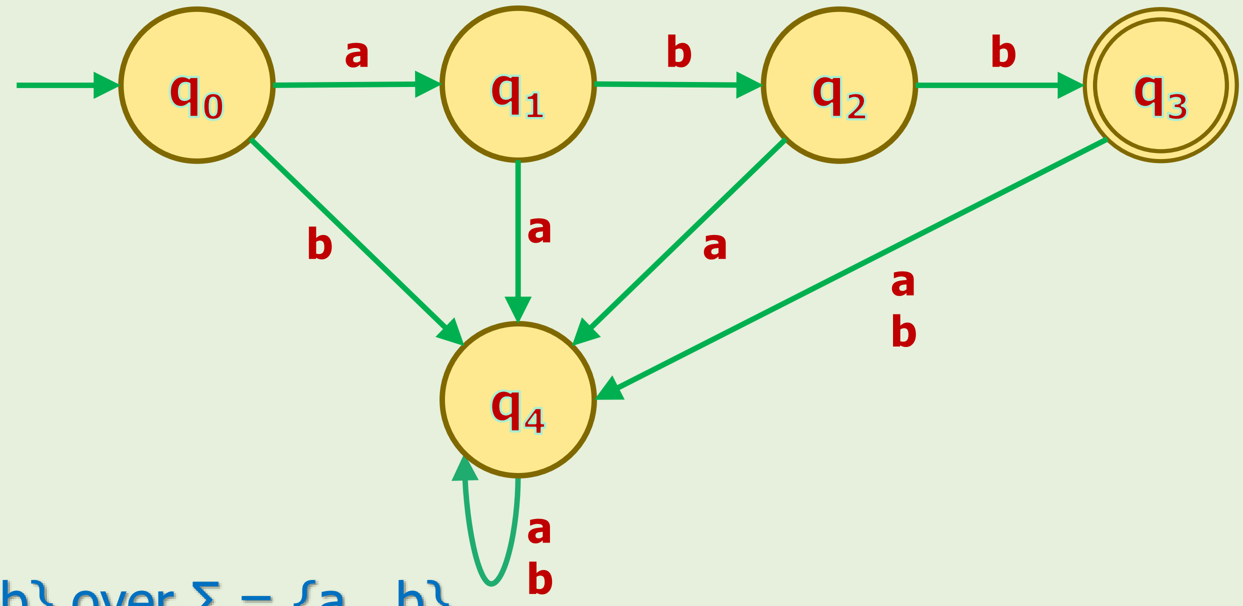
Use Scantron

Analysis Examples

Analysis Examples

Example 9

- What language does the following DFA accept?
- What is Σ?

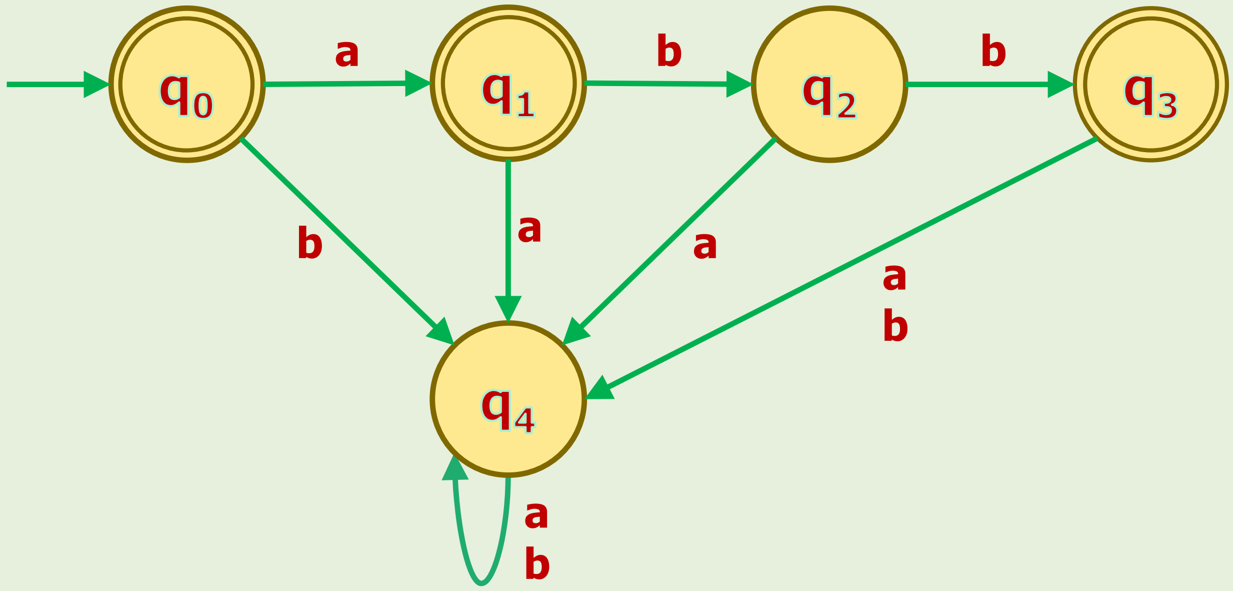


- L = {abb} over Σ = {a , b}

Analysis Examples

Example 10

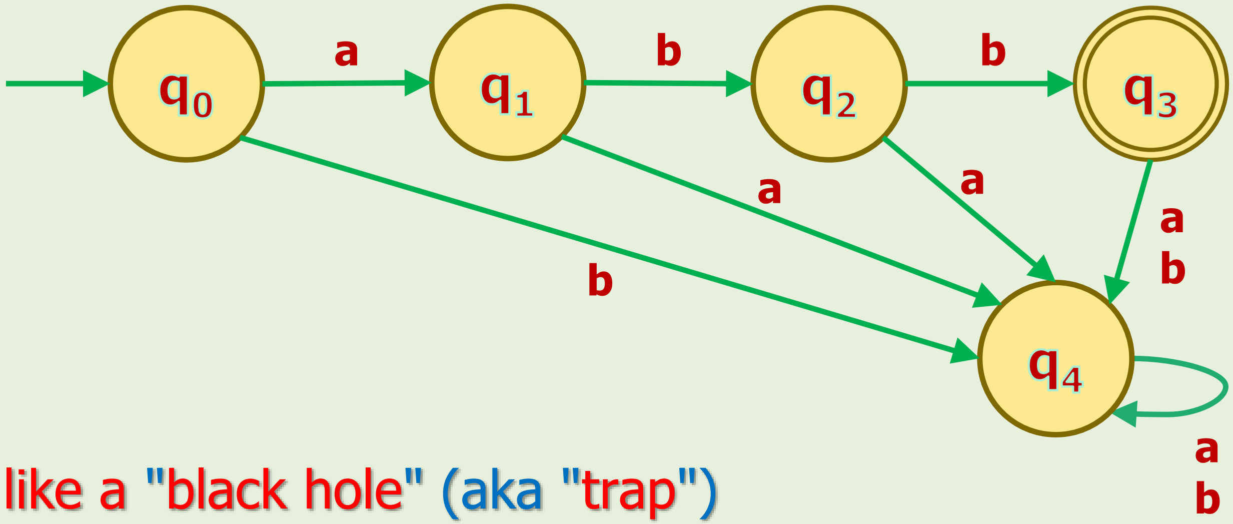
- What language does the following DFA accept?



- L = {λ, a, abb} over Σ = {a , b}

Notes

1. We don't need to show the "output" and the "clock" anymore.
2. The role of q_4 in the previous examples:

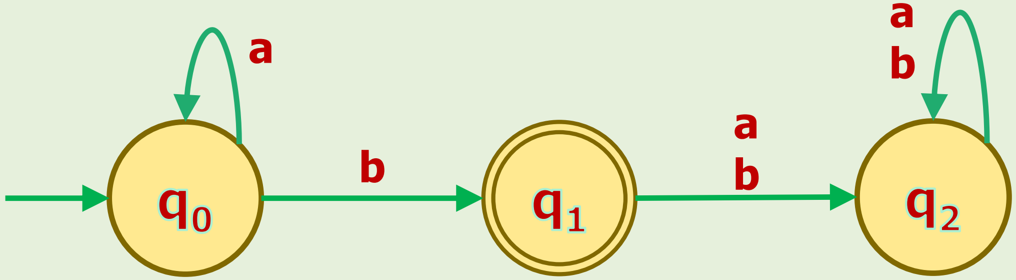


- q_4 acts like a "black hole" (aka "trap")
  because if the machine transits to q_4,
  it gets stuck in it and would not be able to transit to other states.
- Sometimes I call it "hell"!
- Now we understand the meaning of "Go to hell!"!

Analysis Examples

Example 11

- What language does the following DFA accept?
  - Represent the language by set builder method.



Design Examples

Design Examples

Example 12

- Let L ={ad , ada , adam} over Σ = {a , d , m}.
- Design a DFA to accept L.

Design Important Notes

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

   It accepts all strings of **L**.

   AND

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

- Have we designed the previous example correctly?

Design Important Notes

2. To test your machine, both groups accepted strings and rejected strings, should be picked from Σ*.

   - We are not allowed to input strings from outside Σ*.
   - Otherwise the behavior of the machine is not guaranteed.

A million dollar question!

- Can you ever claim that your design works fine?
- Never, because Σ* is infinite!
- So, theoretically every design has potential bugs!

Design Examples

Example 13

- Let L be the set of strings over Σ = {a , b} starting with prefix ab.

  1. Write a set-builder for L.
  2. Design a DFA to accept L.

Design Examples

Example 14

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

1. Write a set-builder for L.
2. Design a DFA to accept L.

Design Examples

Example 15

- Let L be the set of strings over Σ = {1} that contains even unary numbers.

1. Write a set-builder for L.
2. Design a DFA to accept L.

Design Examples: DFA over $\Sigma = \{a , b\}$

Example 16: Empty Language

- $L = \{ \}$

Example 17: All Strings

- $L = \Sigma^*$

Example 18

- $L = \{\lambda\}$

Homework

- $L = \Sigma^+$

Design Examples

Example 19

- Let $L = \{a^n b^m : n \geq 0, m \geq 0\}$
- Design a DFA to accept L.

Design Examples

Example 20

- Let L be the set of strings over Σ = {0 , 1} consisting substring 001.
  1. Write a set-builder for L.
  2. Design a DFA to accept L.

Design Examples

Example 21

- Let L be the set of strings over Σ = {0 , 1} consisting all strings except substring 001.
  - Design a DFA to accept L.

- What is the difference between this language and the previous one?

How to Construct a DFA to Accept $\overline{L}$

Algorithm

- Construct a DFA to accept L.
- Turn all accepting states to regular states.
- Turn all regular states to accepting states.

Homework: DFA Design

- For each of the following languages over Σ = {a , b}:
  1. Write a set-builder.
  2. Design a DFA for each.

  - The set of strings that contains exactly one 'a'
  - The set of strings that contains at least one 'a'
  - The set of strings ending with prefix ab
  - All strings with no more than three 'a's
  - All strings with at least three 'a's

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

Prepare Your Development Environment

JFLAP (Java Formal Language and Automata Package)

- We'll use JFLAP tools in this course to develop and test our automata.
- **Official website**: http://www.jflap.org/
- **Download it from Canvas**: Files/Misc/JFLAP.jar

- Or Download the stable version 7.0 (thick version with SVG May 15, 2011) from: http://www.jflap.org/jflaptmp/
- Tutorial: http://www.jflap.org/tutorial/

JFLAP Demo

JFLAP

Basic Features

- Creating states
- Defining a state as initial state or final state
- Creating transitions
- Deleting
- Shift-Enter to create multiple transitions
- Multiple running (testing your design)
- Debugging: step-by-state
- Saving machines (xml file)

Other Features

- Changing state's name or label
- Adding comment
- Changing edges shape
- Selecting multiple objects and moving
- Zoom-in and out

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