CSE 105
THEORY OF COMPUTATION

"Winter" 2018

http://cseweb.ucsd.edu/classes/wi18/cse105-ab/

Definitions

pp. 13, 14, 64-65

- **Alphabet** non-empty finite set (Σ)
- **Symbol** element of alphabet
  character.
- **String** over Σ finite list of symbols from Σ
  word
- **Language** over Σ set of strings over Σ
- **Regular expression** over Σ

  syntactic expression built up recursively
- **Language** described by a regular expression

  set of strings matching pattern given by r.e.

Examples

1. $R = a$, where $a \in \Sigma$
2. $R = \varepsilon$
3. $R = \emptyset$
4. $R = (R_1 \cup R_2)$
5. $\boxed{R = (R_1 \circ R_2)}$
6. (R_1^*)

$(\{0\} \cup \{1\}) \cup \{1\}$

- L ((0 U 1)U1) = { 0, 1 }
- L ($(\Sigma\Sigma\Sigma\Sigma)^*$) = { }
- L ($1^*\emptyset 0$) = { } $= \emptyset$

$\{1\}^* \circ \emptyset = \emptyset$

$$\Sigma = \{0,1\}$$

$$\Sigma \circ \Sigma \circ \Sigma \circ \Sigma = \{0000, 0001, 0010, 0011, \ldots\}$$

$$|\Sigma\Sigma\Sigma\Sigma| = 16.$$

$$(\Sigma\Sigma\Sigma\Sigma)^* = \{\varepsilon, 0000, 0001, 0010, \ldots, 00101001, \ldots$$

$$= \{w \in \{0,1\}^* \mid |w| = 4k \text{ for } k \in \mathbb{Z}, k \geq 0\}$$

$$R_1 \circ R_2 = \{xy \mid x \in R_1, y \in R_2\}$$

A more complicated example

Which of the following strings is **not** in the language described by

$\{\varepsilon, 00, 0000, 000000, \ldots\}$

$$\Big(\underbrace{\big((00)^*(11) \big)}_{\{11,\,0011,\,000011,\,\ldots\}} \cup 01 \Big)^*$$

$\{01, 11, 0011, 00011, \ldots\}$

$= \{ \varepsilon, \ldots$

A. 00

B. 01

C. 1101

D. ε

E. ~~I don't know~~

And another …

aabaabb

Let L be the language over {a,b} described by the regular expression

ababε $((a \cup \varnothing)\, b^*)^*$

Which of the following is **not** true about L?

A. Some strings in L have equal numbers of a's and b's
B. L contains the string aaaaaa
C. a's never follow b's in any string in L
D. L can also be represented by the regular expression $(ab^*)^*$
E. More than one of the above.

Regular expressions in practice

- **Compilers**: first phase of compiling transforms Strings to Tokens keywords, operators, identifiers, literals
  - One regular expression for each token type

- **Other software tools**: grep, Perl, Python, Java, Ruby, …

Regular expressions

Which regular expressions describe languages that include the string a ?

A. a*b*
B. a(ba)*b
C. a* U b*
D. (aaa)*
E. (ε U a)b = {b, ab}

Frequency: AB

To change your remote frequency
1. Press and hold power button until flashing
2. Enter two-letter code
3. Checkmark / green light indicates success

In practice

- How do computers check if a string is in the language described by a regular expression?

```
grep 'password' /etc/passwd
```

Pre-class reading

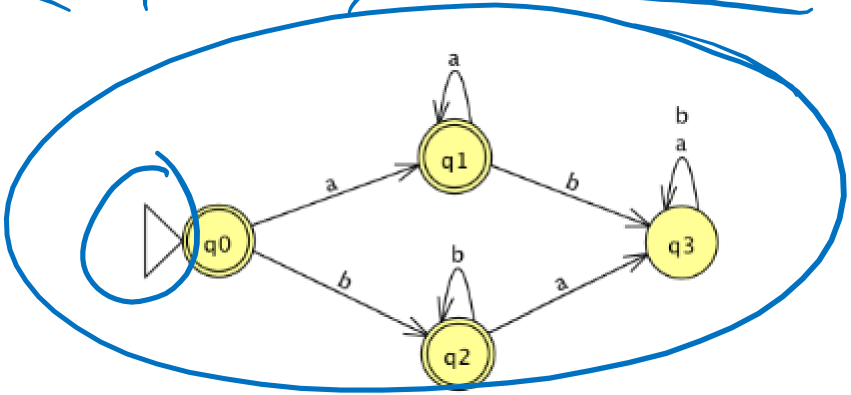
- Tracing the computation of a finite automata using its state diagram.
- Formal definition of finite automaton.

From the website:

DFA *Reading* Sec 1.1: Figure 1.4 (p. 34), Definition 1.5 (p. 35)
Optional extra practice: Chapter 1 Exercise # 1, 2, 3

Deterministic Finite Automaton

Sipser p. 34



Start state(s)? state with an arrow coming from nowhere

Accept state(s)? Double circle

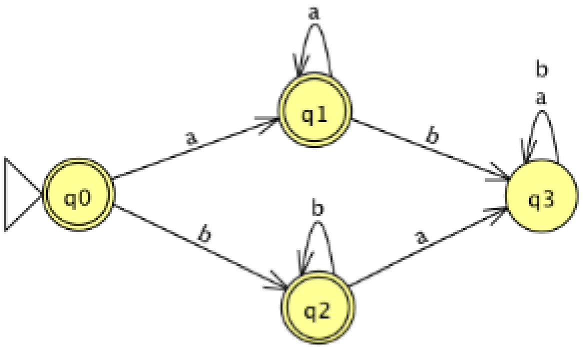
Transitions? edges.

Deterministic Finite Automaton

Sipser p. 34, 40

Input: string

Output: T/F
yes/no



Computation of the machine on an input string

Sequence of states in the machine, starting with the initial state, determined by transitions of the machine as it reads additional input symbols.

Deterministic Finite Automaton

Sipser p. 34, 40

Computation of the machine on an input string

Sequence of states in the machine, starting with the initial state, determined by transitions of the machine as it reads additional input symbols.

Machine accepts the input string if after reading the entire string you end up at an accept state

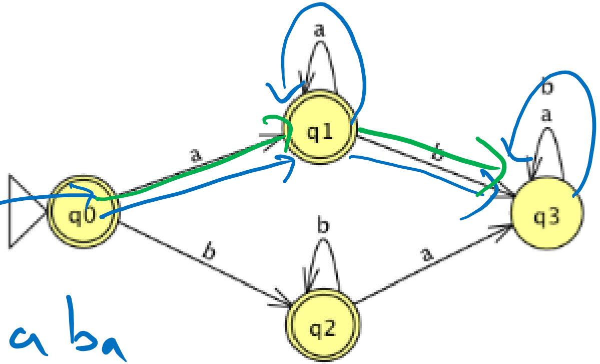
Machine rejects the input string if

The **language recognized by the machine** is the set of strings it accepts.

Examples

aaba

1.



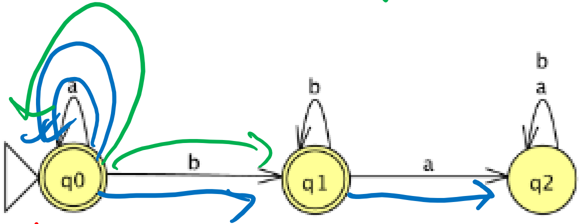
accepts ε

rejects aaba

rejects ab

2.

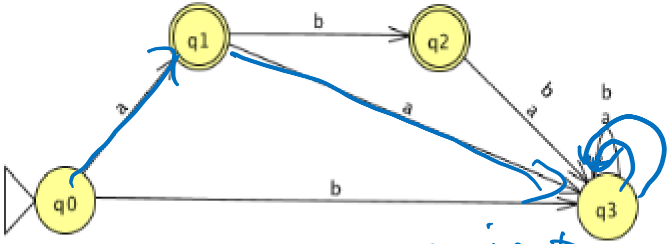
accepts ab



rejects aaba

accepts ε

3.



rejects ε

rejects

Which of these automata recognize **the same** language?

A. All of them.
B. 1. and 2.
C. 1. and 3.
D. 2. and 3.
E. None of them (they each recognize different languages).

Deterministic finite automaton

Sipser p. 35 Def 1.5

A **finite automaton** is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where

1. Q is a finite set called the states
2. Σ is a finite set called the alphabet
3. $\delta : Q \times \Sigma \to Q$ is the transition function
4. $q_0 \in Q$ is the start state
5. $F \subseteq Q$ is the set of accept states.

No circles and arrows, same information!

Deterministic finite automaton

Sipser p. 35 Def 1.5

A **finite automaton** is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where

1. Q is a finite set called the states
2. Σ is a finite set called the alphabet
3. $\delta : Q \times \Sigma \to Q$ is the transition function
4. $q_0 \in Q$ is the start state
5. $F \subseteq Q$ is the set of accept states.

Can there be more than one **start state** in a finite automaton?

A. Yes, because of line 4.
B. No, because of line 4.
C. I don't know

Deterministic finite automaton

Sipser p. 35 Def 1.5

A **finite automaton** is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where

1. Q is a finite set called the states
2. Σ is a finite set called the alphabet
3. $\delta : Q \times \Sigma \to Q$ is the transition function
4. $q_0 \in Q$ is the start state
5. $F \subseteq Q$ is the set of accept states.

How many outgoing arrows from each state?

A. May be different number at each state.
B. Must be 2.
C. Must be |Q|.
D. Must be |Σ|
E. I don't know.

An example

Define M = $(\{q1, q2, q3, q4\}, \{a, b\}, \delta, q1, \{q4\})$ where the function δ is specified by its table of values:

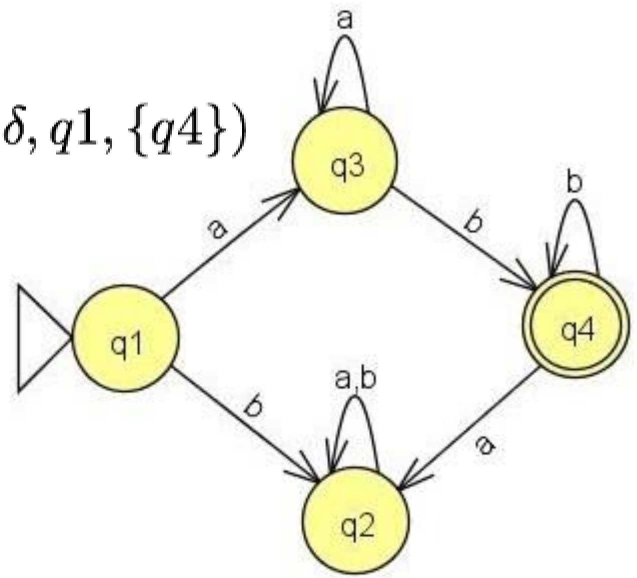
Input in Q x Σ	Output in Q
(q1,a)	q3
(q2,a)	q2
(q3,a)	q3
(q4,a)	q2

Input in Q x Σ	Output in Q
(q1,b)	q2
(q2,b)	q2
(q3,b)	q4
(q4,b)	q4

Draw the state diagram for the DFA with this formal definition.

An example

$$(\{q1, q2, q3, q4\}, \{a, b\}, \delta, q1, \{q4\})$$



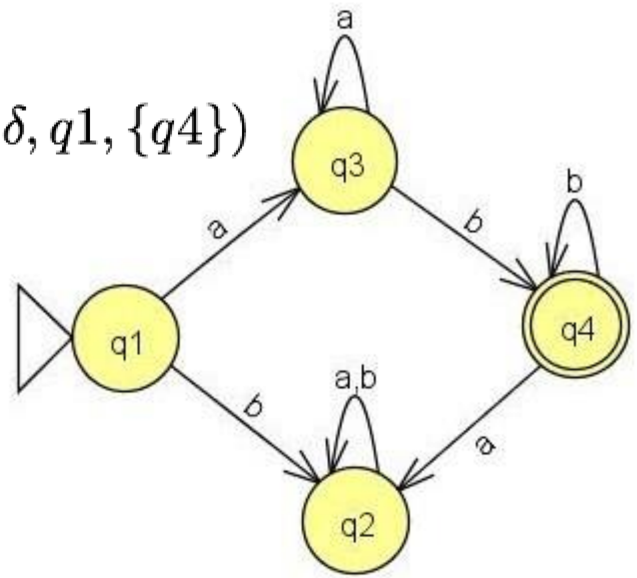
What's an example of a

- length 1 string accepted by this DFA?
- length 1 string rejected by this DFA?

- length 2 string accepted by this DFA?
- length 2 string rejected by this DFA?

An example

$(\{q1, q2, q3, q4\}, \{a, b\}, \delta, q1, \{q4\})$



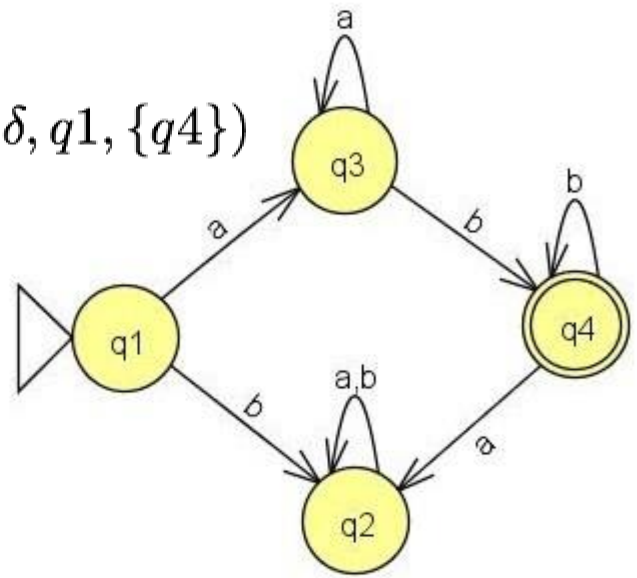
What's the best description of the language recognized by this DFA?

A. Starts with b and ends with a or b
B. Starts with a and ends with a or b
C. a's followed by b's
D. More than one of the above
E. I don't know.

and using set notation?

An example

$$(\{q1, q2, q3, q4\}, \{a, b\}, \delta, q1, \{q4\})$$

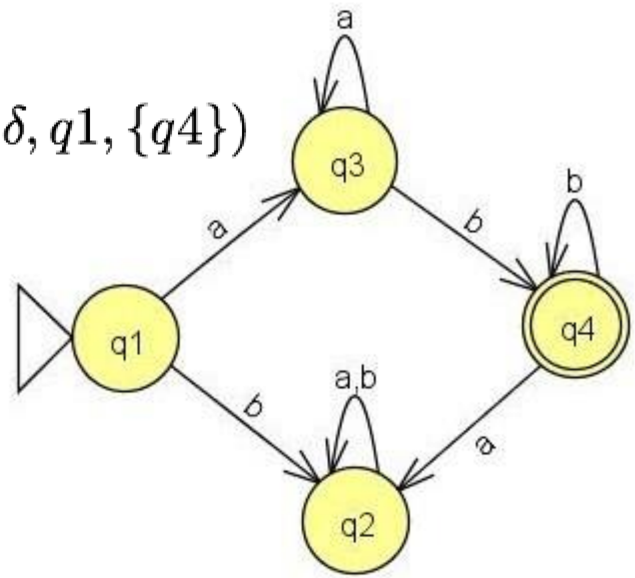


This DFA recognizes
the language of all strings
of the form a's followed by b's

i.e. $\{ a^n b^k \mid n,k \geq 1\}$

An example

$(\{q1, q2, q3, q4\}, \{a, b\}, \delta, q1, \{q4\})$



$\{ a^n b^k \mid n,k \geq 1\}$

Is this the same as the language described by

A. a*b*
B. a(ba)*b
C. a* U b*
D. a(ab)*b
E. (ε U a)b

For next time

- Individual Homework 0 **due Saturday**
  - Set up course tools: *Gradescope*, Piazza
  - Read all the questions + relevant examples in the book
  - Start working ☺
  - *Review CSE 20 / Math 109 / CSE 21 / Sipser Ch 0 as needed.*
- Discussion section Thursday **"Breadth"**
  - Regular expressions and DFAs

Pre class-reading for Friday: Example 1.21