CSE 105
THEORY OF COMPUTATION

Spring 2018

http://cseweb.ucsd.edu/classes/sp18/cse105-ab/
Definitions

- **Alphabet**  
  non-empty finite set

- **Symbol**  
  element of alphabet

- **String** over \( \Sigma \)  
  finite list of symbols from \( \Sigma \)

- **Language** over \( \Sigma \)  
  set of strings over \( \Sigma \)

- **Regular expression** over \( \Sigma \)  
  \( R \)  
  syntactic expression built up recursively

- **Language** described by a regular expression \( L(R) \)  
  set of strings matching pattern given by r.e.
Definitions

For two sets of strings $A$, $B$ over the same alphabet

**Union** \[ A \cup B = \{ x \mid x \in A \text{ or } x \in B \} \]

**Concatenation** \[ A \circ B = \{ xy \mid x \in A \text{ and } y \in B \} \]

**Kleene Star** \[ A^* = \{ x_1 x_2 \cdots x_k \mid k \geq 0 \text{ and each } x_i \in A \} \]
Regular expressions

Which regular expression describes a language that includes the string a ?

A. $a^*b^*$
B. $a(ba)^*b$
C. $a^* U b^*$
D. $(aaa)^*$
E. $(\varepsilon U a)b$

L($a^*b^*$) = \{ $a^ib^j$ / $i, j \geq 0$ \}
L($(aaa)^*$) = \{ $a^i$ / $i \geq 0$ \}
L($a^* U b^*$) = \{ $a^i$ / $i \geq 0$ \} U \{ $b^i$ / $i \geq 0$ \}
In practice

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

`grep 'password' /etc/passwd`
First model

• Text processing
grep, regexp

• Natural language processing

• Hardware design
Moore machines, Mealy machines: CSE 140

• Controllers / Robots
SPIS!
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 (DFA)

\[ \Sigma = \{a, b\} \]

Start state(s)? \[ q_0 \]
Accept state(s)? \[ q_0, q_1, q_2 \]
Transitions?

\[ \begin{align*}
q_0 \quad a &\rightarrow q_1 \\
q_0 \quad b &\rightarrow q_2 \\
q_1 \quad b &\rightarrow q_2 \\
q_1 \quad a &\rightarrow q_0 \\
q_2 \quad a &\rightarrow q_0 \\
q_2 \quad b &\rightarrow q_1 
\end{align*} \]
Deterministic Finite Automaton

Input: String

Output: 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

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 computation of the machine on string ends in an accept state.

Machine rejects the input string if computation of the machine on strings ends in a non-accept state.

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

\[ M \text{ is DFA }, \quad L(M) = \{ w \in \Sigma^* | M \text{ accepts } w \} \]
Examples

1. \( L(M_1) = \{ a^* \cup b^* \} \) comp of \( M_1 \) on \( \varepsilon \): \( q_0, q_1 \) rejects \( q_3 \)

2. \( L(M_2) = \{ w \mid a \text{ is a substring of } w \} \)

3. \( L(M_3) = \{ \varepsilon, ab \} \) comp of \( M_3 \) on \( \varepsilon \): \( q_0 \) rejects

Which of these automata accept each string \( a^n b^m \), where \( n, m \geq 0 \)?

A. \(1\)
B. \(2\)
C. \(3\)
D. More than one of them.
E. None of the them.
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.

Pre class-reading for Friday: Example 1.21
A finite automaton is a 5-tuple \((Q, \Sigma, \delta, q_0, F)\) where

1. \(Q\) is a finite set called the states
2. \(\Sigma\) is a finite set called the alphabet
3. \(\delta : Q \times \Sigma \rightarrow Q\) is the transition function
4. \(q_0 \in Q\) is the start state
5. \(F \subseteq Q\) is the set of accept states.
A finite automaton is a 5-tuple \((Q, \Sigma, \delta, q_0, F)\) where

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