CSE 105
THEORY OF COMPUTATION

Fall 2016

http://cseweb.ucsd.edu/classes/fa16/cse105-abc/
Today's learning goals Sipser Ch 1.1

• Use and design a finite automaton via its
  - Formal definition
  - state diagram
  - haskell implementation

• Identify the strings and languages accepted by a
given finite automaton

• Design a finite automaton which accepts a given
language
Review: Finite automaton

• Input: finite **string** over a fixed **alphabet**
• Output: "accept" or "reject"

**Computation**: sequence of states traversed by the machine

**Language** of the machine is the set of strings it accepts
Finite automaton

- Input: finite string over a fixed alphabet
- Output: "accept" or "reject"

Does this DFA accept the string 10110?

A. Yes
B. No
C. I don't know
Finite automaton

- Input: finite string over a fixed alphabet
- Output: "accept" or "reject"

Does this DFA accept the **empty string**?

A. Yes  
B. No  
C. I don't know
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.
Finite automaton

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.

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
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

Can there be zero many accept states?

A. Yes, in which case the language is empty.
B. Yes, in which case the language is all strings.
C. No, because of line 5.
D. I don't know.
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. $\delta : Q \times \Sigma \to Q$ is the transition function
4. $q_0 \in Q$ is the start state
5. $F \subseteq Q$ is the set of accept states

Can one state have two different transitions labelled with the same symbol going out of it?

A. Yes, because of 2.
B. Yes, because of 3.
C. No, because of 2.
D. No, because of 3.
E. I don't know.
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.

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 \(|\Sigma|\)
E. I don't know.
Function computed by a DFA

- The mathematical definition \((Q, \Sigma, \delta, q_0, F)\) only capture the syntax of a DFA
- We also need to define how each DFA specifies a function \(f_M: \Sigma \rightarrow \{\text{True, False}\}\)
- Helper function: \(\delta^*: Q \times \Sigma^* \rightarrow Q\)
  - \(\delta^*(q, \text{""}) = q\)
  - \(\delta^*(q, (aw)) = \delta^*(\delta(q, a), w)\) [for any \(a\) in \(\Sigma\) and \(w\) in \(\Sigma^*\)]
  - Equivalent (but less efficient) definition: \(\delta^*(q, (wa)) = \delta(\delta^*(q, w), a)\)
Implementing DFAs in Haskell
An example

\((\{q_1, q_2, q_3, q_4\}, \{a, b\}, \delta, q_1, \{q_4\})\)

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

A. Start 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.
An example

\[(\{q_1, q_2, q_3, q_4\}, \{a, b\}, \delta, q_1, \{q_4\})\]

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

i.e. \[\{ a^n b^k \mid n, k \geq 1 \}\]
Another example

What is the best description of language recognized by this automaton?

A. \{ a^n b^k \mid n,k \geq 1 \}
B. \{ a^n b^k \mid n \geq 1, b \geq 0 \}
C. \{ awb \mid w \text{ in } \{a,b\}^* \}
D. \{ aw \mid w \text{ in } \{a,b\}^* \}
E. I don't know
Regular languages \hspace{1cm} \text{sipser p. 35 def 1.5}

• If $A$ is the set of strings that DFA $M$ recognizes (accepts)
  • We say $A$ is the \text{language} of $M$
  • We write $L(M) = A$
  • We conclude that $A$ is \text{regular} because...

A language is \text{regular} if there is some finite automaton that recognizes \text{exactly} it.
Vocabulary review

From CSE20, etc. See Chapter 0

- \{ a,b,c,d,e \}  The set whose elements are a,b,c,d,e
- \{ a,b \}^*  The set of finite strings over a,b
  - Includes empty string
  - Includes a, aa, aaa
  - Includes b, bb, bbb
  - Includes ab, ababab, aaaaaaabb
  - Does not include infinite sequences of a's and b's
  - Has infinitely many different elements

- | abababab | = 6  The length of the string abababab is 6
- | { a,b,c,d,e } | = 5  The size of the set \{a,b,c,d,e\} is 5
Specifying an automaton

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

What's the best representation of \( \delta \) for this DFA?

A. \( q1 \rightarrow b, \ q1 \rightarrow a, \ q2 \rightarrow a, \ q2 \rightarrow b, \ q3 \rightarrow a,b. \)
B. \( \{(q1,b,q1),(q1,a,q2),(q2,a,q2), (q2,b,q3),(q3,a,q3),(q3,b,q3)\} \)
C. \( \delta(b) = \text{same}, \delta(a) = \text{change} \)
D. No description other than the state diagram (circles & arrows) is possible.
E. I don't know.
What state(s) should be in $F$ so that the language of this machine is

$$\{ w \mid \text{ab is a substring of } w \}$$

A. $\{q2\}$  
B. $\{q3\}$  
C. $\{q1,q2\}$  
D. $\{q1,q3\}$  
E. I don't know.
Specifying an automaton

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

What state(s) should be in F so that the language of this machine is \{ w | b's never occur after a's in w \}?

A. \{q2\}
B. \{q3\}
C. \{q1,q2\}
D. \{q1,q3\}
E. I don't know.
**Terminology: summary**

- **Alphabet**: nonempty finite set of **symbols**
- **String** over an alphabet: finite sequence of symbols
- **Language** over an alphabet: some set of strings

- **DFA** over an alphabet: deterministic finite automaton
  - Input: finite string over a fixed alphabet
  - Output: "accept" or "reject"
  - \( L(M) = \{ w \mid M \text{ accepts } w \} \)
- **Regular language**
  language that is \( L(M) \) for some DFA \( M \)

---

**Diagram notes**

- **Start state**: (triangle/arrow)
- **Accept state**: (double circle)
Next time

• Homework 1  **due Wednesday!**
  • Set up course tools: *Gradescope*, Piazza, JFLAP, haskell
  • Complete the assignment and submit by 11:59pm

• Closure Properties of Regular Languages
  • What languages are regular?
  • Can regular languages be combined together?