CS 301 - Lecture 1
Deterministic Finite Automata
Fall 2008

Course Material

• Website: http://www.cs.colostate.edu/~cs301
  – Syllabus, Outline, Grading Policies
  – Homework and Slides
• Instructor: Dan Massey
  – Office hours: 2-3pm Tues and Wed
  – Email: massey@cs.colostate.edu
• Teaching Assistant: William Springer
  – Email: wmspring@cs.colostate.edu

Grading and Policies

• Grading
  – 10% Homework
  – 30% Quizes
  – 30% Midterm
  – 30% Final
• Grading Policy
  – No credit for late homework. No exceptions.
  – No credit for missed exams. No make-up exams.

Workload

• Weekly Reading Assignments
  – Course will cover most of both books
• Weekly Homework Assignments
  – Available from course website by Friday morning
  – Due in lecture the following Thursday
• Exams
  – Midterm + final (comprehensive)
  – In class, closed book, one double-sided cheat sheet allowed
Administrative

- You are responsible for knowing course, department, and university policies
  - READ THE SYLLABUS
  - PROVIDES LINKS TO DEPT POLICIES
- Plagiarism – see definition
  - http://writing.colostate.edu/guides/teaching/plagiarism/
- Conflict Resolution web site
  - http://www.conflictresolution.colostate.edu/

About These Slides

- Slides Originally Developed by Prof. Costas Busch (2004)
  - Many thanks to Prof. Busch for developing the original slide set.
- Adapted with permission by Prof. Dan Massey (Spring 2007)
  - Subsequent modifications

Languages

- A language is a set of strings

String: A sequence of letters

- Examples: “cat”, “dog”, “house”, ...
- Defined over an alphabet:
  \[ \Sigma = \{a, b, c, \ldots, z\} \]

Alphabets and Strings

- We will use small alphabets: \( \Sigma = \{a, b\} \)
- Strings
  \[ a \]
  \[ ab \]
  \[ abba \]
  \[ baba \]
  \[ aaabbaabab \]
  \[ u = ab \]
  \[ v = bbbaaa \]
  \[ w = abba \]
String Concatenation

\[ w = a_1a_2\cdots a_n \]
\[ v = b_1b_2\cdots b_m \]
\[ wv = a_1a_2\cdots a_n b_1b_2\cdots b_m \]

\[ abba \quad \text{Concatenation} \Rightarrow \quad abababaaa \]

\[ bbbaaa \]

String Reverse

\[ w = a_1a_2\cdots a_n \]
\[ w^R = a_n\cdots a_2a_1 \]

\[ ababaaabbb \quad \text{Reverse} \Rightarrow \quad bbaaababa \]

String Length

\[ w = a_1a_2\cdots a_n \]
\[ \text{Length} = |w| = n \]

\[ |abba| = 4 \]
\[ |aa| = 2 \]
\[ |a| = 1 \]

Length and Concatenation

\[ |uv| = |u| + |v| \]
\[ u = aab, \quad |u| = 3 \]
\[ v = abaab, \quad |v| = 5 \]

\[ |uv| = |aababaab| = 8 \]
\[ |uv| = |u| + |v| = 3 + 5 = 8 \]

Does this prove the formula is true?
Can you prove the formula is true?
Empty String

• A string with no letters: \( \lambda \)
  \[ |\lambda| = 0 \]

• Observations:
  \( \lambda w = w \lambda = w \)

\( \lambda abba = abba \lambda = abba \)

Substring

• Subsequence of consecutive characters

<table>
<thead>
<tr>
<th>String</th>
<th>Substring</th>
</tr>
</thead>
<tbody>
<tr>
<td>( abbab )</td>
<td>( ab )</td>
</tr>
<tr>
<td>( abbab )</td>
<td>( abba )</td>
</tr>
<tr>
<td>( abbab )</td>
<td>( b )</td>
</tr>
<tr>
<td>( abbab )</td>
<td>( bbab )</td>
</tr>
</tbody>
</table>

Prefix and Suffix

<table>
<thead>
<tr>
<th>prefix</th>
<th>suffix</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda )</td>
<td>( abab )</td>
</tr>
<tr>
<td>( a )</td>
<td>( bbab )</td>
</tr>
<tr>
<td>( ab )</td>
<td>( bab )</td>
</tr>
<tr>
<td>( abb )</td>
<td>( ab )</td>
</tr>
<tr>
<td>( abba )</td>
<td>( b )</td>
</tr>
<tr>
<td>( abbab )</td>
<td>( \lambda )</td>
</tr>
</tbody>
</table>

More Concatenation

\[ w^n = \underbrace{ww \cdots w}_{n} \]
\[ w^0 = \lambda \]

\( (abba)^2 = abbaabba \)
\( (abba)^0 = \lambda \)
### The * Operation

Given an alphabet $\Sigma$,

\[ \Sigma^* \text{ is the set of all strings over the alphabet} \]

\[ \Sigma = \{a,b\} \]
\[ \Sigma^* = \{\lambda,a,b,aa,ab,ba,bb,aaa,aab,\ldots\} \]

### The + Operation

Given an alphabet $\Sigma$,

\[ \Sigma^+ \text{ is the set of all strings over the alphabet minus } \lambda \]

\[ \Sigma = \{a,b\} \]
\[ \Sigma^* = \{\lambda,a,b,aa,ab,ba,bb,aaa,aab,\ldots\} \]
\[ \Sigma^+ = \{a,b,aa,ab,ba,bb,aaa,aab,\ldots\} \]

### Languages

A Language is any subset of $\Sigma^*$

Given an alphabet:

\[ \Sigma = \{a,b\} \]
\[ \Sigma^* = \{\lambda,a,b,aa,ab,ba,bb,aaa,\ldots\} \]

Some Languages over this alphabet include:

\[ \{\lambda\} \]
\[ \{a,aa,aab\} \]
\[ \{\lambda,abba,baba,aa,ab,aaaaaa\} \]

### Languages, Sets, and Notations

Sets

\[ \emptyset = \{\} \neq \{\lambda\} \]

Set size

\[ |\{\}| = |\emptyset| = 0 \]

Set size

\[ |\{\lambda\}| = 1 \]

String length

\[ |\lambda| = 0 \]
An Infinite Language

\[ L = \{ a^n b^n : n \geq 0 \} \]

\[ \lambda \quad \begin{cases} a \in L & \text{and} \quad a b \notin L \\ ab \quad \in L & \text{and} \quad a b b \notin L \\ a a b b \quad \in L & \text{and} \quad a a a b b b \notin L \end{cases} \]

What is the alphabet associated with this language?

Languages and Operations

A language is a set and set operations apply

\[ \{ a, ab, aaaa \} \cup \{ bb, ab \} = \{ a, ab, bb, aaaa \} \]
\[ \{ a, ab, aaaa \} \cap \{ bb, ab \} = \{ ab \} \]
\[ \{ a, ab, aaaa \} - \{ bb, ab \} = \{ a, aaaa \} \]

The Complement of a Language

Intuitively, all strings not in the language. More precisely:

\[ L = \Sigma^* - L \]
\[ \{ a, ba \} = \{ \lambda, b, aa, ab, bb, aaa, \ldots \} \]

Is \( ccc \) in the complement of this language?

Reverse

\[ L^R = \{ w^R : w \in L \} \]
\[ \{ ab, aab, baba \}^R = \{ ba, baa, abab \} \]
\[ L = \{ a^n b^n : n \geq 0 \} \]
\[ L^R = \{ b^n a^n : n \geq 0 \} \]
Concatenation

\[ L_1 L_2 = \{xy : x \in L_1, y \in L_2\} \]

\[ \{a, ab, ba\} \{b, aa\} \]

\[ = \{ab, aaa, abb, abaa, bab, baaa\} \]

More Concatenation

\[ L^n = L L \cdots L \]

\[ L^0 = \{\lambda\} \]

\[ \{a, bba, aaa\}^0 = \{\lambda\} \]

\[ \{a, b\}^3 = \{a, b\} \{a, b\} \{a, b\} = \{aaa, aab, aba, abb, baa, bab, bba, bbb\} \]

Star-Closure (Kleene *)

\[ L^* = L^0 \cup L^1 \cup L^2 \cdots \]

\[ \{a, bb\}^* = \{\lambda, a, bb, aa, abb, bba, bbbb, aaa, aabb, abba, abbb, \ldots\} \]

Is Our Notation Sufficient?

Describe a language by listing all the strings in the language
But this only works for finite languages
Describe a language by listing some pattern such as:
\[ L = \{a^n b^n : n \geq 0\} \]
\[ L^2 = \{a^n b^n a^m b^m : n, m \geq 0\} \]
\[ aabbaaabb \in L^2 \]

But this has limits as well....
Consider the language of all valid English sentences
Automata

Finite Automaton

Input
String

Control unit
Storage

Output

Finite Accepter

Input
String

Output

“Accept” or “Reject”

Transition Graph

Abba -Finite Accepter

Initial state
transition
final state “accept”
Initial Configuration

Input String

Reading the Input

Input String
String Rejection

Input finished

Output: "accept"
Input finished

The Empty String
Would it be possible to accept the empty string?

Another Example

Output: "reject"
Formalities

\[ M = (Q, \Sigma, \delta, q_0, F) \]

- Deterministic Finite Acceptor (DFA)
- \( Q \) : set of states
- \( \Sigma \) : input alphabet
- \( \delta \) : transition function
- \( q_0 \) : initial state
- \( F \) : set of final states
Input Alphabet $\Sigma$

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

Set of States $Q$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$

Initial State $q_0$

$F = \{q_4\}$
Transition Function $\delta$

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

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_5$

$\delta(q_2, b) = q_3$
Transition Function $\delta$

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>$a$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$q_1$</td>
<td>$q_5$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$q_5$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$q_5$</td>
<td>$q_3$</td>
</tr>
<tr>
<td>$q_3$</td>
<td>$q_4$</td>
<td>$q_5$</td>
</tr>
<tr>
<td>$q_4$</td>
<td>$q_5$</td>
<td>$q_5$</td>
</tr>
</tbody>
</table>

Extended Transition Function $\delta^*$

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

$\delta^*(q_0, ab) = q_2$

$\delta^*(q_0, abba) = q_4$
Observation: There is a walk from $q$ to $q'$ with label $w$

\[
\delta^* (q, w) = q'
\]

\[
\delta^* (q_0, abbbaa) = q_5
\]

Example: There is a walk from $q_0$ to $q_5$ with label $abbbaa$

\[
\delta^* (q_0, abbbaa) = q_5
\]

Recursive Definition

\[
\delta^* (q, \lambda) = q
\]

\[
\delta^* (q, w \sigma) = \delta (\delta^* (q, w), \sigma)
\]

\[
\delta^* (q, w) = q'
\]

\[
\delta (q, \sigma) = q'
\]

\[
\delta^* (q, w \sigma) = \delta (\delta^* (q, w), \sigma)
\]
Languages Accepted by DFAs

- Take DFA $M$

- Definition:
  - The language $L(M)$ contains all input strings accepted by $M$
  
  $L(M) = \{ \text{strings that drive } M \text{ to a final state} \}$

What’s Next

- Read
  - Linz Chapter 1, Chapter 2.1 - 2.3
  - JFLAP Startup, Chapter 1, 2.1

- Next Lecture Topics from Chapter 2.1 and 2.2
  - Regular Languages
  - NonDeterministic Finite Automata

- Homework 1
  - Due Thursday