Deterministic Finite-State Automata

Deepak D’Souza

Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

12 August 2013
Outline

1. Introduction

2. Formal Definitions and Notation
Example DFA 1

DFA for “Odd number of a’s”

- How a DFA works.
Example DFA 1

DFA for “Odd number of a’s”

- How a DFA works.
- Each state represents a property of the input string read so far:
  - State $e$: Number of a’s seen is even.
  - State $o$: Number of a’s seen is odd.
Example DFA 2

DFA for “Contains the substring $abb$”

Each state represents a property of the input string read so far:
Example DFA 2

DFA for “Contains the substring \textit{abb}”

Each state represents a property of the input string read so far:

- **State $\epsilon$:** Not seen \textit{abb} and no suffix in \textit{a} or \textit{ab}.
- **State \textit{a}:** Not seen \textit{abb} and has suffix \textit{a}.
- **State \textit{ab}:** Not seen \textit{abb} and has suffix \textit{ab}.
- **State \textit{abb}:** Seen \textit{abb}.
Example DFA 3

Accept strings over \( \{0, 1\} \) which satisfy even parity in length 4 blocks.

- Accept “0101 1010”
- Reject “0101 1011”

DFA for “Even parity checker”
Example DFA 4

Accept strings over \{a, b, /, *\} which don’t end inside a C-style comment.

- Scan from left to right till first “/*” is encountered; from there to next “*/” is first comment; and so on.
- Accept “ab/ * aaa */ abba” and “ab/ * aa/ * aa */ bb */”.
- Reject “ab/ * aaa*” and “ab/ * aa/ * aa */ bb/ * a”.

DFA for “C-comment tracker”

```
pbc / *
pec out a a, / / *a
```
Example DFA 4

Accept strings over \( \{a, b, /, \ast\} \) which don’t end inside a C-style comment.

- Scan from left to right till first “/\*” is encountered; from there to next “\*/” is first comment; and so on.
- Accept “ab/ * aaa* /abba” and “ab/ * aa/ * aa* /bb* /”.
- Reject “ab/ * aaa*” and “ab/ * aa/ * aa * /bb/ * a”.

DFA for “C-comment tracker”
An *alphabet* is finite set of set of symbols or “letters”. Eg. $A = \{a, b, c\}$, $\Sigma = \{0, 1\}$.

A *string* or *word* over an alphabet $A$ is a finite sequence of letters from $A$. Eg. $aba$ is string over $\{a, b, c\}$.

Empty string denoted by $\epsilon$.

Set of all strings over $A$ denoted by $A^*$.

What is the “size” or “cardinality” of $A^*$?
Definitions and notation

- An **alphabet** is a finite set of symbols or "letters". Eg. $A = \{a, b, c\}$, $\Sigma = \{0, 1\}$.
- A **string** or **word** over an alphabet $A$ is a finite sequence of letters from $A$. Eg. $aaba$ is a string over $\{a, b, c\}$.
- Empty string denoted by $\epsilon$.
- Set of all strings over $A$ denoted by $A^*$.
  - What is the "size" or "cardinality" of $A^*$?
  - Infinite but **Countable**: Can enumerate in **lexicographic** order:
    \[
    \epsilon, \ a, \ b, \ c, \ aa, \ ab, \ldots
    \]
Definitions and notation

- An alphabet is finite set of set of symbols or “letters”. Eg. \( A = \{ a, b, c \}, \Sigma = \{ 0, 1 \} \).
- A string or word over an alphabet \( A \) is a finite sequence of letters from \( A \). Eg. \( aaba \) is string over \( \{ a, b, c \} \).
- Empty string denoted by \( \epsilon \).
- Set of all strings over \( A \) denoted by \( A^* \).
  - What is the “size” or “cardinality” of \( A^* \)?
  - Infinite but \textbf{Countable}: Can enumerate in lexicographic order:
    \[ \epsilon, \ a, \ b, \ c, \ aa, \ ab, \ldots \]
- Operation of concatenation on words: String \( u \) followed by string \( v \): written \( u \cdot v \) or simply \( uv \).
  - Eg. \( aabb \cdot aaa = aabbaaa \).
A language over an alphabet $A$ is a set of strings over $A$. Eg. for $A = \{a, b, c\}$:

- $L = \{abc, aaba\}$.
- $L_1 = \{\epsilon, b, aa, bb, aab, aba, baa, bbb, \ldots\}$.
- $L_2 = \{\}$.
- $L_3 = \{\epsilon\}$.

How many languages are there over a given alphabet $A$?
A **language** over an alphabet $A$ is a set of strings over $A$. Eg. for $A = \{a, b, c\}$:

- $L = \{abc, aaba\}$.
- $L_1 = \{\epsilon, b, aa, bb, aab, aba, baa, bbb, \ldots\}$.
- $L_2 = \{\}$.
- $L_3 = \{\epsilon\}$.

How many languages are there over a given alphabet $A$?

- **Uncountably infinite**
- Use a diagonalization argument:

<table>
<thead>
<tr>
<th></th>
<th>$\epsilon$</th>
<th>a</th>
<th>b</th>
<th>aa</th>
<th>ab</th>
<th>ba</th>
<th>bb</th>
<th>aaa</th>
<th>aab</th>
<th>aba</th>
<th>abb</th>
<th>bbb</th>
<th>\ldots</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_0$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
</tr>
<tr>
<td>$L_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
</tr>
<tr>
<td>$L_2$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>\ldots</td>
<td></td>
</tr>
<tr>
<td>$L_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
</tr>
<tr>
<td>$L_4$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
</tr>
<tr>
<td>$L_5$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>\ldots</td>
</tr>
<tr>
<td>$L_6$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
</tr>
<tr>
<td>$L_7$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>\ldots</td>
</tr>
<tr>
<td>\ldots</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Definitions and notation: Languages

- Concatenation of languages:
  \[ L_1 \cdot L_2 = \{ u \cdot v \mid u \in L_1, \ v \in L_2 \}. \]

  Eg. \( \{abc, aaba\} \cdot \{\epsilon, a, bb\} = \{abc, aaba, abca, aabaa, abcbb, aababb\} \).
A *Deterministic Finite-State Automaton* \( A \) over an alphabet \( A \) is a structure of the form

\[
(Q, s, \delta, F)
\]

where

- \( Q \) is a finite set of “states”
- \( s \in Q \) is the “start” state
- \( \delta : Q \times A \rightarrow Q \) is the “transition function.”
- \( F \subseteq Q \) is the set of “final” states.
A *Deterministic Finite-State Automaton* \( A \) over an alphabet \( A \) is a structure of the form

\[
(Q, s, \delta, F)
\]

where

- \( Q \) is a finite set of “states”
- \( s \in Q \) is the “start” state
- \( \delta : Q \times A \rightarrow Q \) is the “transition function.”
- \( F \subseteq Q \) is the set of “final” states.

**Example of “Odd a’s” DFA:**

Here: \( Q = \{ e, o \} \), \( s = e \), \( F = \{ o \} \),

and \( \delta \) is given by:

\[
\begin{align*}
\delta(e, a) &= o, \\
\delta(e, b) &= e, \\
\delta(o, a) &= e, \\
\delta(o, b) &= o.
\end{align*}
\]
 Definitions and notation: Language accepted by a DFA

- \( \hat{\delta} \) tells us how the DFA \( \mathcal{A} \) behaves on a given word \( u \).
- Define \( \hat{\delta} : Q \times A^* \rightarrow Q \) as
  
  \[
  \begin{align*}
  \hat{\delta}(q, \epsilon) &= q \\
  \hat{\delta}(q, w \cdot a) &= \delta(\hat{\delta}(q, w), a).
  \end{align*}
  \]
- Language accepted by \( \mathcal{A} \), denoted \( L(\mathcal{A}) \), is defined as:
  
  \[
  L(\mathcal{A}) = \{ w \in A^* \mid \hat{\delta}(s, w) \in F \}.
  \]
- Eg. For \( \mathcal{A} = \text{DFA for “Odd a’s”} \),
  
  \[
  L(\mathcal{A}) = \{ a, ab, ba, aaa, abb, bab, bba, \ldots \}.
  \]
A language $L \subseteq A^*$ is called \textit{regular} if there is a DFA $\mathcal{A}$ over $A$ such that $L(\mathcal{A}) = L$.

Examples of regular languages: “Odd a’s”, “strings that don’t end inside a C-style comment”, $\{\}$, any \textit{finite} language.

Are there non-regular languages?
A language $L \subseteq A^*$ is called *regular* if there is a DFA $A$ over $A$ such that $L(A) = L$.

Examples of regular languages: “Odd a’s”, “strings that don’t end inside a C-style comment”, $\{\}$, any *finite* language.

Are there non-regular languages?
- Yes, uncountably many, since Reg is only countable while class of all languages is uncountable.