Deterministic Finite-State Automata

CS 301:Theory of Computation

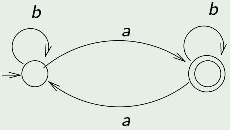
Sumesh Divakaran
Department of Computer Science and Engineering
College of Engineering Trivandrum

5^{th} August 2019

Outline

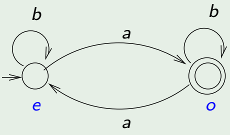
Example DFA 1

DFA for ?



Example DFA 1

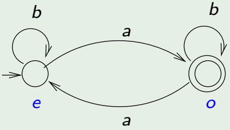
DFA for ?



DFA for "Odd number of a's"

Example DFA 1

DFA for ?

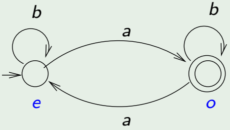


DFA for "Odd number of a's"

How a DFA works?

Example DFA 1

DFA for ?



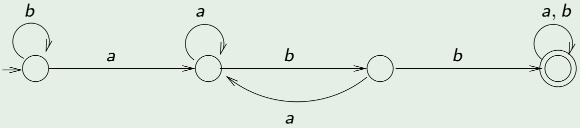
DFA for "Odd number of a's"

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.

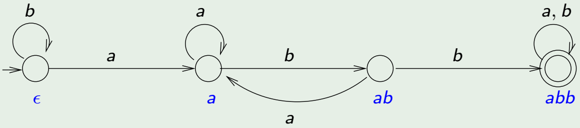
Example DFA 2

DFA for ?



Example DFA 2

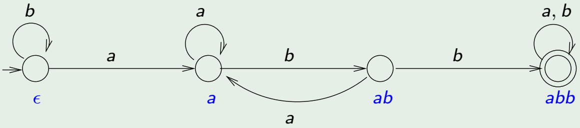
DFA for ?



DFA for "Contains the substring *abb*"

Example DFA 2

DFA for ?



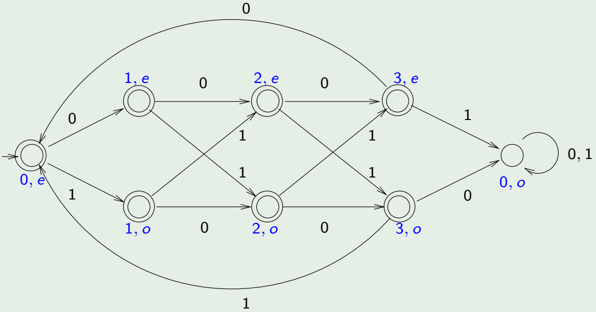
DFA for "Contains the substring *abb*"

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

- State ϵ: Not seen *abb* and no suffix in *a* or *ab*.
- State *a*: Not seen *abb* and has suffix *a*.
- State *ab*: Not seen *abb* and has suffix *ab*.
- State *abb*: Seen *abb*.

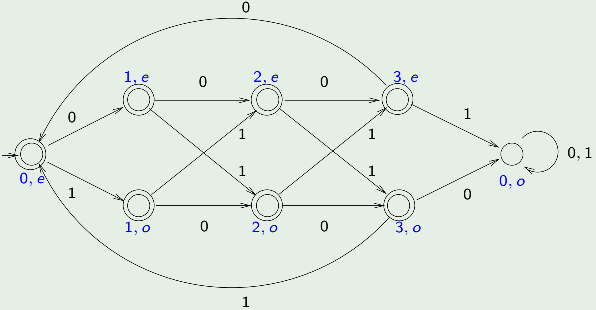
Example DFA 3

DFA for ?



Example DFA 3

DFA for ?



DFA for "Even parity checker"

Accept strings over $\{0, 1\}$ which have even parity in each length 4 block.

- Accept "$0101 \cdot 1010$"
- Reject "$0101 \cdot 1011$"

Example DFA 4

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

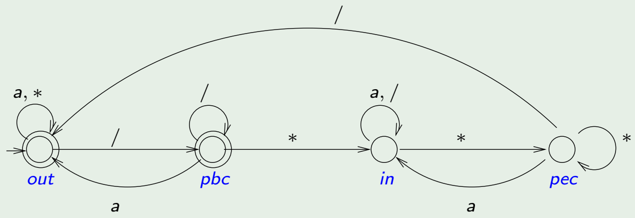
- Scan from left to right till first "/*" is encountered; from there to next "*/" is first comment; and so on.
- Accept "$ab/*aaa*/abba$" and "$ab/*aa/*aa*/bb*/$".
- Reject "$ab/*aaa*$" and "$ab/*aa/*aa*/bb/*a$".

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"



Definitions and notation

- An alphabet is a finite set of symbols or "letters". Eg. $\Sigma = \{a, b, c\}$ or $\Sigma = \{0, 1\}$.
- A string or word over an alphabet Σ is a finite sequence of letters from Σ. Eg. "$aaba$" is string over $\{a, b, c\}$.
- Empty string (the string of length zero) is denoted by ϵ.
- Set of all strings over Σ denoted by Σ^*.
    - What is the "size" or "cardinality" of Σ^*?

Definitions and notation

- An alphabet is a finite set of symbols or "letters". Eg. $\Sigma = \{a, b, c\}$ or $\Sigma = \{0, 1\}$.
- A string or word over an alphabet Σ is a finite sequence of letters from Σ. Eg. "$aaba$" is string over $\{a, b, c\}$.
- Empty string (the string of length zero) is denoted by ϵ.
- Set of all strings over Σ denoted by Σ^*.
  - What is the "size" or "cardinality" of Σ^*?
  - Infinite but Countable: Can enumerate in lexicographic order:

$$\epsilon,\ a,\ b,\ c,\ aa,\ ab, \ldots$$

.

Definitions and notation

- An alphabet is a finite set of symbols or "letters". Eg. $\Sigma = \{a, b, c\}$ or $\Sigma = \{0, 1\}$.
- A string or word over an alphabet Σ is a finite sequence of letters from Σ. Eg. "$aaba$" is string over $\{a, b, c\}$.
- Empty string (the string of length zero) is denoted by ϵ.
- Set of all strings over Σ denoted by Σ^*.
  - What is the "size" or "cardinality" of Σ^*?
  - Infinite but 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. $aabb \cdot aaa = aabbaaa$.

Definitions and notation: Languages

- A language over an alphabet Σ is a set of strings over Σ. Eg. for $\Sigma = \{a, b, c\}$:
  - $L = \{abc, aaba\}$.
  - $L_1 = \{\epsilon, b, aa, bb, aab, aba, baa, bbb, \ldots\}$.
  - $L_2 = \{\}$.
  - $L_3 = \{\epsilon\}$.
- How many languages are there over a given alphabet Σ?

Definitions and notation: Languages

- A language over an alphabet Σ is a set of strings over Σ. Eg. for $\Sigma = \{a, b, c\}$:
    - $L = \{abc, aaba\}$.
    - $L_1 = \{\epsilon, b, aa, bb, aab, aba, baa, bbb, \ldots\}$.
    - $L_2 = \{\}$.
    - $L_3 = \{\epsilon\}$.
- How many languages are there over a given alphabet Σ?
    - Uncountably infinite
    - Use a diagonalization argument:

	ϵ	a	b	aa	ab	ba	bb	aaa	aab	aba	abb	bbb	...
L_0	0	1	0	0	0	1	1	0	0	0	0	0	...
L_1	0	0	0	0	0	0	0	0	0	0	0	0	...
L_2	1	1	0	1	0	1	1	0	0	1	0	1	...
L_3	0	0	0	0	0	0	0	0	0	0	0	0	...
L_4	0	1	0	0	0	1	1	0	0	0	0	0	...
L_5	1	1	0	1	0	1	1	0	0	1	0	1	...
L_6	0	1	0	0	0	1	1	0	0	0	0	0	...
L_7	0	0	0	0	0	0	1	0	0	0	1	0	...
⋮													

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. $\{abc, aaba\} \cdot \{\epsilon, a, bb\} =$ $\{abc, aaba, abca, aabaa, abcbb, aababb\}$.

Definitions and notation: DFA

A Deterministic Finite-State Automaton $\mathcal{A}$ over an alphabet Σ

Definitions and notation: DFA

A Deterministic Finite-State Automaton $\mathcal{A}$ over an alphabet $\Sigma$is a structure of the form

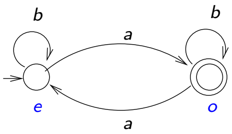
$$(Q, s, \delta, F)$$

where

- Q is a finite set of "states"
- $s \in Q$ is the "start" state
- $\delta : Q \times \Sigma \rightarrow Q$ 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 δ is given by:

$$\begin{aligned} \delta(e, a) &= o, \\ \delta(e, b) &= e, \\ \delta(o, a) &= e, \\ \delta(o, b) &= o. \end{aligned}$$



Definitions and notation: Language accepted by a DFA

- $\widehat{\delta}$ tells us how the DFA $\mathcal{A}$ behaves on a given word u.
- Define $\widehat{\delta} : Q \times \Sigma^* \to Q$ as
  - $\widehat{\delta}(q, \epsilon) = q$
  - $\widehat{\delta}(q, w \cdot a) = \delta(\widehat{\delta}(q, w), a)$.
- Language *accepted* by $\mathcal{A}$, denoted $L(\mathcal{A})$, is defined as:

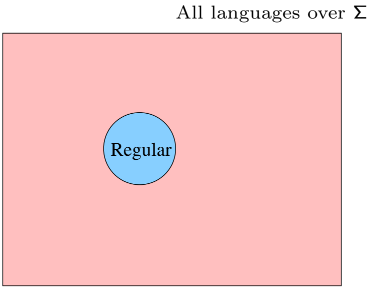
$$L(\mathcal{A}) = \{w \in \Sigma^* \mid \widehat{\delta}(s, w) \in F\}.$$

- Eg. For $\mathcal{A}$ = DFA for "Odd a's",

$$L(\mathcal{A}) = \{a, ab, ba, aaa, abb, bab, bba, \ldots\}.$$

Regular Languages

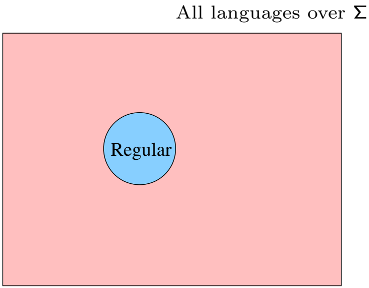
- A language $L \subseteq \Sigma^*$ is called *regular* if there is a DFA $\mathcal{A}$ over Σ such that $L(\mathcal{A}) = L$.
- Examples of regular languages: "Odd a's", "strings that don't end inside a C-style comment", $\{\}$, any finite language.



- Are there non-regular languages?

Regular Languages

- A language $L \subseteq \Sigma^*$ is called *regular* if there is a DFA $\mathcal{A}$ over Σ such that $L(\mathcal{A}) = L$.
- 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.