UMC205: Automata and Computability

Naman Mishra

January 2024

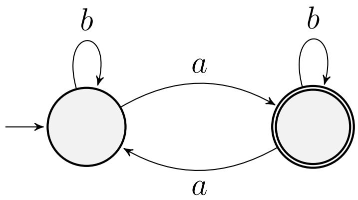
Contents

1 Languages **1**

2 Deterministic Finite-State Automata **3**

Lecture 01: Tue 02 Jan '24

A *finite state automata* which accepts the language with an odd number of a's



A language is *regular* if it is accepted by a finite state automata. It may be deterministic or non-deterministic, the set of languages accepted by both are the same.

A *pushdown automata* is a finite state automata with a stack.

A *context-free grammar* is a language that is accepted by a pushdown automata.

Lecture 02: Thu 04 Jan '24

1 Languages

Definition 1.1. An *alphabet* is a non-empty finite set of symbols or "letters".

A *string* or *word* over an alphabet A is a finite sequence of letters from A. Equivalently, a string is a map from a prefix (possibly empty) of $\mathbb{N}$ to A. The length of a string s, notated $\#s$, is the cardinality of its domain. The empty string is denoted ϵ.

The set of all strings over A is denoted A^*.

Example. $A = \{a, b, c\}$ and $\Sigma = \{0, 1\}$ are both alphabets. $aaba$ is a string over $A = \{a, b, c\}$.

Proposition 1.2. Let A be an alphabet. Then A^* is countably infinite.

Proof. Let $n = \#A$. Let $f\colon A \to \{1, \ldots, n\}$ be a numbering of A. Replacing each letter in a string with its number gives a representation of the string as a natural number in base $n + 1$. This gives an injection $A^* \to \mathbb{N}$ and so A^* is countable. Infiniteness is obvious.

Alternatively, consider the strings in their Lexicographic order. $\square$

Definition 1.3 (Language)**.** A *language* over an alphabet A is a subset of A^*.

Example. Let $A = \{a, b, c\}$. Then $\{abc, aaba\}$, $\{\epsilon, b, aa, bb, aab, aba, bbb, \ldots\}$, $\{\epsilon\}$, $\{\}$ are all languages over A.

Definition 1.4 (Concatenation)**.** Let u, v be strings over an alphabet A. Then $u \cdot v$ or simply uv is the string obtained by appending v to the end of u.

For two languages L_1, L_2 over A, define their concatenation

$$L_1 \cdot L_2 \coloneqq \{uv \mid u \in L_1, v \in L_2\}.$$

We will also write ua where u is a string and a is a letter to mean u concatenated with the string of length 1 consisting of the letter a.

Definition 1.5 (Lexicographic order)**.** Let $(A, <)$ be a totally ordered alphabet. We say $u < v$ for $u, v \in A^*$ if either $\#u < \#v$ or $\#u = \#v$ and $u = pxu'$, $v = pyv'$ for some $p, u', v' \in A^*$ and $x, y \in A$ with $x < y$.

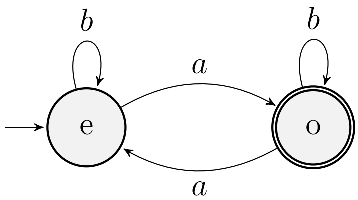
This is called the *lexicographic order* on A^*.

Proposition 1.6. Let A be an alphabet. Then the set of all languages over A is uncountable.

Proof. Diagonilization. Suppose there is an enumeration L of all languages over A. Let s be an enumeration of A^*. Then define $L' = \{s \in A^* \mid s \notin L_s\}$. Then L' is a language over A that is not in L. □

Definition 1.7 (Concatenations)**.** Let L_1, L_2 be languages over an alphabet A. Then $L_1 \cdot L_2$ is the language $\{uv \mid u \in L_1, v \in L_2\}$.

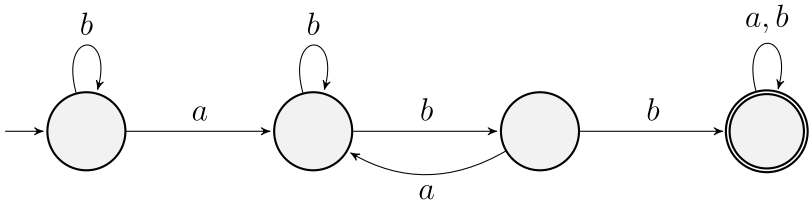
2 Deterministic Finite-State Automata



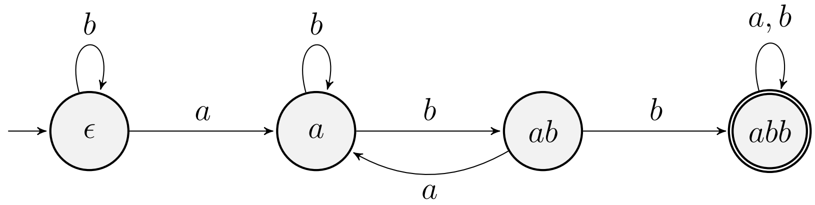
Each state represents a property of the input string read so far. State e is the start state, and state o is the only accepting state.

In this case, State e corresponds to an even number of a's read, and State o corresponds to an odd number of a's read. This can be proven by induction to conclude that the automaton accepts the language $\{w \in \{a, b\}^* \mid \#_a(w)\}$.

Example. Let $A = \{a, b\}$. Consider the DFA



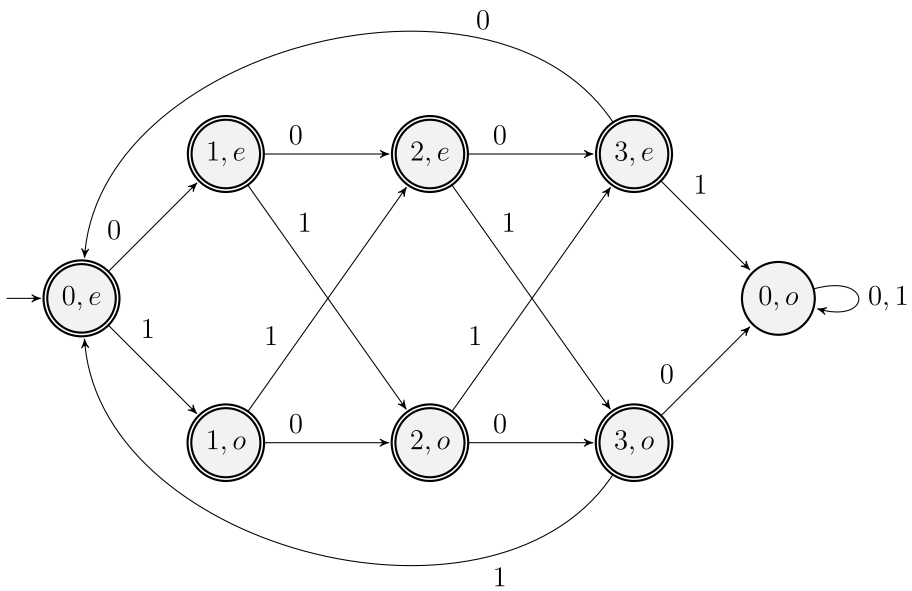
We label the nodes as ϵ, a, ab and abb



and consider the property corresponding to each state.

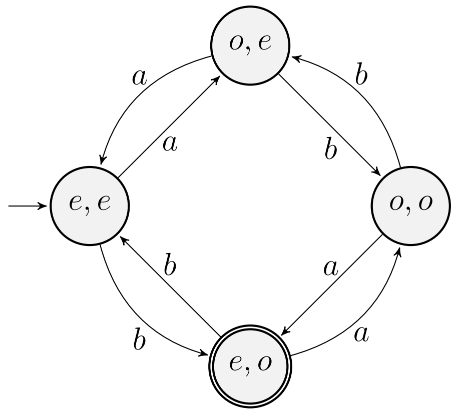
- State ϵ

Another example of a DFA which accepts strings over $\{0, 1\}$ which have even parity in each length 4 block.



Problem 2.1. Give a DFA that accepts strings over the alphabet $\{a, b\}$ containing an even number of a's and an odd number of b's.

Solution.



Problem 2.2. Give a DFA that accepts strings over $\{a, b, /, *\}$ which don't end inside a C-style comment, *i.e.*, comments of the form `/* ... */`.

Solution.