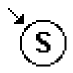
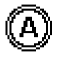
202212021
202212022
202212023
202212024
202212025

Components:

- **State** - A resting place while the machine reads more input, if more input is available. States are typically named. Canonical names for states (which we will get to later) commonly consist of the lowercase letter 'q' followed by a number, e.g. q_O. State names must be unique. Graphically, states are represented by a circle with the name inside.
- **Start State** - For programmers, this is known as the *program entry point*. It is the state that the machine naturally starts in before it reads any input. The name for the start state will usually either be S or, canonically, q_O (it is numbered with the lowest number of all the states). Graphically, the start state is represented as a state with an unlabeled arrow pointing from nothing to it. 
- **Accepting State** or **Final State** - A set of states which the machine may halt in, provided it has no input left, in order to accept the string as part of the language. Throughout this course we will exclusively use the term **Accepting** because I think it is a more descriptive term, but be aware that other places will use the term **Final** to mean the same thing. Graphically, accepting states are indicated distinctly from other states in any of a number of ways. Common ways include bolding or underlining the name, or using a double circle instead of a single. If the machine has only 1 accepting state, it is oftentimes named with the letter A.  It is worth noting that a machine with no accepting states does not accept any string as part of the language (not even the empty string) - it is essentially the Empty Language, ∅
- Finite State Automata:

  It is a collection of three things:

  1. A finite set of states, out of which one is designated as the initial state, called the start state, and some (maybe none) of which are designated as

     final states.
  2. An alphabet Σ of possible input letters.
  3. A finite set of transitions that tell for each state and for each letter of the input alphabet which state to go next.

- Deterministic Finite Automata:

  It is a mathematical model of a simple computational device that

  reads a string of symbols over the input alphabet Σ, and either accepts or

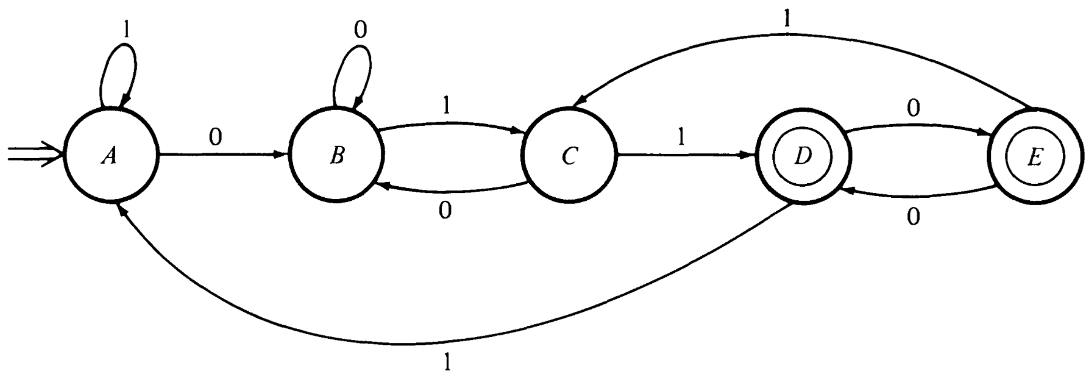
  reject the input string.

Deterministic Finite Automaton (DFA) is defined as a 5-tuple

(Q, Σ, δ,s, F) consisting of :

- ➔ A finite set Q (the set of states)
- ➔ A finite set of symbols Σ (the input alphabet)
- ➔ A transition function δ: Q × Σ → Q mapping the current state q ∈ Q
- ➔ and input symbol a ∈ Σ to a new state δ(q, a) ∈ Q
- ➔ An initial states ∈ Q (the start state)
- ➔ A set of accepting states F (the final states)

- Exp

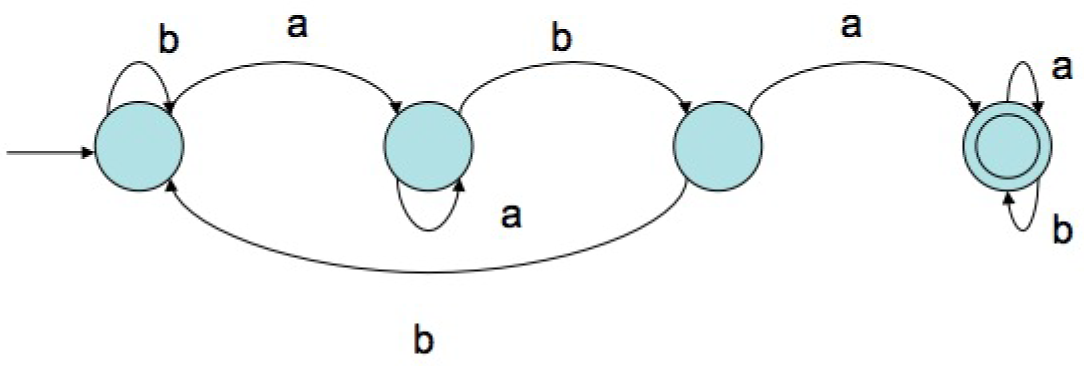


(*a*)

$A \rightarrow 0B$
$A \rightarrow 1A$
$B \rightarrow 0B$
$B \rightarrow 1C$
$C \rightarrow 0B$
$C \rightarrow 1D$
$C \rightarrow 1$
$D \rightarrow 0E$
$D \rightarrow 0$
$D \rightarrow 1A$
$E \rightarrow 0D$
$E \rightarrow 0$
$E \rightarrow 1C$

(*b*)

- Another Exp.



This DFA accepts strings that have aba

- somewhere in it.
- Once the existence of aba is ascertained, the rest of the input is ignored!

.

- **Rejecting State** - Any state in the machine which is not denoted as an accepting state. The string is only rejected if the machine *halts* in a rejecting state with no more input left. Rejecting states have no special representation, because every state is either accepting or rejecting.

Language

- Definition – A language is a subset of ∑* for some alphabet ∑. It can be finite or infinite.
- Example – If the language takes all possible strings of length 2 over ∑ = {a, b}, then L = { ab, aa, ba, bb }
- (∑ is a finite set of symbols, called the alphabet of the automaton.)
- Language over Σ : the set of strings that can be generated from Σ – Sigma star Σ* : set of all possible strings over the alphabet Σ Σ = {a, b}   Σ* = {ε, a, b, aa, ab, ba, bb, aaa, aab, ...}– Sigma plus Σ+ : Σ+ = Σ* -{ε} – Special languages: ∅ = {} (empty language) ≠ {ε} (language of empty string) A formal language : a subset of Σ*

THE LANGUAGE ACCEPTED BY A DFA

L(M) denotes the language accepted by a DFA M

L(M) = f!j! 2 and (q0; !) 2 Fg

Similarly

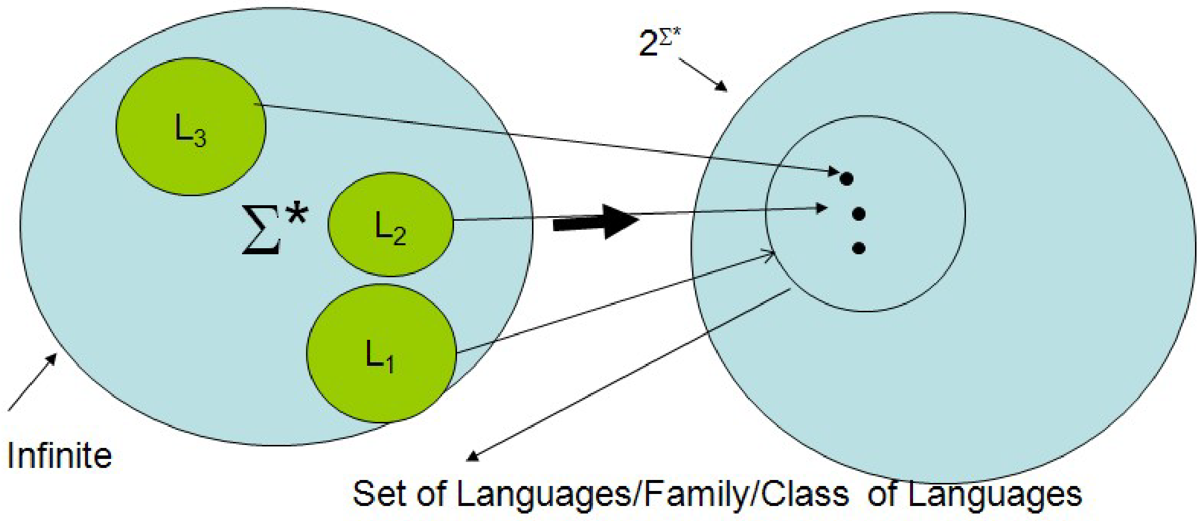
L(M) = f!j! 2 and (q0; !) 62 Fg

REGULAR LANGUAGES

L1,L2 are regular then ***L1 INTERSECTION/UNION L2*** are each regular.

A language L is called a regular language if and only

if there exists a DFA M such that L(M) = L.



Problem: Construct deterministic finite automata (DFA) for strings not containing consecutive two a's and starting with a.

Explanation:

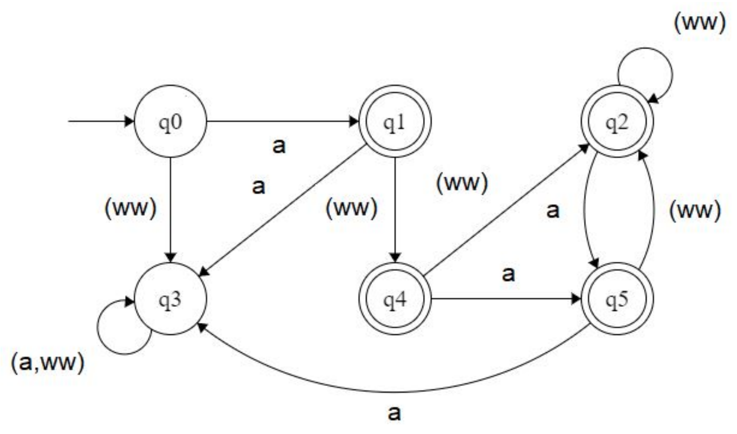
Accept Strings that not contain consecutive two a's. Check if a given string contain consecutive two a's or not. The any occurrence of (b-z) should not affect the scenario. Strings should follow this pattern:

a.(ww|(ww.a))*

where, **ww** = all possible character except a

All those strings that are not fall in the above mentioned pattern are not accepted.

Deterministic finite automata (DFA) of strings that not contain consecutive two a's given as following below. The initial and starting state in this dfa is qo.



Required DFA

—---

Finite automata have no auxiliary storage. It remembers something by changing a state. So we may say it has finite storage capability.

“Memory in finite automata is present in the form of states Q only and according to automata principle: any automata can have only finite sets of states. Hence finite automata have finite memory, this is the reason automata for regular language are called finite automata.”

Deterministic Finite Acceptor (DFA)

•It is a simple model of computation.

•It has no temporary storage.

•It has only a finite number of states.

•It is deterministic.

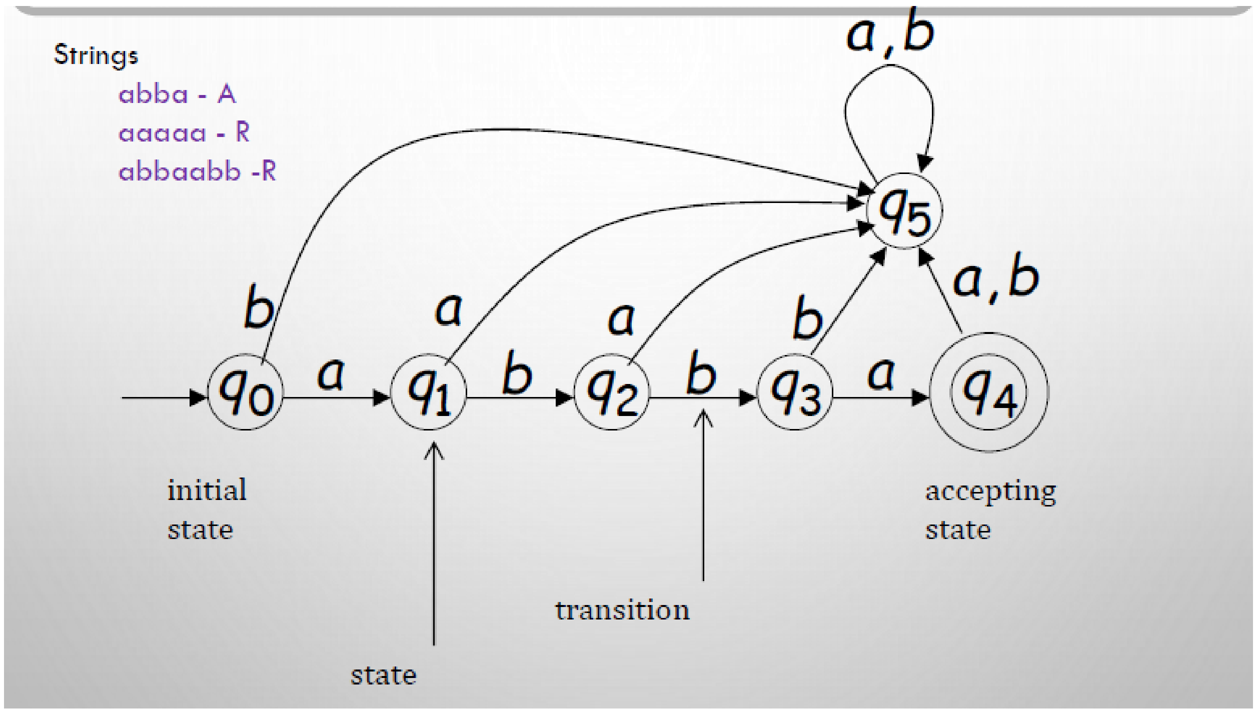
•Next move is uniquely determined by the current configuration.

•It is an acceptor.

•The output response is limited to a simple “yes” or “no”.

———————————————————————————————————————

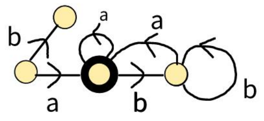
DFA-EXAMPLE (ACCEPT/REJECT)



We can not compute this number with a DFA, since the number can be arbitrarily large!

Counting is impossible using finite automata but possible with modulo counting with finite automata..

Rejected state:



Accept state: