Regular Languages

Finite Automata

A finite automaton is a system for processing any finite sequence of symbols, where the symbols are chosen from a finite set of symbols.

The goal is to determine whether the sequence has a certain property by simply reading the symbols of the sequence from the beginning to the end.

An Illustrating Example of Finite Automata

A coin exchanger takes nickels or dimes and delivers quarters. It takes coins one at a time. When the deposited amount reaches or goes beyond 25 cents it delivers a quarter. There is no "change" button and any change is carried over as a deposit.

For example, if the deposit is currently 20 cents, upon receiving a dime, the machine delivers a quarter and the deposit becomes 5 cents.

Question

If you have a bag full of nickels and dimes and use this machine to change for quarters, do you break even?

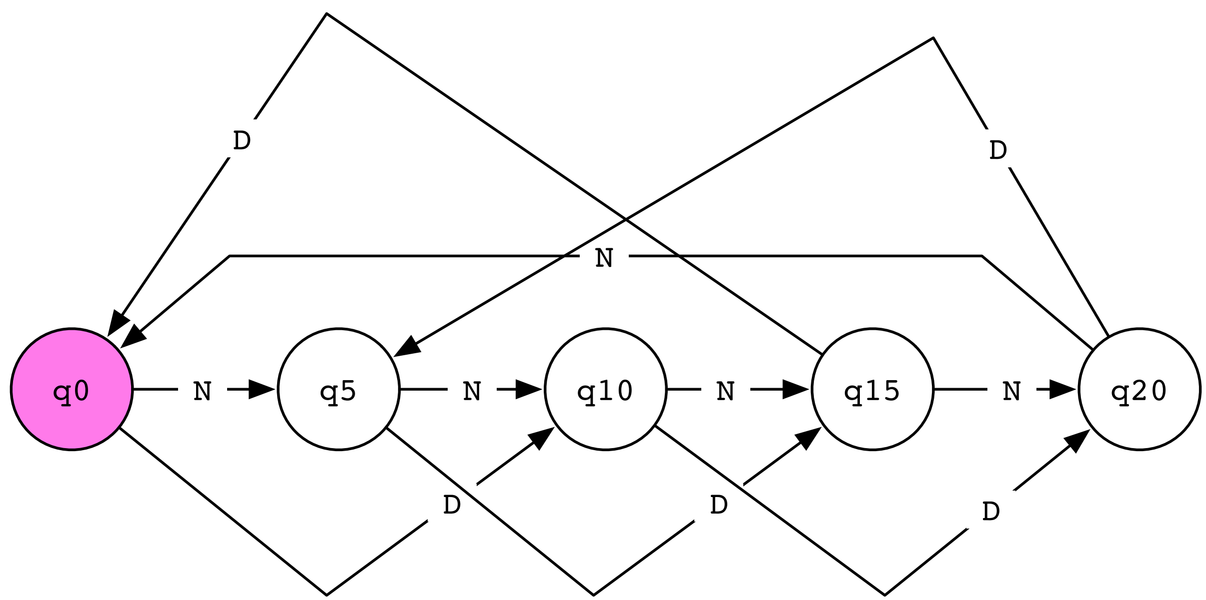
Relationship Between Deposit and Coin Inserted

The deposit amount in cents is one of 0, 5, 10, 15, and 20. Upon receiving a coin, the deposit changes as follows:

Current	Coin Inserted	
Deposit	Nickel	Dime
0	5	10
5	10	15
10	15	20
15	20	**0**
20	**0**	5

Visual Representation of the Relationship

Let N stand for "nickel" and D for "dime". For $a \in \{0, 5, 10, 15, 20\}$, let q_a represent the status in which the deposit is a cents.



Finite Automata

A **finite automaton** is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

1. Q is a finite set called the **states**,
2. Σ is a finite set called the **alphabet**,
3. $\delta : Q \times \Sigma \longrightarrow Q$ is the **transition function**,
4. $q_0 \in Q$ is the **initial state**, and
5. $F \subseteq Q$ is the set of **accepting states** or the set of **final states**.

Let $M = (Q, \Sigma, \delta, q_0, F)$ be an FA. A string $w = w_1 \cdots w_n$ is **accepted** by M if there exists a sequence $(p_0, \ldots, p_n)$ of states in Q such that $p_0 = q_0$, $p_n \in F$, and for every i, $1 \leq i \leq n$, $\delta(p_{i-1}, w_i) = p_i$.

The Language Decided by a Finite Automaton

The language **decided** by M, denoted $L(M)$, is the language over Σ such that

(*) for every string w over Σ, $w \in L(M) \Leftrightarrow M$ accepts w.

An FA for the Coin Changer

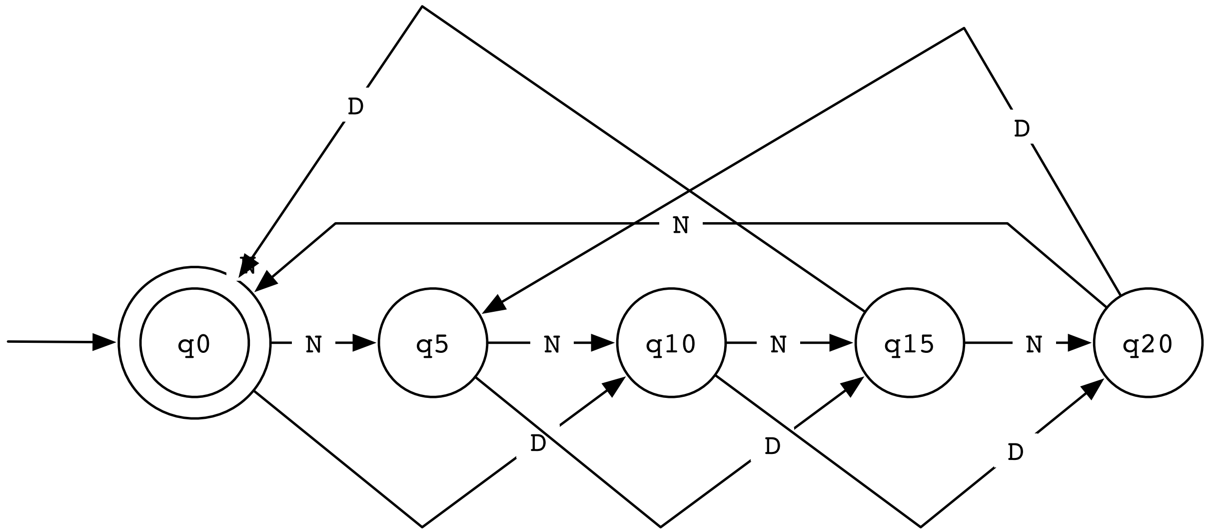
Let $\Sigma = \{N, D\}$.
Let $Q = \{q_0, q_5, q_{10}, q_{15}, q_{20}\}$.
The transition function is:

state	N	D
q_0	q_5	q_{10}
q_5	q_{10}	q_{15}
q_{10}	q_{15}	q_{20}
q_{15}	q_{20}	q_0
q_{20}	q_0	q_5

$F = \{q_0\}$.

Our FA accepts $NNND$ and $DDDDD$ but not NNN.

The Coin Changer As a Finite Automaton

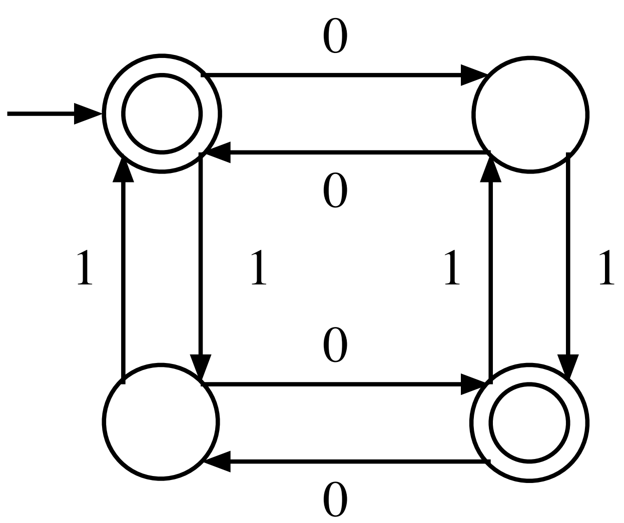


Regular Languages

The **regular languages** is the class of languages accepted by finite automata.

Example 1

An FA that accepts the strings over 0 and 1 with either (an even number of 0s and an even number of 1s) or (an odd number of 0s and an odd number of 1s)

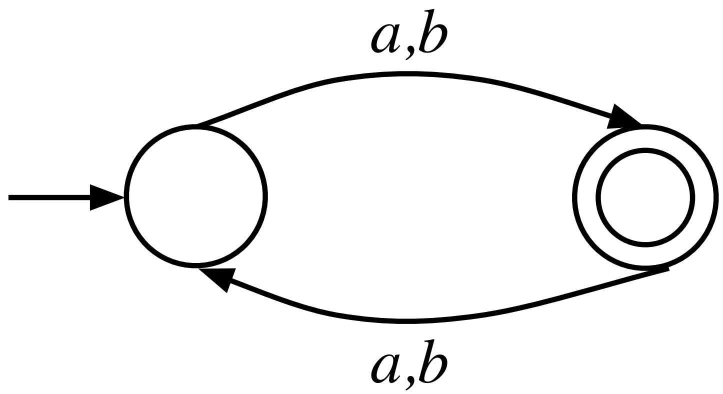


Drawing rules:

- The initial state has an incoming edge from outside.
- Accept states are represented with double circles.
- Every node has one outgoing edge for each symbol.

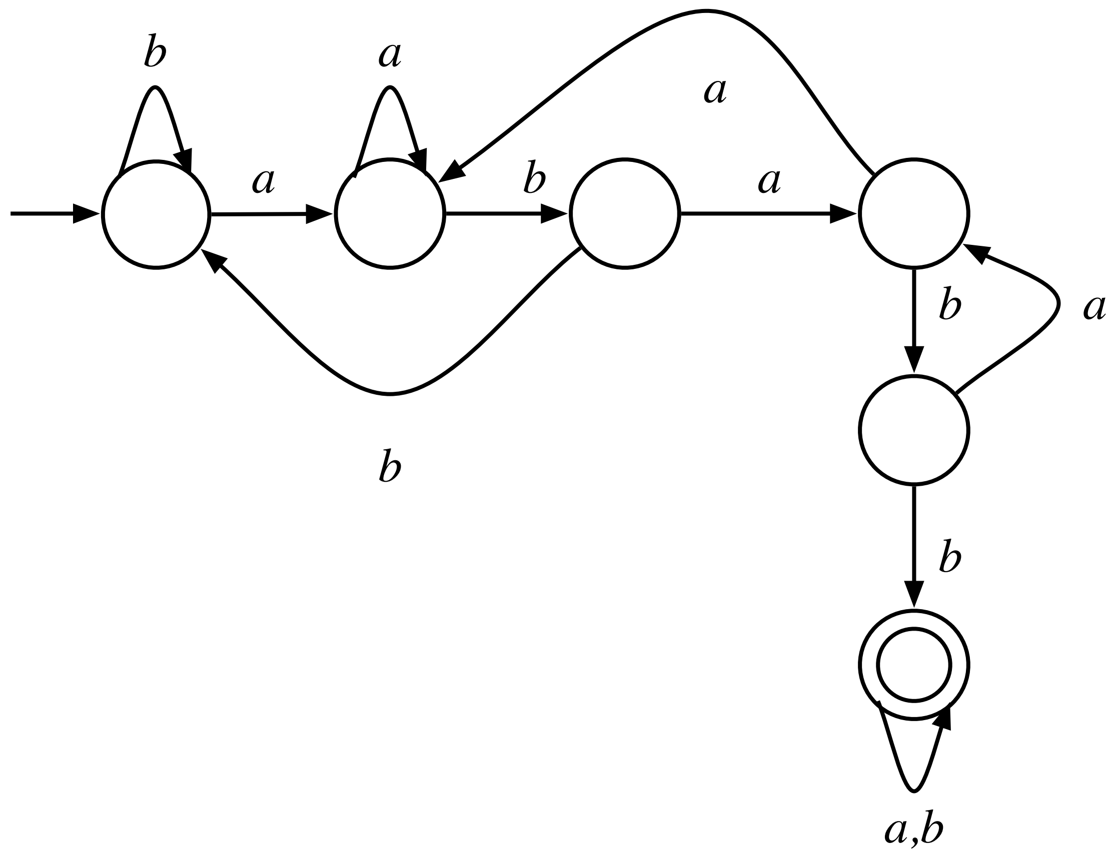
Example 2

An FA that accepts the set of all words over $\{a, b\}$ having odd length



Example 3

An FA that accepts the set of all words over $\{a, b\}$ containing $ababb$ as a subword



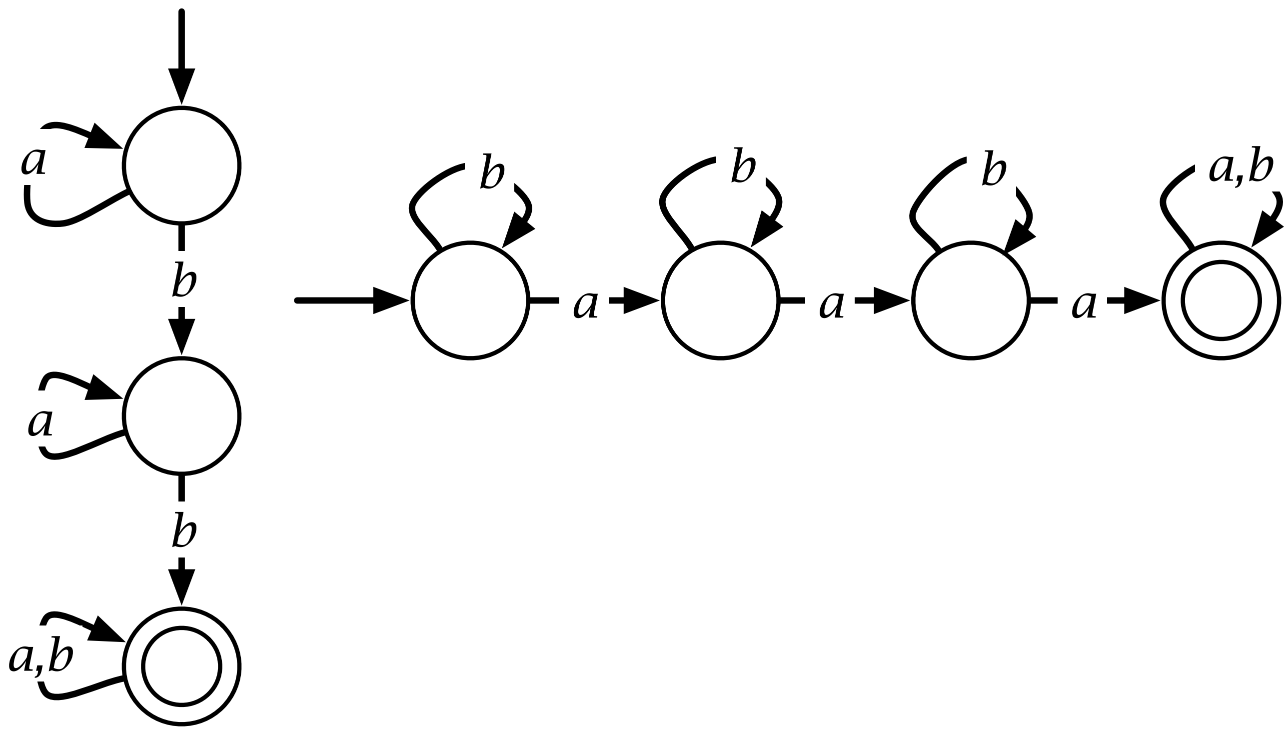
Example 4

An FA that accepts the set of all words over $\{a, b\}$ containing at least three a's and at least two b's

Example 4

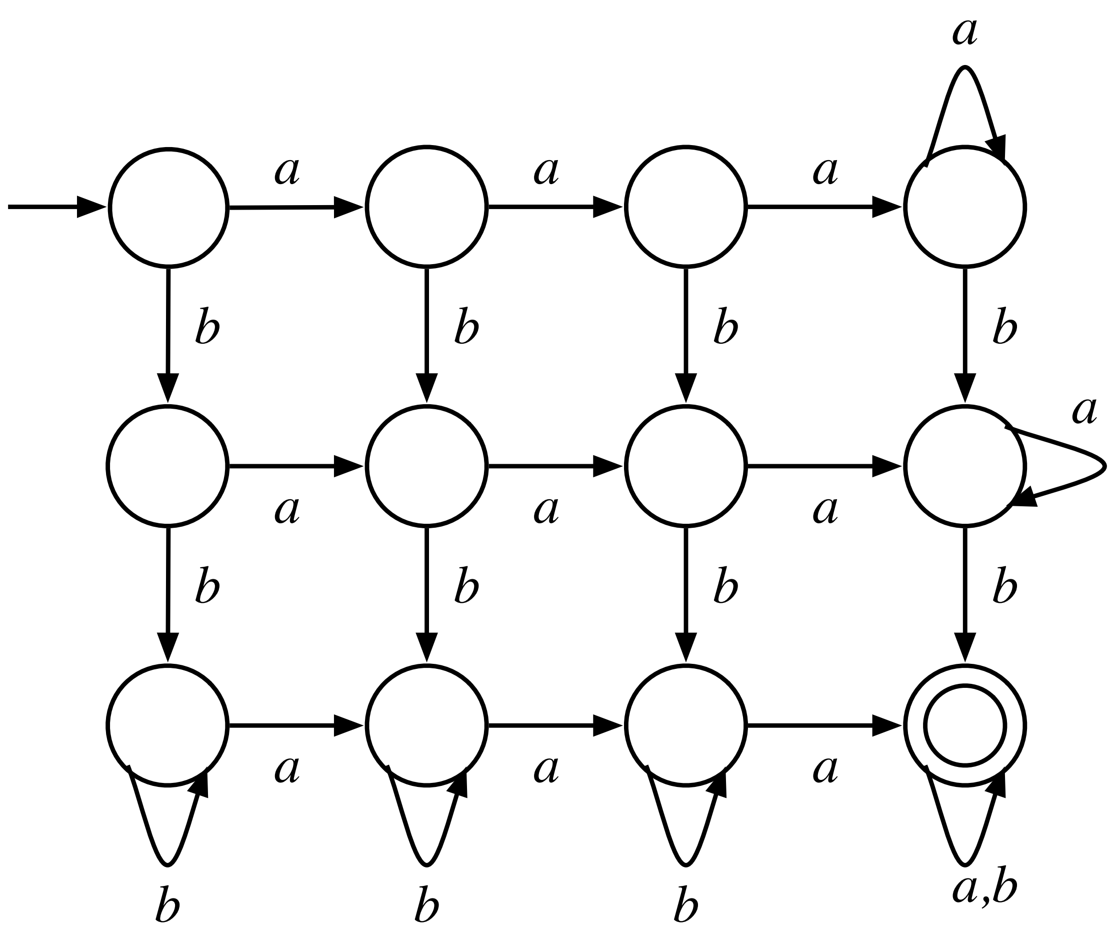
An FA that accepts the set of all words over $\{a, b\}$ containing at least three a's and at least two b's

"At least two b's" and "at least three a's"



Example 4

An FA that accepts the set of all words over $\{a, b\}$ containing at least three a's and at least two b's



Example 5

An FA that accepts the set of all words over $\{a, b\}$ containing either as least three a's or at least two b's

Example 5

An FA that accepts the set of all words over $\{a, b\}$ containing either as least three a's or at least two b's