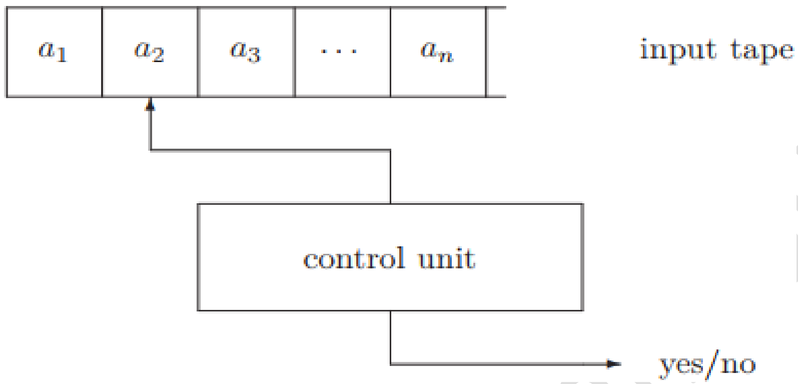
Finite State Automata

The term "Automata" is derived from the Greek word "αὐτόματα" which means "self-acting". An automaton (Automata in plural) is an abstract self-propelled computing device which follows a predetermined sequence of operations automatically. An automaton with a finite number of states is called a **Finite Automaton** (FA) or **Finite State Machine** (FSM).

الآتي هو شكل رسومي يمثل تنقلات بين مجموعة حالات تسيطر عليها قواعد عن طريق شريط من المدخلات.



The above device consisting of a tape and a control circuit, which satisfy the following conditions:

1. The tape starts from the left end and extends to the right without an end.
2. The tape is divided into squares in each a symbol.
3. The tape has a read only head.
4. The head moves to the right one square every time it reads a symbol. It never moves to the left. When it sees no symbol, it stops, and the automata terminates its operation.
5. There is a control determines the state of the automaton and also controls the movement of the head.

Formal definition of a Finite Automaton

An automaton can be represented by a 5-tuple (Q, ∑, δ, q, F),
Where,

Q is a finite set of **states** = $\{q_0, q_1, q_2, \ldots\}$.
∑ is a finite set of symbols = {a, b, …, 0, 1, …}, called the **alphabet** of the automaton.
δ is the **transition function**.
q is the **initial state** from where any input is processed ($q \in Q$).
F is a set of **final state/states** of Q ($F \subseteq Q$).

Deterministic & Non-Deterministic Finite Automata

Finite Automaton can be classified into two types:

- Deterministic Finite Automaton (DFA) الاتوماتا المنتهي الحتمي
- Non-deterministic Finite Automaton (NDFA / NFA) الاتوماتا المنتهي اللاحتمي

Deterministic Finite Automaton (DFA)

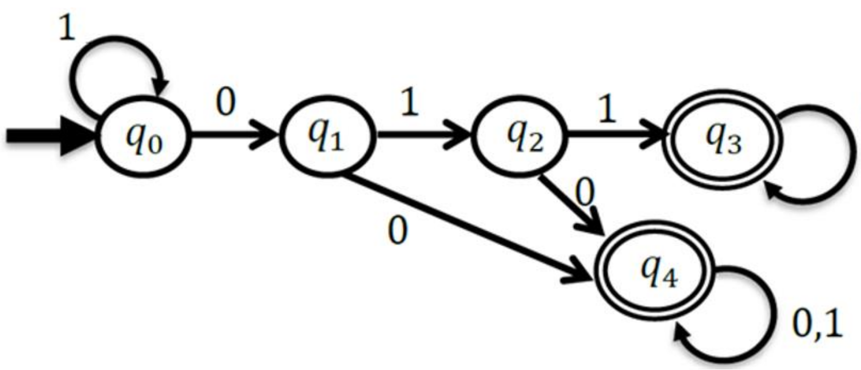
In DFA, for each input symbol, one can determine the state to which the machine will move. Hence, it is called **Deterministic Automaton**. As it has a finite number of states, the machine is called **Deterministic Finite Machine** or **Deterministic Finite Automaton.**

(*) أستاذ مساعد، قسم الذكاء الاصطناعي، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

Mohammed Chachan Younis

الاتوماتا المنتهي الحتمي: هو اوتوماتا عدد حالاته منتهي، وكل حالة q_i يخرج منها عدد اسهم بعدد رموز الأبجدية، وكل سهم يكون مختلف عن الأخر بالرمز المختار له.

المثال الآتي هو اوتوماتا منتهي حتمي على الأبجدية $\sum=\{0, 1\}$ حيث نلاحظ أنه من كل حالة يخرج عدد اسهم مساوٍ لعدد رموز الأبجدية ولا يوجد سهمين لهما نفس الرمز يخرجان من نفس الحالة.



Formal Definition of a DFA

A DFA can be represented by a 5-tuple (Q, ∑, δ, q_0, F) where:

- **Q** is a finite set of states.
- ∑ is a finite set of symbols called the alphabet.
- **δ** is the transition function where $\delta: Q \times \sum \rightarrow Q$.
- **q_0** is the initial state from where any input is processed ($q_0 \in Q$).
- **F** is a set of final state/states of Q ($F \subseteq Q$).

Graphical Representation of a DFA

A DFA is represented by digraphs called **state diagram**.

- The vertices represent the states.
- The arcs labelled with an input alphabet show the transitions.
- The initial state is denoted by an empty single incoming arc.
- The final state is indicated by double circles.

Example

Let a deterministic finite automaton be →

- Q = {a, b, c},
- ∑ = {0, 1},
- q = {a},
- F = {c}, and

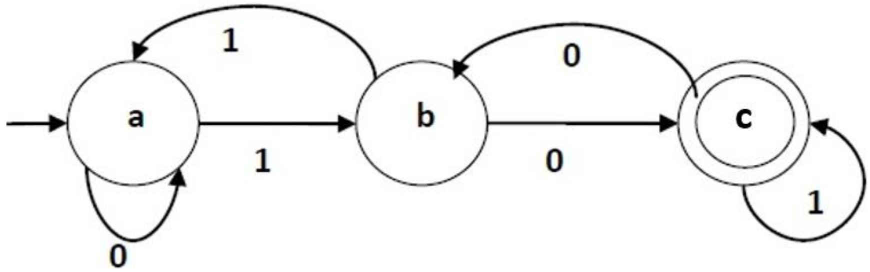
Transition function **δ** as shown by the following table:

Present State	Next State for Input 0	Next State for Input 1
a	a	b
b	c	a
c	b	c

(*) أستاذ مساعد، قسم الذكاء الاصطناعي، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

Mohammed Chachan Yo...

Its graphical representation would be as follows:



Non-deterministic Finite Automaton (NDFA)

In NDFA, for a particular input symbol, the machine can move to any combination of the states in the machine. In other words, the exact state to which the machine moves cannot be determined. Hence, it is called **Non-deterministic Automaton**. As it has a finite number of states, the machine is called **Non-deterministic Finite Machine** or **Non-deterministic Finite Automaton**.

Formal Definition of an NDFA

An NDFA can be represented by a 5-tuple (Q, ∑, δ, q_0, F) where:

Q is a finite set of states.

∑ is a finite set of symbols called the alphabets.

δ is the transition function where δ: $Q \times \sum \rightarrow 2^Q$

(Here the power set of Q (2^Q) has been taken because in case of NDFA, from a state, transition can occur to any combination of Q states).

q_0 is the initial state from where any input is processed ($q_0 \in Q$).

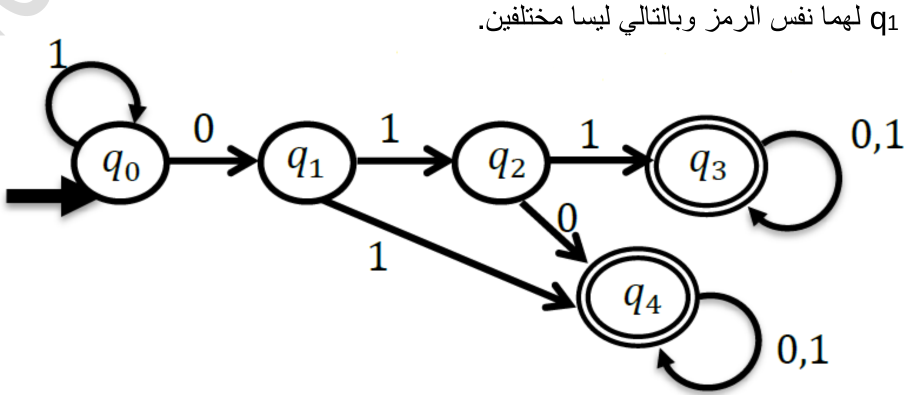
F is a set of final state/states of Q ($F \subseteq Q$).

Graphical Representation of an NDFA: (same as DFA)

An NDFA is represented by digraphs called **state diagram**.

- The vertices represent the states.
- The arcs labeled with an input alphabet show the transitions.
- The initial state is denoted by an empty single incoming arc.
- The final state is indicated by double circles.

الاتوماتا المنتهي اللاحتمي: هو اوتوماتا عدد حالاته منتهي، وكل حالة q_i ليس بالضرورة أن يخرج منها عدد اسهم بعدد رموز الابجدية (ممكن أقل وممكن أكثر)، وايضاً ليس بالضرورة كل سهم مختلف عن الآخر بالاختيار وممكن للحالة الواحدة أن يوجد اختيار ينقلنا لعدة حالات مختلفة. المثال الآتي هو اوتوماتا منتهي لا حتمي على الأبجدية {0, 1}=∑ حيث نلاحظ خروج سهمين من الحالة q_1 لهما نفس الرمز وبالتالي ليسا مختلفين.



Mohammed Chachan Younis

(*) أستاذ مساعد، قسم الذكاء الاصطناعي، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

Example

Let a non-deterministic finite automaton be →

$Q = \{a, b, c\}$

$\sum = \{0, 1\}$

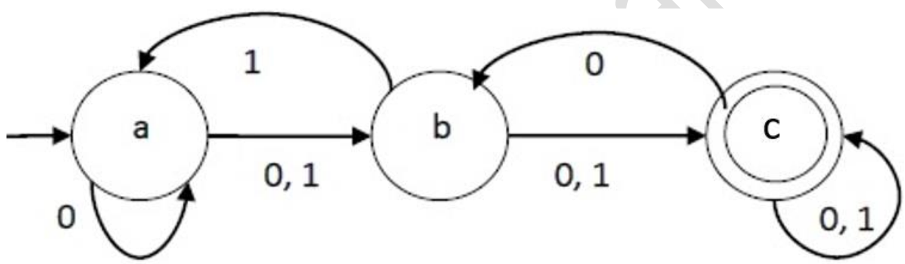
$q_0 = \{a\}$

$F = \{c\}$, and

The Transition function **δ** as shown below:

Present State	Next State for Input 0	Next State for Input 1
a	a, b	b
b	c	a, c
c	b, c	c

Its graphical representation would be as follows:



ملاحظة: في الأوتوماتا المنتهي الحتمي فإن الأسهم التي تخرج من كل حالة تكون مساوية لعدد رموز الأبجدية بينما في الأوتوماتا المنتهي اللاحتمي فليس بالضرورة أن تساوي عدد رموز الأبجدية، وأيضاً في الأوتوماتا المنتهي اللاحتمي قد يخرج من حالة واحدة سهمان (أو أكثر) يحملان نفس قيمة الرمز ويصلان إلى نفس الحالة (أو أكثر) وهذا الأمر غير ممكن في الأوتوماتا المنتهي الحتمي.

DFA vs NDFA

The following table lists the differences between DFA and NDFA:

DFA	NDFA
The transition from a state is to a single particular next state for each input symbol. Hence it is called *deterministic*.	The transition from a state can be to multiple next states for each input symbol. Hence it is called *non-deterministic*.
Empty string transitions are not seen in DFA.	NDFA permits empty string transitions.
Backtracking is allowed in DFA.	In NDFA, backtracking is not always possible.
Requires more space.	Requires less space.
A string is accepted by a DFA, if it transits to a final state.	A string is accepted by a NDFA, if at least one of all possible transitions ends in a final state.

(*) أستاذ مساعد، قسم الذكاء الاصطناعي، كلية علوم الحاسوب والرياضيات، جامعة الموصل.