

# *Computation Theory*

## نظرية الحساب

تأليف

د. محمد خليل أبو زلطة

د. مصباح جمعة عقل

د. زياد عبد الكريم القاضي

الطبعة الأولى

2009 م - 1430 هـ



مكتبة المجتمع العربي للنشر والتوزيع

## الفهرس

الصفحة

المحتوى

### الوحدة الأولى

#### مدخل الى نظرية الحساب

- 11 ..... 1.1 المنطق الاقتراحي
- 19 ..... 2.1 المجموعات
- 34 ..... 3.1 العلاقات
- 37 ..... 4.1 الاقتران
- 37 ..... 5.1 التعابير المنتظمة

### الوحدة الثانية

#### آلة الحالة المنتهية

- 51 ..... 1-2 مقدمة
- 54 ..... 2-2 النموذج الرياضي للآلة المنتهية
- 61 ..... 3-2 اهم المصطلحات
- 80 ..... 4-2 اللغة المقبولة من آلة الحالة المنتهية
- 89 ..... 5-2 اللغة المكتملة

### الوحدة الثالثة

#### آلة الحالة المنتهية غير المحدودة

- 93 ..... 1-3 تعريف آلة الحالة المنتهية غير المحدودة
- 108 ..... 2-3 لغة الآلة المنتهية غير المحدودة
- 114 ..... 3-3 تحويل آلة الحالة المنتهية غير المحدودة الى آلة محدودة
- 121 ..... 4-3 التخلص من ايبلسون في الآلة المنتهية غير المحدودة
- 124 ..... 5-3 تطبيقات الآلات الحالة المنتهية

## الوحدة الرابعة

## آلة الحالة المنتهية المستخدمة للحزمة

- 129 ..... 1-4 المفهوم العام لآلة الحالة المستخدمة للحزمة
- 165 ..... 2-4 التعريف الشكلي لآلة الحالة المنتهية المستخدمة للحزمة
- 169 ..... 3-4 تمارين

## الوحدة الخامسة

## آلة يتورينج

- 179 ..... 1-5 التعريف بالآلة يتورينج
- 202 ..... 2-5 النموذج الرياضي لآلة يتورينج
- 206 ..... 3-5 تطبيقات آلة يتورينج

## الوحدة السادسة

## آلة مور وآلة ميلي

- 231 ..... 1-6 مقدمة
- 238 ..... 2-6 آلة مور وآلة ميلي
- 245 ..... 3-6 تصميم آلة الحالة المنتهية
- 259 ..... المراجع

تستخدم مفاهيم نظرية الحساب في كثير من التطبيقات العملية حيث تستخدم في تصميم الدارات المنطقية وفي تصميم المترجمات ومعالجات النصوص والانظمة البرمجية التي تعتمد على عملية تمييز الانماط مثل انظمة معالجة الصور الرقمية وانظمة معالجة الصوت وانظمة الذكاء الصناعي المختلفة.

ونظرا لاهمية نظرية الحساب واستخداماتها المختلفة في تطبيقات علم الحاسوب وهندسته وغيره من تطبيقات فق جاء هذا الكتاب لتعريف القارئ العربي باهمية هذه النظرية واطلاعه على اهم مواضيعها ونخص بالذكر:

- الات الحالة المنتهية المحدودة.
- الات الحالة المنتهية غير المحدودة.
- الة الحالة باستخدام الحزمة.
- الة تيورينج.
- الة مور و الة ميلي.

هذا وقد حاولنا الاكثار من الامثلة التوضيحية املا منا في تسهيل فهم مبادئ نظرية الحساب امليين نكون قد وفقنا في اصال المعلومة بشكل واضح وسهل.

والله ولي التوفيق.....



الوحدة الأولى

مدخل الى نظرية

الحساب

1

تستخدم في نظرية الحساب بعض المواضيع الرياضية ومن هذه المواضيع:

- 1 اساسيات المنطق.
- 2 المجموعات.
- 3 الاقترانان.
- 4 العلاقات.
- 5 التعابير المنتظمة.

وسوف نستعرض في هذه الوحدة وبشكل مختصر هذه المواضيع نظرا لاستخداماتها الكثيرة في الوحدات اللاحقة علما بانها يمكن الرجوع الى كتاب الرياضيات المنفصلة للمؤلفين وذلك مزيدا للمعلومات.

### 1.1 المنطق الاقتراحي Propositional Logic:

يعرف الاقتراح Proposition على انه جملة تصريحية تاخذ قيمة الصواب

او الخطا فمثلا الجمل التالية صحيحة:

- 6 العدد 10 هو عدد زوجي.
- 7 العدد 4 هو مربع كامل.
- 8 عمان عاصمة الاردن.
- 9 في حين تكون الجمل التالية خاطئة:
- 10 جرش عاصمة الاردن.
- 11 العدد 3 عدد زوجي.
- 12 يقبل العدد 20 القسمة على 9 بدون باقي.

هذا ويمكن ربط الجمل التصريحية معا باستخدام مجموعة من العلاقات المنطقية مثل علاقة "و" and وتأخذ الجملة التصريحية هنا ايضا قيمة واحدة هي الصواب او الخطا فمثلا الجمل التالية صحيحة:

- عمان او جرش مدينة اردنية.
- العدد 10 زوجي ويقبل القسمة على 5 بدون باقى.

اما الجمل التالية فهي خاطئة:

- عمان وجرش عاصمة الاردن.
- العدد 21 فردي و قبل القسمة على 4 بدون باقى.

وفيما يلي سوف نستعرض اهم العمليات المنطقية المستخدمة لربط الجمل:

### 1. علاقة الضرب المنطقي (Conjunction(And)

تستخدم هذه العلاقة لربط اكثر من جملة وتكون النتيجة النهائية للجملة صحيحة اذا كانت كافة الجمل المرتبطة بهذه العلاقة صحيحة ويبين جدول الصواب التالي عملية الربط باستخدام هذه العلاقة:

$p$	$q$	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

## 2. علاقة الجمع المنطقي (OR) Disjunction

تستخدم هذه العلاقة لربط اكثر من جملة و تكون النتيجة صحيحة اذا كانت على الاقل قيمة احدى الجمل صحيحة ويبين الجدول التالي جدول الصواب لهذه العلاقة:

$p$	$q$	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

## 3. علاقة النفي Negation

وتستخدم هذه العلاقة لنفي جملة لتصبح النتيجة صحيحة اذا كانت الجملة خاطئة وبالعكس ويبين الجدول التالي جدول الصواب لهذه العلاقة:

$p$	$\neg p$
T	F
F	T

## 4. العلاقة الشرطية احادية الاتجاه Conditional

وتربط هذه العلاقة بين جملتين وتكون النتيجة صائبة اذا كانت الجملة الثانية صحيحة او كانت الجملتان خاطئتين والجدول التالي يبين جدول الصواب لهذه العلاقة:

$p$	$q$	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

### 5. العلاقة الشرطية ثنائية الاتجاه Biconditional

تربط هذه العلاقة بين جملتين وتكون النتيجة صائبة اذا تشابهت قيم

الجملتين ويبين الجدول التالي جدول الصواب لهذه العلاقة:

$p$	$q$	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

### 6. الحقيقة Tautology

جملة او اكثر مرتبطة معا وتكون نتيجتها دائما صحيحة مثل:

$$p \vee \neg p$$

### 7. التناقض Contradiction

جملة او اكثر مرتبطة معا وتكون نتيجتها دائما خاطئة مثل:

$$p \wedge \neg p$$

وفيما يلي اهم قواعد المنطق والمثلة للحقائق:

### List of Identities:

1.  $P \Leftrightarrow (P \vee P)$  ----- idempotence of  $\vee$
2.  $P \Leftrightarrow (P \wedge P)$  ----- idempotence of  $\wedge$
3.  $(P \vee Q) \Leftrightarrow (Q \vee P)$  ----- commutativity of  $\vee$
4.  $(P \wedge Q) \Leftrightarrow (Q \wedge P)$  ----- commutativity of  $\wedge$
5.  $[(P \vee Q) \vee R] \Leftrightarrow [P \vee (Q \vee R)]$  ----- associativity of  $\vee$
6.  $[(P \wedge Q) \wedge R] \Leftrightarrow [P \wedge (Q \wedge R)]$  ----- associativity of  $\wedge$
7.  $\neg(P \vee Q) \Leftrightarrow (\neg P \wedge \neg Q)$  ----- DeMorgan's Law
8.  $\neg(P \wedge Q) \Leftrightarrow (\neg P \vee \neg Q)$  ----- DeMorgan's Law
9.  $[P \wedge (Q \vee R)] \Leftrightarrow [(P \wedge Q) \vee (P \wedge R)]$  ----- distributivity of  $\wedge$  over  $\vee$
10.  $[P \vee (Q \wedge R)] \Leftrightarrow [(P \vee Q) \wedge (P \vee R)]$  ----- distributivity of  $\vee$  over  $\wedge$
11.  $(P \vee \text{True}) \Leftrightarrow \text{True}$
12.  $(P \wedge \text{False}) \Leftrightarrow \text{False}$
13.  $(P \vee \text{False}) \Leftrightarrow P$
14.  $(P \wedge \text{True}) \Leftrightarrow P$
15.  $(P \vee \neg P) \Leftrightarrow \text{True}$
16.  $(P \wedge \neg P) \Leftrightarrow \text{False}$
17.  $P \Leftrightarrow \neg(\neg P)$  ----- double negation
18.  $(P \rightarrow Q) \Leftrightarrow (\neg P \vee Q)$  ----- implication
19.  $(P \leftrightarrow Q) \Leftrightarrow [(P \rightarrow Q) \wedge (Q \rightarrow P)]$  ----- equivalence
20.  $[(P \wedge Q) \rightarrow R] \Leftrightarrow [P \rightarrow (Q \rightarrow R)]$  ----- exportation
21.  $[(P \rightarrow Q) \wedge (P \rightarrow \neg Q)] \Leftrightarrow \neg P$  ----- absurdity
22.  $(P \rightarrow Q) \Leftrightarrow (\neg Q \rightarrow \neg P)$  ----- contrapositive

اما قوانين الاستنتاج الاساسية فهي كما يلي:

### List of Implications:

1.  $P \Rightarrow (P \vee Q)$  ----- addition
2.  $(P \wedge Q) \Rightarrow P$  ----- simplification

3.  $[P \wedge (P \rightarrow Q)] \Rightarrow Q$  ----- modus ponens
4.  $[(P \rightarrow Q) \wedge \neg Q] \Rightarrow \neg P$  ----- modus tollens
5.  $[\neg P \wedge (P \vee Q)] \Rightarrow Q$  ----- disjunctive syllogism
6.  $[(P \rightarrow Q) \wedge (Q \rightarrow R)] \Rightarrow (P \rightarrow R)$  ----- hypothetical syllogism
7.  $(P \rightarrow Q) \Rightarrow [(Q \rightarrow R) \rightarrow (P \rightarrow R)]$
8.  $[(P \rightarrow Q) \wedge (R \rightarrow S)] \Rightarrow [(P \wedge R) \rightarrow (Q \wedge S)]$
9.  $[(P \leftrightarrow Q) \wedge (Q \leftrightarrow R)] \Rightarrow (P \leftrightarrow R)$

وفيما يلي بعض الامثلة التوضيحية:

1. Construct the truth tables for the following tables for the following statements, and use the results to find logical implications and logical equivalences among them( say which statements imply which others, and which are equivalent to which others).

- a.  $p \rightarrow q) \wedge (p \rightarrow \neg q)$
- b.  $p \vee (p \rightarrow q)$
- c.  $p \wedge (p \rightarrow q)$
- d.  $p \rightarrow q) \wedge (\neg p \rightarrow q)$
- e.  $p \leftrightarrow (p \leftrightarrow q)$
- f.  $q \wedge (p \rightarrow q)$

### Solution:

Here are the solutions for a, c & e...

$$(a)(p \rightarrow q) \wedge (p \rightarrow \neg q)$$

Assuming you have gone through the notes material explaining the basic concepts

Of Logic, we start building the truth tables without much explanation of as to how the truth values came.

p	q	$q^{\neg}$	$q \rightarrow p$	$q^{\neg} \rightarrow p$
T	T	F	T	F
T	F	T	F	T
F	T	F	T	T
F	F	T	T	T

q)  $\rightarrow$ (c) p  $\wedge$ (p

p	q	$q \rightarrow p$	$q \rightarrow p \wedge (p \wedge q)$
T	T	T	T
T	F	F	F
F	T	T	F
F	F	T	F

$\leftrightarrow$ ( p  $\leftrightarrow$ p

(e)  
q)

p	q	$q \leftrightarrow p$	$\leftrightarrow(p \leftrightarrow p \wedge q)$
T	T	T	T
T	F	F	F
F	T	F	T
F	F	T	F

1.14) Show that the statements  $p \vee q \vee r \vee s$  and  $(\neg p \wedge \neg q \wedge \neg r) \rightarrow s$  are equivalent.



Solution:

p	q	r	s	$p \vee q \vee r \vee s$
T	T	T	T	T
T	T	T	F	T
T	T	F	T	T
T	T	F	F	T
T	F	T	T	T
T	F	T	F	T
T	F	F	T	T
T	F	F	F	T
F	T	T	T	T
F	T	T	F	T
F	T	F	T	T
F	T	F	F	T
F	F	T	T	T
F	F	T	F	T
F	F	F	T	T
F	F	F	F	F

p	q	R	s	$p \neg$	$q \neg$	$r \neg$	$s \rightarrow r \neg q \wedge \neg p \wedge \neg$
T	T	T	T	F	F	F	T
T	T	T	F	F	F	F	T
T	T	F	T	F	F	T	T
T	T	F	F	F	T	T	T

T	F	T	T	F	T	F	T
T	F	T	F	F	T	F	T
T	F	F	T	F	T	T	T
T	F	F	F	F	T	T	T
F	T	T	T	T	F	F	T
F	T	T	F	T	F	F	T
F	T	F	T	T	F	T	T
F	T	F	F	T	F	T	T
F	F	T	T	T	T	F	T
F	F	T	F	T	T	T	T
F	F	F	T	T	T	T	T
F	F	F	F	T	T	T	F

The truth values for  $p \vee q \vee r \vee s$  &  $(\neg p \wedge \neg q \wedge \neg r) \rightarrow s$  are same in each case, then we can conclude that  $p \vee q \vee r \vee s$  and  $(\neg p \wedge \neg q \wedge \neg r) \rightarrow s$  are logically equivalent, written as

$$p \vee q \vee r \vee s \Leftrightarrow (\neg p \wedge \neg q \wedge \neg r) \rightarrow s$$

### 2.1 المجموعات Sets:

تعرف المجموعة على انها مجموعة من العناصر ويمكن ان تكون هذه العناصر من نفس النوع او يمكن ان تكون مختلفة.

وفيما يلي بعض الامثلة على المجموعات:

1. The set of students in this class
2. The set N of natural number (all non-negative integers)  $\{0, 1, 2, 3, \dots\}$

3. The set Z of all integers both positive and negative  $\{\dots, -2, -1, 0, 1, 2, \dots\}$
4. The set Q of all rational numbers (numbers that can be expressed as  $p/q$ , where  $p$  and  $q$  are elements of Z)
5. The set R of real numbers.
6. The set C of complex numbers

وفيما يلي اهم الصيغ الرياضية المرتبطة بالمجموعات:

1. اذا كان العنصر من بين عناصر المجموعة فان هذا العنصر ينتمي الى هذه المجموعة.
2. تكون المجموعة خالية اذا لم تحتوي على اي عنصر.
3. تكون المجموعة منتهية اذا كان عدد العناصر محدد.
4. تكون المجموعة لانتهائية اذا ما كان عدد عناصرها غير منتهي.
5. ترتيب العناصر في المجموعة غير مهم.
6. اذا تكررت القيمة الواحدة كعناصر في المجموعة فانها تشكل عنصرا واحدا ويمكن الغاء التكرار دون التأثير على المجموعة.
7. يطلق على عدد العناصر في المجموعة العمق او cardinality.
8. اذا كان احد عناصر المجموعة عنصرا فان هذه المجموعة الجزئية تعتبر عنصرا واحدا وتنتمي للمجموعة الاصلية.

وفيما يلي بعض الامثلة والتي توضح هذه المفاهيم:

$$A = \{1, 3, 5, 7, 9\} \quad 1 \in A, 1 \in B, 1 \in C$$

$$B = \{x | x \text{ is odd}\}$$

$$C = \{1, 3, 5, 7, 9, \dots\}$$

cardinality of  $A = 5$  ( $|A| = 5$ )

$A$  is a proper subset of  $B$ .  $A \subset B$

$C$  is a subset of  $B$ .  $C \subseteq B$

9. اذا كانت مجموعة منتمية لمجموعة اخرى فان اي عنصر فيها ينتمي الى المجموعة الاخرى.

10. تكون المجموعتان متساويتين اذا احتوت كل منهما على نفس العناصر:

Sets and Subsets

$$\text{subsets } A \subseteq B \Leftrightarrow \forall x [x \in A \Rightarrow x \in B]$$

$$A \not\subseteq B \Leftrightarrow \neg \forall x [x \in A \Rightarrow x \in B]$$

$$\Leftrightarrow \exists x \neg [\neg(x \in A) \vee x \in B]$$

$$\Leftrightarrow \exists x [x \in A \wedge x \notin B]$$

$$\text{set equality } C = D \Leftrightarrow (C \subseteq D) \wedge (D \subseteq C)$$

$$C \neq D \Leftrightarrow \neg (C \subseteq D \wedge D \subseteq C)$$

$$\Leftrightarrow C \not\subseteq D \vee D \not\subseteq C$$

### المجموعة الاسية Power set

المجموعة الاسية لمجموعة ما هي مجموعة عدد عناصرها مساو 2 مرفوعا

لاس مساو عدد عناصر المصفوفة:

$$\text{If } |A|=n, \text{ then } |P(A)|=2^n.$$

مثال:

if  $X = \{a, b, c\}$  then

$$P(X) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$$

الضرب الديكارتي لمجموعتين:

الضرب الديكارتي لمجموعتين هو مجموعة عناصرها تشكل كافة الاحتمالات الممكنة لتوليف عناصر المجموعتين.

مثال

if  $A = \{1, 2, 3\}$  and  $B = \{a, b\}$ , then  $A \times B = \{(1, a), (1, b), (2, a), (2, b), (3, a), (3, b)\}$ .

امثلة

Example

$A = \{a, e, i, o, u\}$  is a set and the list of all its elements is given.

Example

$$B = \{x : x \text{ is an integer, } x > 0\}$$

Consider the set  $C = \{1, 3, 5, 7, 9\}$ . We write

$3 \in C$  to mean that 3 belongs to the set  $C$ , and

$-5 \notin C$  to mean that -5 does not belong to  $C$ .

Example

$$\text{Let } A = \{a, e, i, o, u\}, B = \{u, o, i, e, a\} \text{ and } C = \{a, a, e, i, i, o, u\}$$

$$\text{then } A = B = C$$

Example

Let  $X = \{y : y^2 = 4, y \text{ is odd}\}$

then  $X$  is the empty set and we write

$X = \emptyset$ .

Example

Let  $A = \{1, 3, 5\}$  and  $B = \{1, 2, 3, 4, 5\}$

then  $A \subset B$  and  $B \not\subset A$

Example

Let  $A = \{1, 3, 5, 7\}$  and  $B = \{2, 4, 6, 7\}$

then  $A$  and  $B$  are not disjoint because 7 is in both sets:

$7 \in A$  and  $7 \in B$

وفيما يلي اهم المجموعات الشائعة الاستخدام:

- a.  $Z$ =the set of integers=  $\{0, 1, -1, 2, -2, 3, -3, \dots\}$
- b.  $N$ =the set of nonnegative integers or natural numbers
- c.  $Z^+$ =the set of positive integers
- d.  $Q$ =the set of rational numbers=  $\{a/b \mid a, b \text{ is integer, } b \text{ not zero}\}$
- e.  $Q^+$ =the set of positive rational numbers
- f.  $Q^*$ =the set of nonzero rational numbers
- g.  $R$ =the set of real numbers
- h.  $R^+$ =the set of positive real numbers
- i.  $R^*$ =the set of nonzero real numbers
- j.  $C$ =the set of complex numbers

اسئلة:

$$Q1: U = \mathbb{N}. \{x \mid \forall y(y \geq x)\} = ?$$

$$Q2: U = \mathbb{Z}. \{x \mid \forall y(y \geq x)\} = ?$$

$$Q3: U = \mathbb{Z}. \{x \mid \exists y(y \in \mathbb{R} \wedge y^2 = x)\} = ?$$

$$Q4: U = \mathbb{Z}. \{x \mid \exists y(y \in \mathbb{R} \wedge y^3 = x)\} = ?$$

$$Q5: U = \mathbb{R}. \{|x| \mid x \in \mathbb{Z}\} = ?$$

$$Q6: U = \mathbb{R}. \{|x|\} = ?$$

$$A1: U = \mathbb{N}. \{X \mid \forall y(y \geq x)\} = \{0\}$$

$$A2: U = \mathbb{Z}. \{x \mid \forall y(y \geq x)\} = \{ \}$$

$$A3: U = \mathbb{Z}. \{x \mid \exists y(y \in \mathbb{R} \wedge y^2 = x)\}$$

$$= \{0, 1, 2, 3, 4, \dots\} = \mathbb{N}$$

$$A4: U = \mathbb{Z}. \{x \mid \exists y(y \in \mathbb{R} \wedge y^3 = x)\} = \mathbb{Z}$$

$$A5: U = \mathbb{R}. \{|x| \mid x \in \mathbb{Z}\} = \mathbb{N}$$

$$A6: U = \mathbb{R}. \{|x|\} = \text{non-negative reals.}$$

العمليات الاساسية على المجموعات:

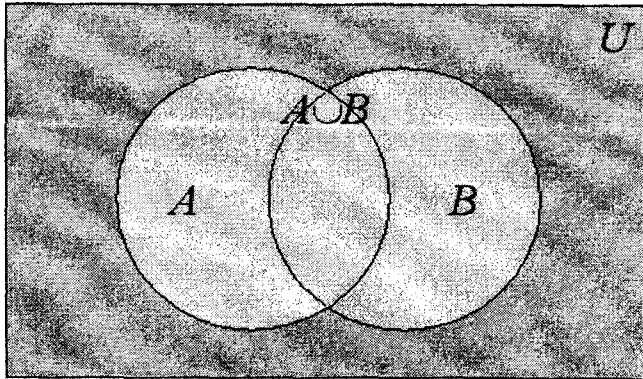
تنفذ على المجموعات العمليات الاساسية التالية:

امثلة:

1. الاتحاد
2. التقاطع
3. الفرق
4. النفي
5. الفرق المتماثل او عملية الاستبعاد

بمثل اتحاد مجموعتين مجموعة عناصرها هي عناصر المجموعة الاولى مضافا اليها عناصر المجموعة الثانية والتي لا تقع في المجموعة الاولى ويمكن تمثيل هذه العملية بمخططات فين كما يلي:

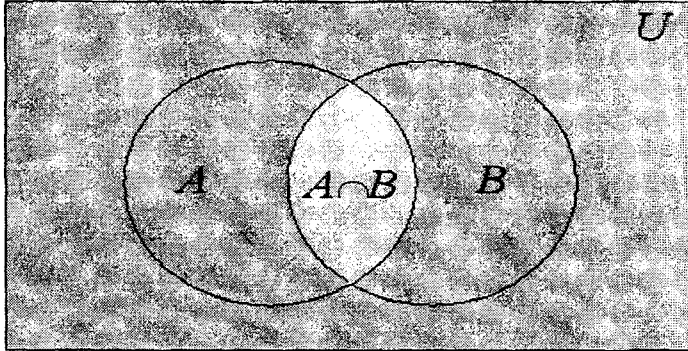
$$A \cup B = \{x \mid x \in A \vee x \in B\}$$





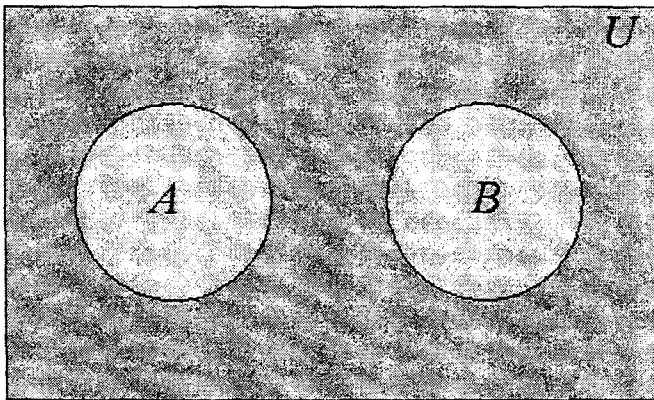
تقاطع مجموعتين هو مجموعة عناصرها هي العناصر التي تقع في المجموعة الاولى وفي نفس الوقت تقع في المجموعة الثانية وفيما يلي كيفية تمثيل هذه العملية:

$$A \cap B = \{x \mid x \in A \wedge x \in B\}$$



إذا لم تحتوي المجموعتان على عناصر مشتركة فإن تقاطعهما عبارة عن مجموعة خالية:

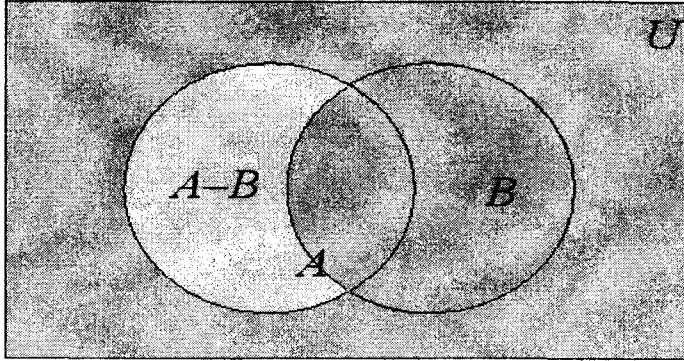
$$A \cap B = \emptyset .$$



الفرق:

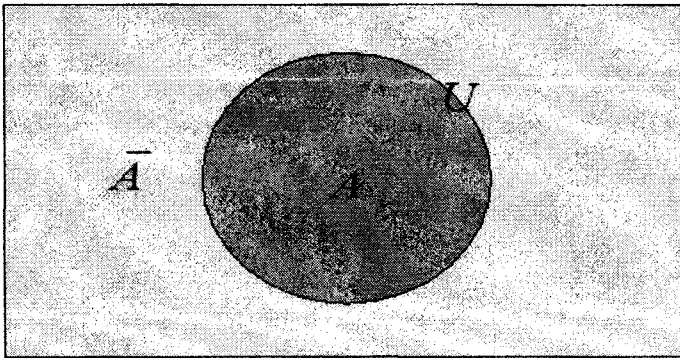
إذا طرحت مجموعة أولى من مجموعة ثانية فإن الناتج مجموعة عناصرها هي عناصر المجموعة الثانية والتي لا تقع في المجموعة الأولى:

$$A-B = \{x \mid x \in A \wedge x \notin B\}$$



نفي المجموعة هو مجموعة عبارة عن عناصر المجموعة الكاملة (العالمية) مطروحا منه عناصر المجموعة:

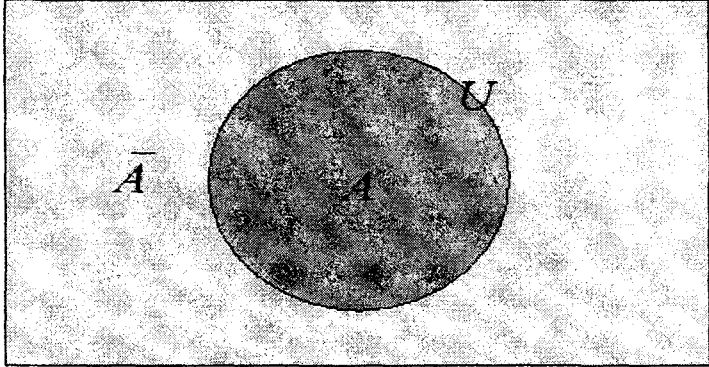
$$\bar{A} = \{x \mid x \notin A\}$$



الطرح المتماثل يعطي مجموعة عبارة عن حاصل جمع عناصر المجموعة

الأولى والثانية باستثناء العناصر المشتركة:

$$\bar{A} = \{x \mid x \notin A\}$$



امثلة:

Example

Let  $A = \{a, b, c\}$ ,  $B = \{1, 2, 3\}$  and  
 $C = \{a, c, e, 1, 3, 5\}$ .

then  $A \cup B = \{a, b, c, 1, 2, 3\}$

$B \cup C = \{1, 2, 3, a, c, e, 5\}$

$C \cup A = \{a, c, e, 1, 3, 5, b\}$

Example

Let  $A = \{a, b, c, d, e\}$ ,  $B = \{c, d, e, f, g\}$  and  $C = \{a, e, i, o, u\}$ .

$$\text{then } A \cap B = \{c, d, e\}$$

$$B \cap C = \{e\}$$

$$C \cap A = \{a, e\}$$

Example

Let  $S = \{a, b, c, d\}$  and  $T = \{c, d, e, f\}$ ,

$$\text{then } S \setminus T = \{a, b\}$$

$$T \setminus S = \{e, f\}$$

Example

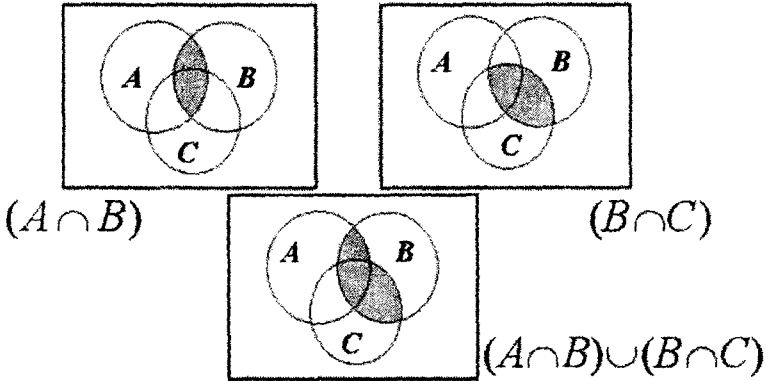
Let the universal set  $U$  be the set containing letters of the English alphabet and  $A = \{a, b, c, x, y, z\}$ .

then  $\bar{A} = \{d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w\}$

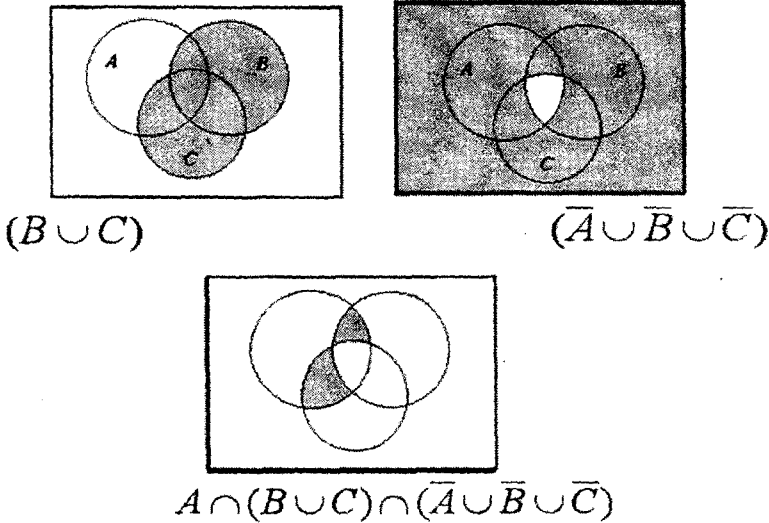
Example

Use Venn diagrams to represent the following set expressions.

(a)  $(A \cap B) \cup (B \cap C)$



(b)  $A \cap (B \cup C) \cap (\bar{A} \cup \bar{B} \cup \bar{C})$



قوانين المجموعات

تستخدم مجموعة من القوانين لتنفيذ العمليات المختلفة على المجموعات

وفيما يلي اهم هذه القوانين:

(1)  $\overline{\overline{A}} = A$  Law of Double Complement

(2)  $\overline{A \cup B} = \overline{A} \cap \overline{B}$  Demorgan's Laws

$\overline{A \cap B} = \overline{A} \cup \overline{B}$

(3)  $A \cup B = B \cup A$  Commutative Laws

$A \cap B = B \cap A$

(4)  $A \cup (B \cap C) = (A \cup B) \cap C$  Associative Laws

$A \cap (B \cup C) = (A \cap B) \cup C$

(5)  $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$  Distributive Laws

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

(6)  $A \cup A = A, A \cap A = A$  Idempotent Laws

(7)  $A \cup \phi = A, A \cap U = A$  Identity Laws

(8)  $A \cup \overline{A} = U, A \cap \overline{A} = \phi$  Inverse Laws

(9)  $A \cup U = U, A \cap \phi = \phi$  Domination Laws

(10)  $A \cup (A \cap B) = A$  Absorption Laws

$A \cap (A \cup B) = A$

امثلة:

Example

Let  $A, B$  and  $C$  be sets. Use laws of algebra of sets to simplify the following set expressions.

(a)  $(A \cap B) \cup (B \cap C)$

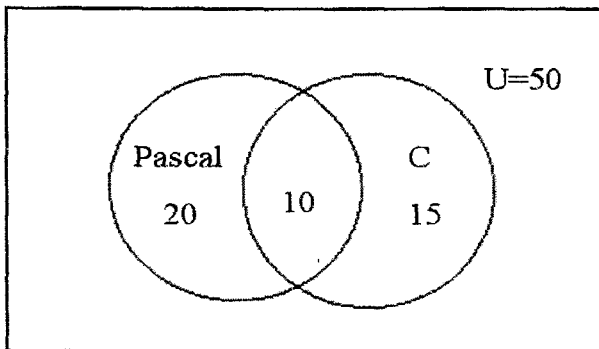
$= (A \cap B) \cup (C \cap B)$  Commutative law

$= (A \cup C) \cap B$  Distributive law

$$\begin{aligned}
 & (b) \quad (A \cap B) \cup (A \cap B \cap \bar{C} \cap D) \cup (\bar{A} \cap B) \\
 & = (A \cap B) \cup (\bar{A} \cap B) \cup (A \cap B \cap \bar{C} \cap D) \quad \text{Commutative law} \\
 & = ((A \cup \bar{A}) \cap B) \cup (A \cap B \cap \bar{C} \cap D) \quad \text{Distributive law} \\
 & = (U \cap B) \cup (A \cap B \cap \bar{C} \cap D) \quad \text{Inverse law} \\
 & = B \cup (A \cap B \cap \bar{C} \cap D) \quad \text{Identity law} \\
 & = B \quad \text{Absorption law}
 \end{aligned}$$

### Example

In a class of 50 college students, 30 study Pascal, 25 study C and 10 study both computer languages. How many students do not study computer language ?



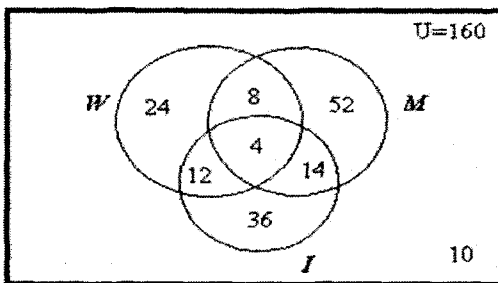
No. of students who do not study computer language is

$$50 - 20 - 10 - 15 = 5 \text{ students}$$

Example

In a survey of 160 passengers, an airline found that 48 preferred wine with their meals, 78 preferred mixed drinks, and 66 preferred ice tea. In addition, 12 enjoyed wine and mixed drinks, 18 enjoyed mixed drinks and ice tea, and 16 enjoyed ice tea and wine, and 4 passengers enjoyed them all.

- How many passengers want only iced tea with their meals?
- How many passengers do not like any of them?



a) No. of passengers = 36

b) No. of passengers

$$= 160 - 24 - 52 - 36 - 12 - 8 - 14 - 4$$

$$= 10$$

**Problem:**

Without using the Venn Diagrams, show that the symmetric difference operation  $\oplus$  satisfies the Associative Property. You may use basic properties of set operations (union, intersections, complementing) such as commutativity, associativity, distributivity and De Morgan's laws without proof.



**Solution:**

We need to prove for arbitrary sets  $A$ ,  $B$  and  $C$   
 $(A \oplus B) \oplus C = A \oplus (B \oplus C)$

Note that  $A \oplus B = (A \cap \bar{B}) \cup (\bar{A} \cap B)$  and  
 $\overline{(A \oplus B)} = (A \cap B) \cup (\bar{A} \cap \bar{B})$

Hence the left hand side =  $((A \oplus B) \cap \bar{C}) \cup \overline{(A \oplus B) \cap C}$   
 $=$   
 $((A \cap \bar{B}) \cup (\bar{A} \cap B)) \cap \bar{C} \cup (((A \cap B) \cup (\bar{A} \cap \bar{B})) \cap C)$   
 $=$   
 $(A \cap B \cap \bar{C}) \cup (\bar{A} \cap \bar{B} \cap \bar{C}) \cup (A \cap \bar{B} \cap C) \cup (\bar{A} \cap B \cap C)$

Similarly it can be shown that the right hand side is equal to this last expression. Thus the Associativity holds.

### 3.1 العلاقات :Relations

لتكن كل  $A, B$  مجموعة عندها فان العلاقة التي تربط هاتين المجموعتين هي حاصل الضرب الكارتيزي لهاتين المجموعتين.

مثال

لناخذ المجموعة التالية:

$$A = \{2, 3, 5, 6\}$$

العلاقة التي تربط عناصر هذه المجموعة بحيث يقسم العدد الثاني على الاول بدون باقى هي:

$$R = \{(2, 2), (3, 3), (5, 5), (6, 6), (2, 6), (3, 6)\}.$$

هذه ويمكن الرجوع لكتاب الرياضيات المنفصلة لمزيد من المعلومات عن العلاقات والاقترانات وما يهمن هنا هو القاء نظرة سريعة على اهم خصائص العلاقات الا وهي:

الانعكاس

R is reflexive if for every  $a \in A$ ,  $a R a$ .

التماثل

R is symmetric if for every a and b in A, if  $a R b$ , then  $b R a$ .

التعدي

R is transitive if for every a, b and c in A, if  $a R b$  and  $b R c$ , then  $a R c$ .

التساوي او التكافؤ

R is an equivalence relation on A if R is reflexive, symmetric and transitive

وفيما يلي بعض الامثلة التوضيحية:

1. In each case, a relation on the set  $\{1, 2, 3\}$  is given. Of the three properties, reflexivity, symmetry, and transitivity, determine which ones the relation has.

Give reasons for each of them.

- a.  $R = \{(1, 3), (3, 1), (2, 2)\}$ .  
 b.  $R = \{(1, 1), (2, 2), (3, 3), (1, 2)\}$ .

**Solution:**

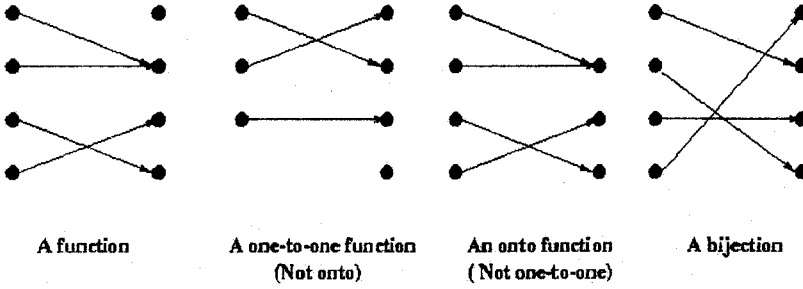
- a. It is Symmetry. Because wherever there is something like  $(a, b)$  we also have  $(b, a)$  Here it is  $(1, 3)$  we also have  $(3, 1)$  and  $(2, 2)$ .  
 b. It is reflexive because for every 'a' we have  $(a, a)$ . Here it is  $(1, 1), (2, 2), (3, 3)$ .
2. Three relations are given on the set of all non-empty subsets of  $N$ . In each case, say whether the relation is Reflexive or Symmetric or it is Transitive.
- a.  $R$  is defined by:  $A R B$  if and only if  $A \subseteq B$ .  
 b.  $R$  is defined by:  $A R B$  if and only if  $A \cap B$  is not equal to NULL.  
 c.  $R$  is defined by:  $A R B$  if and only if  $1 \in A \cap B$ .

**Solution:**

- a. It is reflexive .It is NOT symmetric .It is transitive.  
 b. It is reflexive .It is symmetric. It is transitive.  
 c. It is NOT reflexive .It is symmetric .It is transitive.

### 1.4 الاقتران

الاقتران هو علاقة تربط بين متغيرين او هدفين بحيث تعطي القيمة الواحدة (او اكثر من قيمة) من قيم المتغير الاول قيمة واحدة فقط من قيم المتغير الثاني اي ان العلاقة بين المتغير الاول والثاني تكون اما من النوع واحد لواحد او كثير لواحد وكما هو مبين في الشكل التالي:



وللمزيد من المعلومات عن الاقترانات يمكن الرجوع الى كتاب الرياضيات المنفصلة للمؤلفين.

### 1.5 التعبيرات المنتظمة Regular expressions

التعبير المنتظم هو تمثيل للغة مؤلفة من مجموعة من الرموز بحيث تستخدم هذه المجموعة من قيل الالات المنتهية والتي سوف نستعرضها في هذه الكتاب سواء لقبولها او رفضها.

وفيما نورد اهم خصائص هذه التعبيرات:

- التعبير المؤلف من رمز واحد يشار اليه بمجموعة مؤلفة من رمز واحد مثل:

$$C = \{ "c" \}$$

- التعبير الفارغ هو التعبير الذي لا يحتوي على رموز:

$$\epsilon = \{""\}$$

- اتحاد تعبيرين هو تعبير يعبر عنه كما يلي:

$$A+B = \{s \mid s \in A \text{ or } s \in B\}$$

- دمج تعبيرين هو الاخر تعبير يمثل كما يلي:

$$AB = \{ab \mid a \in A \text{ and } b \in B\}$$

- تكرار تعبير هو الاخر تعبير يمثل كما يلي:

$$A^* = \cup_{i \geq 0} A^i \text{ where } A^i = A \dots A \text{ (i times)}$$

$$A^* = \{\epsilon\} + A + AA + AAA + \dots$$

$$A^+ = A + AA + AAA + \dots = AA^*$$

وفيما اهم الرموز المستخدمة لبناء التعابير المنتظمة:

Symbol	Stands for...
.	any single character
x*	x, zero or more times
x+	x, one or more times
x?	x once, or not at all(optional x)
x{n}	x exactly n times
x{n,m}	x, at least n but not more than m times
x y	either x or y
xy	x followed by y

(x)	x as capturing group(more later)
[abc]	one of a or b or c, same as a b c
[^abc]	any character except a, b or c
[a-zA-Z]	a to z or A to Z(inclusive)

والجدول التالي يبين بعض الامثلة على التعبيرات المنتظمة:

<i>RE</i>	<i>Description</i>
$(0+1)^*111$	The set of strings containing only 0s and 1s that end in three consecutive 1s
$0^*1(0+1)^*$	The set of strings containing only 0s and 1s that have at least one 1
$0^*+0^*10^*$	The set of strings containing only 0s and 1s that have at most one 1
$\Sigma^*$	String of any characters
$\{a,\dots,z,A,\dots,Z\}(\{a,\dots,z,A,\dots,Z,0,\dots,9,\_ \})^*$	The set of identifiers in Pascal
$\Sigma\Sigma\Sigma\dots\Sigma(80 \text{ times})$	A line of 80 characters
$1^+$	A string of 1s, having at least one 1
$(\Sigma-\{a,e,i,o,u\})^*$	A string of letters not containing any vowel

وفيما يلي اهم القواعد المستخدمة للتعامل مع التعبيرات المنتظمة:

$$\alpha+\alpha \equiv \alpha$$

$$\alpha+\phi \equiv \alpha$$

$$\alpha\cdot\phi \equiv \phi\cdot\alpha \equiv \phi$$

$$\varepsilon\cdot\alpha \equiv \alpha\cdot\varepsilon \equiv \alpha$$

$$(\varepsilon + \alpha)^* \equiv \alpha^*$$

$$\varepsilon + \alpha\alpha^* \equiv \alpha^*$$

$$\varepsilon + \alpha^*\alpha \equiv \alpha$$

$$(\alpha\beta)^*\alpha \equiv \alpha(\beta\alpha)^*$$

$$\alpha\alpha^* \equiv \alpha^*\alpha$$

$$(\alpha^*\beta)^*\alpha^* \equiv (\alpha + \beta)^*$$

$$\alpha^*(\beta\alpha^*)^* \equiv (\alpha + \beta)^*$$

ويمكن استخدام هذه القواعد لاثبات تساوي التعابير المنتظمة وكما هو

موضح في المثال التالي:

Prove that  $0^* + 0^*1(\varepsilon + 00^*1)^* 000^* = \varepsilon + (0 + 10)^*0$

$$\text{LHS} = 0^* + 0^*1(\varepsilon + 00^*1)^* 000^*$$

$$= (\varepsilon + 00^*) + 0^*1(00^*1)^* 000^*$$

$$= (\varepsilon + 00^*) + 0^*10(0^*10)^* 00^*$$

$$= \varepsilon + (\varepsilon + 0^*10(0^*10)^*) 00^*$$

$$= \varepsilon + (0^*10)^* 0^*0$$

$$= \varepsilon + (0+10)^* 0$$

تستخدم لتوليد التعبيرات المتظمة او اللغة المؤلفة من مجموعة من الرموز المنتهية مجموعة من القواعد grammar المحددة والمعرفة سابقا وتمثل هذه القواعد رياضيا كما يلي:

A grammar  $G=(V, T, P, S)$

ويضم هذا النموذج:

- مجموعة منتهية من المتغيرات غير النهائية والتي يمكن ان تتفرع لتشكيل سلسل من الرموز الخاصة باللغة.
- مجموعة منتهية من المتغيرات النهائية والتي لا تتفرع والتي تشكل الرموز الداخلة فب اللغة.
- مجموعة قواعد التوليد والتي تشكل تشكل عملية الاستدعاء الذاتي لتوليد الرموز.
- رمز البداية للغة اوللتعبير المنتظم.

مثال:

Example:

Terminal: a

Non-terminal: S

Productions:  $S \rightarrow aS$

$S \rightarrow \epsilon$



من المثال السابق وباستخدام هذه القاعدة يمكن توليد اي تكرار من الحرف

المحدد.

مثال:

قاعدة توليد.  $\{alb\}$ :

$$P: S \rightarrow \varepsilon \mid a \mid b$$

$$S \rightarrow aSa$$

$$S \rightarrow bSb$$

$$G = (\{S\}, \{a,b\}, P, S)$$

مثال:

قاعدة توليد لغة او تعبير منتظم مؤلف من سلسلة فيها عدد متساو من

الاحرف  $a, b$

$$S \rightarrow aSbS \mid bSaS \mid \varepsilon$$

وفيما يلي بعض الامثلة التوضيحية:

### Example 1:

Terminal:  $a$

Nonterminal:  $S$

Productions:  $S \rightarrow aS$

$S \rightarrow \varepsilon$

The derivation for a<sup>4</sup> is:

$$S \Rightarrow aS$$

$$\Rightarrow aaS$$

$$\Rightarrow aaaS$$

$$\Rightarrow aaaaS$$

### Example 2:

Terminal: a

Nonterminal: S

Productions:  $S \rightarrow SS$

$$S \rightarrow a$$

$$S \rightarrow \varepsilon$$

Derivation of a<sup>2</sup> is as follows:

$$S \Rightarrow SS$$

$$\Rightarrow SSS$$

$$\Rightarrow SSa$$

$$\Rightarrow SSSa$$

$$\Rightarrow SaSa$$

$$\Rightarrow \varepsilon aSa$$

$$\Rightarrow \varepsilon a \varepsilon a = aa$$

Example 3:

Terminals: a, b

Nonterminals: S

Productions:

$S \rightarrow aS$

$S \rightarrow bS$

$S \rightarrow a$

$S \rightarrow b$

More compact notation:

$S \rightarrow aS \mid bS \mid a \mid b$

Derive abbab as follows:

$S \Rightarrow aS$

$\Rightarrow abS$

$\Rightarrow abbS$

$\Rightarrow abbaS$

$\Rightarrow abbab$

CFL is  $(a+b)^+$

**Example 4:**

Terminals: a, b

Nonterminals: S, X

Productions:

$$S \rightarrow XaaX$$

$$X \rightarrow aX \mid bX \mid \varepsilon$$

CFL is  $(a + b)^*aa(a + b)^*$

Derive abbaaba as follows:

$$S \Rightarrow XaaX$$

$$\Rightarrow aXaaX$$

$$\Rightarrow abXaaX$$

$$\Rightarrow abbXaaX$$

$$\Rightarrow abb\_aaX = abbaaX$$

$$\Rightarrow abbaabX$$

$$\Rightarrow abbaabaX$$

$$\Rightarrow abbaaba\_ = abbaaba$$

هذا ويمكن تمثيل مجموعة الانتاج والتوليد بالهيكل الشجري وكما هو

مبين في الامثلة التالية:

**Example 1: CFG:**

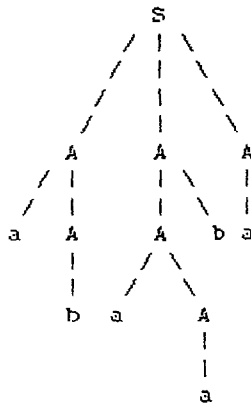
Terminals: a, b

Nonterminals: S, A

Productions:  $S \rightarrow AAA \mid AA$

$A \rightarrow AA \mid aA \mid Ab \mid a \mid b$

String abaaba has derivation tree:



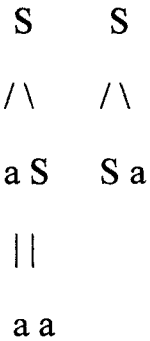
**Example2:**

Terminals: a, b

Nonterminals: S

Productions:  $S \rightarrow aS \mid SA \mid a$

The word aa can be generated by two different trees:



**Example 3:**

Terminals: a, b

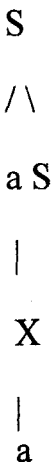
Nonterminals: S, X

Productions:  $S \rightarrow aS \mid aSb \mid X$

$X \rightarrow Xa \mid a$

The word aa has two different derivations that correspond to different syntax trees:

1.  $S \Rightarrow aS \Rightarrow aX \rightarrow aa$



الوحدة الثانية

آلة الحالة المنتهية

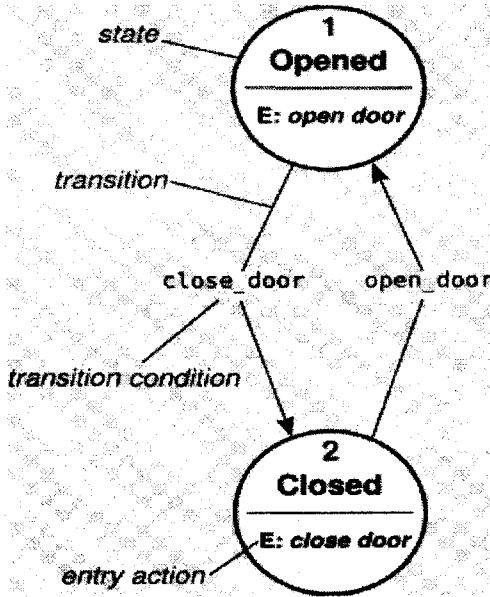
*Finite state  
machine(FSM)*

2

1.2 مقدمة

آلة الحالة المنتهية (FSM) or finite state automaton or state machine هي نموذج يبين سلوك الوحدات الذاتية المخصصة لحل مشكلة معينة وذلك اعتمادا على مجموعة من العوامل الداخلة في النموذج مثل مجموعة الحالات التي تقع فيها الآلة ومجموعة الانتقالات من حالة الى اخرى ومجموعة الافعال او الاحداث المولدة نتيجة لعملية الانتقال من حالة الى اخرى ويتاثير المدخلات المستخدمة في الآلة.

ويبين الشكل التالي نمودجا او مخططا لآلة الحالة والتي يمكن التعبير عنها بمجموعة العوامل التالية:



- مجموعة المدخلات.
- مجموعة الحالات ومن بينها الحالة الابتدائية.
- مجموعة المخرجات.



- دالة الانتقال والتي تفيد بنقل الآلة من حالة محددة الى حالة اخرى محددة اعتمادا على البيانات المتوفرة حاليا.

تستخدم الحالة لتخزين معلومات عن سلوك الآلة في الماضي وهي تعكس التغيرات الناجمة في الآلة نتيجة لقراءة او معالجة مجموعة من المدخلات.

ترتبط عملية انتقال الآلة من حالة الى اخرى بشرط او اكثر وعادة ما يرتبط الشرط بقيمة الحالة الحالية وقيم المدخلات الحالية ويتم التعبير عن دالة الانتقال من خلال جدول يشبه الى حد ما الجدول المبين ادناه:

*State transition table*

Current State -> Condition	State A	State B	State C
Condition X	...	...	...
Condition Y	...	State C	...
Condition Z	...	...	...

تستخدم الآلات المنتهية في كثير من التطبيقات وبشكل عام تصنف الآلات

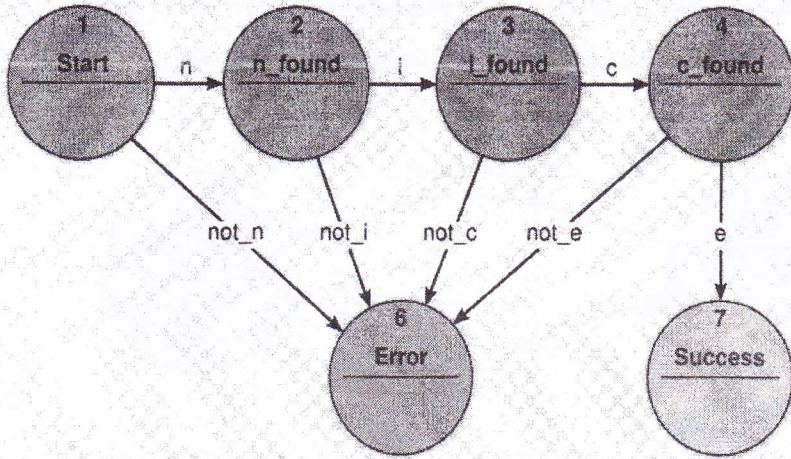
المنتهية الى:

### 1. الآلة المميزة Recognizer

وتستخدم هذه الآلة في الغالب لتمييز مجموعة من المدخلات والتعرف

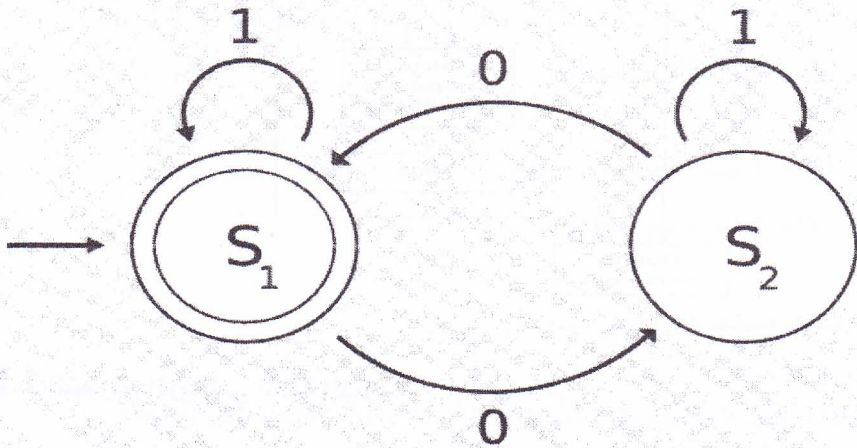
عليها او بمعنى اخر التعرف على نمط معين من البيانات تشكل جزءا من المدخلات

والمثال التالي يبين مخطط الحالات لآلة منتهية تعمل على تمييز الكلمة nice

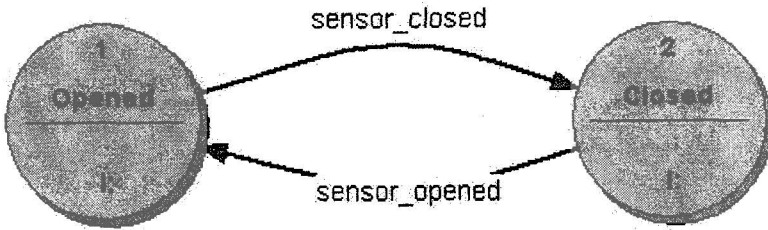


2. الآلة المنتهية التي تقبل مجموعة من المدخلات Acceptors

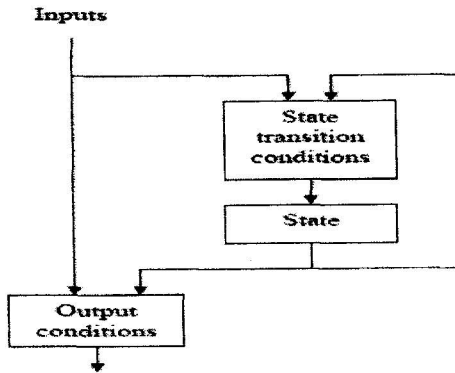
ويقبل هذا النوع من الآلات مجموعة من المدخلات والمثال التالي يبين مخطط آلة منتهية تحدد فيما إذا كان الرقم الثنائي زوجي أم فردي.



ومن الامثلة على الالات المنتهية المتحسسات والتي يمكن ان تتاثر بالمدخلات لتبقى في نفس الحالة او تنتقل الى حالة جديدة وكما هو مبين في الشكل التالي:



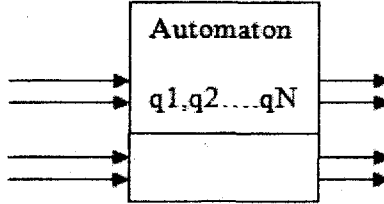
وبشكل عام فان الآلة المنتهية ما هي الا وحدة تقبل مجموعة من المدخلات وتنتقل من حالة الى حالة اخرى او يمكن ان تبقى في نفس الحالة اعتمادا على قيم المدخلات والحالة الحالية وعند الانتقال فانها تعمل على توليد بعض المخرجات ويبين الشكل التالي اهم مكونات الآلة المنتهية اعتمادا على هذا التعريف:



## 2.2 النموذج الرياضي للآلة المنتهية:

كما اشرنا سابقا فان الآلة المنتهية ما هي الا وحدة تحكم تحتفظ بمعلومات عن سلوك الآلة في الماضي بناء على المدخلات التي تمت قراءتها وتشكل مجموعة الحالات حالة الآلة المنتهية واعتمادا على الحالة الحالية والبيانات الحالية المقروءة فان الآلة يمكن ان تنقل الى حالة جديدة او تبقى في نفس الحالة

منتجة بذلك الخرجات اذا لزم الامر او تطلب الامر من ان تقوم الوحدة الذاتية بانتاج مخرجات والشكل التالي يبين نموذج الآلة المنتهية:



وبناء على ما تقدم يمكن وصف الآلة المنتهية بما يلي:

- مجموعة المدخلات.
- مجموعة المخرجات (ان وجدت).
- مجموعة الحالات بما فيها الحالة الابتدائية والحالات النهائية.
- دالة الانتقال.

هناك نوعان من الآلات المنتهية:

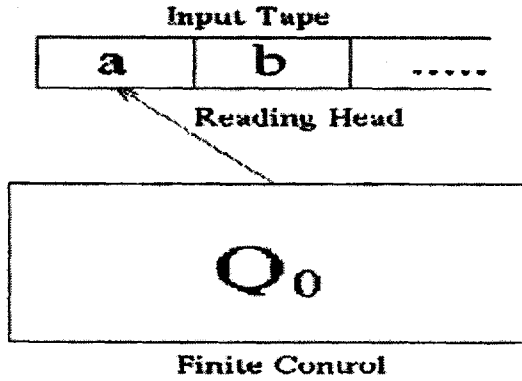
1. الآلة المنتهية المحدودة.
2. الآلة المنتهية غير المحدودة.

وفيما سوف نستعرض هذين النوعين.

### 1. الآلة المنتهية المحدودة Deterministic Finite Automata

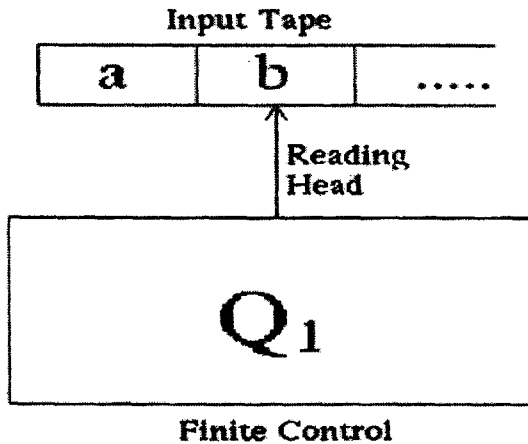
يمكن تصور هذه الآلة كما هو مبين في الشكل التالي على انها وحدة مؤلفة

من ما يلي:



- شريط المدخلات والمؤلف من مجموعة من الرموز.
- وحدة التحكم والتي تحتفظ بمجموعة الحالات.
- رأس القراءة والكتابة والمخصص فقط للقراءة (قراءة المدخلات بدون كتابة المخرجات).
- يتحرك رأس القراءة فقط باتجاه اليمين وبعد كل عملية قراءة ينتقل رأس القراءة لموقع.

واحد فقط باتجاه اليمين وكما هو مبين في الشكل التالي:



تمتلك آلة الحالة المحدودة حالة نهائية واكثر ويجب ان ينتهي تنفيذها في احدى الحالات النهائية.

يتم وصف الآلة المنتهية رياضيا بالنموذج التالي:

$$FA = \langle Q, I, \delta, q_0, F \rangle$$

والذي يشمل:

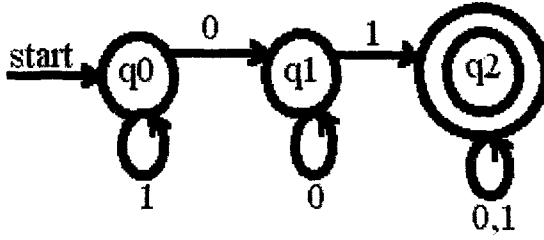
- مجموعة محدودة من الحالات.
- مجموعة منتهية من رموز المدخلات.
- دالة الانتقال والتي تاخذ المعاملات الممثلة بالحالة الحالية والرمز المقروء للانتقال الى الحالة الجديدة.
- الحالة الابتدائية.
- مجموعة الحالات المقبولة او مجموعة الحالات النهائية والمنتمية الى مجموعة الحالات الكلية.

يتم في المخطط التعبير عن الحالة بالاثرة اما الانتقال من حالة الى اخرى فيعبر عنها السهم على ان يكتب عليه الرمز المقروء.

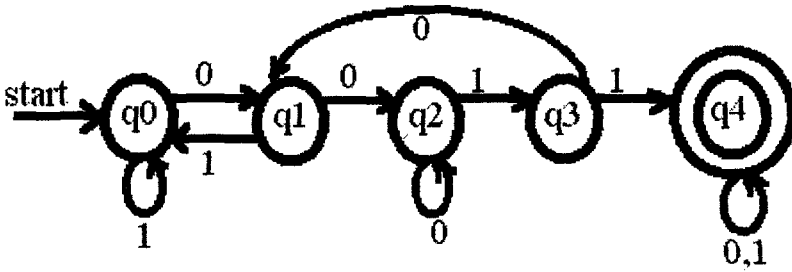
تستخدم الآلات المنتهية المحدودة لتمييز النمط في سلسلة الرموز او تعمل على اكتشاف تسلسل معين لمجموعة من الرموز في سيل المدخلات.

امثلة:

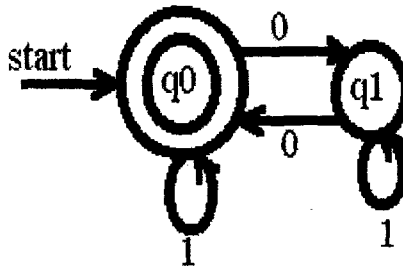
1. ابن الآلة المنتهية المحدودة والتي تعمل على اكتشاف الصفر متبوعا بالواحد في مجموعة من المدخلات المؤلفة من الصفر والواحد:



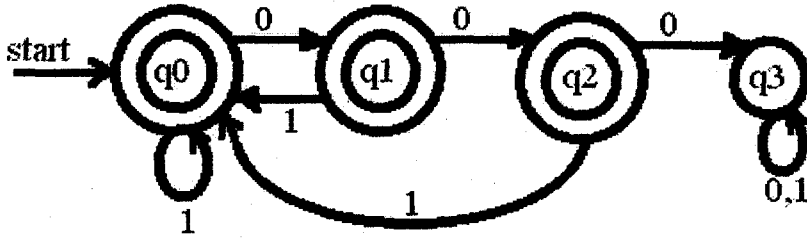
2. ابن الآلة المنتهية المحدودة والتي تعمل على اكتشاف صفرين متتابعين متبوعين بواحدين متتابعين (أي اكتشاف السلسلة الجزئية 0011).



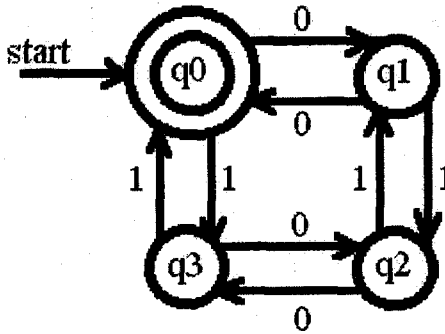
3. ابن الآلة المنتهية والتي تقبل سلسلة من الرموز مؤلفة من عدد زوجي من الازرار واي عدد من الوحدات:



4. ابن الآلة المنتهية والتي تقبل كافة مجموعة الرموز باستثناء 3 وحدات متتابعة:



5. ابن الآلة المنتهية التي تقبل عدد زوجي من الأصفار وعدد زوجي من الوحدات.



6. ابن آلة الحالة المنتهية والتي تحقق العلاقة: باقي قسمة  $2 = 5$  على  $5$  هي:

لاحظ هنا ان مجموعة المدخلات التي تقود الى الحالة النهائية هي:

10

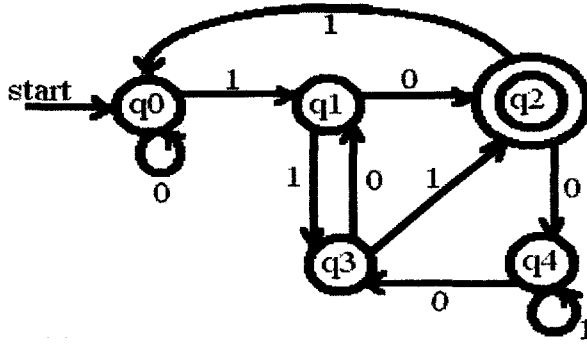
111

1100

10110

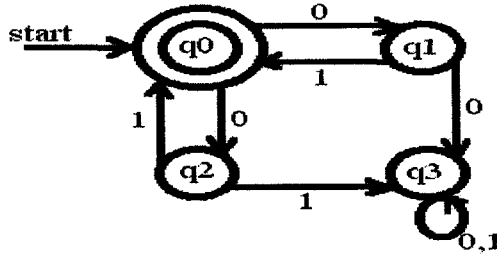


وغيرها الكثير وحسب ما هو مبين في مخطط الآلة التالي:



7. ابن مخطط آلة الحالة المنتهية التي تقبل تكرار الصفر والواحد  $(0+1)^*$

بعدد متساو من الأصفار والوحدات علما بأن كل بداية يجب أن تحتوي على الأكثر على صفر إضافي عن الوحدات أو على الأكثر واحد إضافي عن عدد الأصفار:



مما تقدم يمكن النظر إلى آلة الحالة المنتهية كمعدات (وسوف نتطرق إلى

هذا لاحقا في هذا الكتاب إن شاء الله) مكونة من الأجزاء التالية:-

- مسجل داخلي.
- مجموعة من القيم التي تكتب في المسجل.
- شريط الرموز.
- رأس القراءة.
- مجموعة من التعليمات والمثلة لدالة الانتقال.

## 3.2 اهم المصطلحات:

1. اللغة المقبولة من قبل الالة المنتهية هي مجموعة الرموز المقروءة والتي تؤدي قراءتها الى الانتقال من الحالة الابتدائية الى احدي الحالات النهائية.

فمثلا اذا كانت مجموعة الرموز مؤلفة من:

$$\Sigma = \{a, b\}$$

فهناك مجموعة من الحالات لبناء الالات المنتهية والتي تقب اللغات المؤلفة من نماذج السلسل الحرفية التالية:

## Strings

$a$

$ab$

$abba$

$baba$

$aaabbbaabab$

$u = ab$

$v = bbbaaa$

$w = abba$

2. تطبق على السلاسل الرمزية مجموعة من العمليات اهمها الدمج والقراءة العكسية وتحديد طول السلسلة الرمزية كما هو مبين في الامثلة التالية:

$$w = a_1 a_2 \cdots a_n$$

$$v = b_1 b_2 \cdots b_m$$

$$w = abba$$

$$v = bbbaaa$$

Concatenation

$$wv = a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m$$

$$wv = abbabbbaaa$$

Reverse

$$v^R = b_m \cdots b_2 b_1$$

$$v^R = aaabbb$$

$$w = a_1 a_2 \cdots a_n$$

Length:  $|w| = n$ 

Examples:

$$|abba| = 4$$

$$|aa| = 2$$

$$|a| = 1$$

For any letter:  $|a| = 1$

For any string  $wa$ :  $|wa| = |w| + 1$

Example:  $|abba| = |abb| + 1$   
 $= |ab| + 1 + 1$   
 $= |a| + 1 + 1 + 1$   
 $= 1 + 1 + 1 + 1$   
 $= 4$

$$|uv| = |u| + |v|$$

Example:  $u = aab, |u| = 3$

$$v = abaab, |v| = 5$$

$$|uv| = |aababaab| = 8$$

$$|uv| = |u| + |v| = 3 + 5 = 8$$

3. السلسلة بدون رموز هي سلسلة فارغة ويمكن استخدامها في بعض الاحيان

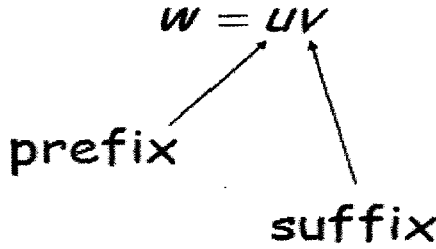
لنقل الالة من حالة الى اخرى دون الحاجة الى مدخلات:

$$|\lambda| = 0$$

$$\lambda W = W\lambda = W$$

$$\lambda abba = abba\lambda = abba$$

4. إذا كانت  $W = uv$  مجموعة الرموز المقبولة من قبل الآلة المنتهية فإن السلسلة الأولى تنفذ أولاً وتدعى سلسلة البداية أما السلسلة الثانية فتتنفذ ثانياً وتدعى السلسلة البعيدة



وإذا ضمت السلسلة مجموعة الأحرف  $abba$  فإنها ستتنفذ بقراءة حرف حرف ومن اليسار إلى اليمين كما يلي:

**Prefixes**

- $\lambda$
- $a$
- $ab$
- $abb$
- $abba$
- $abbab$

**Suffixes**

- $abbab$
- $bbab$
- $bab$
- $ab$
- $b$
- $\lambda$

5. قد تتكرر عملية قراءة مجموعة الرموز وفي هذه الاحالة تستخدم السلسلة الرمزية مرفوعة لقوة تساوي عدد مرات التكرار وكما هو مبين ادناه:

$$w^n = \underbrace{ww \dots w}_n$$

$$(abba)^2 = abbaabba$$

for any :

$$w \quad w^0 = \lambda$$

$$(abba)^0 = \lambda$$

6. تستخدم النجمة مع مجموعة الرموز للاشارة الى كافة السلاسل الرمزية التي يمكن تكوينها من السلسلة الرمزية بما في ذلك السلسلة الرمزية الفالرغة اما اشارة الزائد فتشير الى كافة السلاسل الرمزية التي يمكن الحصول عليها من السلسلة الاصلية باستثناء السلسلة الفارغة:

Example:

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$$

$$\Sigma^+ = \Sigma^* - \lambda$$

$$\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, aab, \dots\}$$

7. اللغة المقبولة من آلة الحالة المنتهية هي سلسلة جزئية تنتمي إلى مجموعة السلاسل الرمزية التي يمكن تكوينها من مجموعة الرموز:

Examples:

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$$

Language  $L(M)$ :-

$$\{\lambda, \{a, aa, aab\}$$

$$\{\lambda, abba, baba, aa, ab, aaaaaa\}$$

8. اللغة غير المنتهية هي مجموعة الرموز المقروءة وغير المنتهية خاصة عندما يكون هناك تكرار في آلة الحالة المنتهية من خلال الوصول إلى حالة نهائية ثم الخروج منها والعودة إليها:

An infinite language

$$L = \{a^n b^n : n \geq 0\}$$

$$\left. \begin{array}{l} \lambda \\ ab \\ aabb \\ aaaaabbbb \end{array} \right\} \in L \quad abb \notin L$$

9. تنفذ على اللغات المقبولة من الآلات المنتهية مجموعة من العمليات أهمها الاتحاد والتقاطع والفرق والمكمل تماما كما تنفذ هذه العمليات على المجموعات:

The usual set operations

$$\{a, ab, aaaa\} \cup \{bb, ab\} = \{a, ab, bb, aaaa\}$$

$$\{a, ab, aaaa\} \cap \{bb, ab\} = \{ab\}$$

$$\{a, ab, aaaa\} - \{bb, ab\} = \{a, aaaa\}$$

Complement:

$$\bar{L} = \Sigma^* - L$$

$$\overline{\{a, ba\}} = \{\lambda, b, aa, ab, bb, aaa, \dots\}$$

10. اللغة المعكوسة تؤخذ من مجموعة الرموز المشكلة للغة الاصلية وتقرأ بشكل معكوس:

Definition:

$$L^R = \{w^R : w \in L\}$$

Examples:

$$\{ab, aab, baba\}^R = \{ba, baa, abab\}$$

$$L = \{a^n b^n : n \geq 0\}$$

$$L^R = \{b^n a^n : n \geq 0\}$$



11. ناتج دمج لغتين هو لغة مقدمتها من اللغة الاولى ونهايتها من اللغة الثانية:

Definition:

$$L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$$

Example:

$$\{a, ab, ba\}\{b, aa\}$$

$$= \{ab, aaa, abb, abaa, bab, baaa\}$$

هذا ويمكن دمج اللغة الواحدة لتكرار قراءتها اكثر من مرة:

Definition:

$$L^n = \underbrace{LL \dots L}_n$$

Example:

$$\{a, b\}^3 = \{a, b\}\{a, b\}\{a, b\} =$$

$$\{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

Special case:

$$L^0 = \{\lambda\}$$

$$\{a, bba, aaa\}^0 = \{\lambda\}$$

Example:

$$L = \{a^n b^n : n \geq 0\}$$

$$L^2 = \{a^n b^n a^m b^m : n, m \geq 0\}$$

$$aabbbaabbb \in L^2$$

12. التغطية او تغطية النجمة هي مجموعة اللغات التي يمكن تغطيتها  
او الوصول اليها من لغة محددة:

Star-Closure (Kleene \*)

Definition:

$$L^* = L^0 \cup L^1 \cup L^2 \dots$$

Example:

$$\{a, bb\}^* = \left\{ \begin{array}{l} \lambda, \\ a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \dots \end{array} \right\}$$

اما التغطية الموجبة في التغطية الكلية للغة مستثنيا منها مجموعة

الرموز الفارغة:

Positive Closure

Definition:

$$L^+ = L^1 \cup L^2 \cup \dots$$

$$= L^* - \{\lambda\}$$

$$\{a, bb\}^+ = \left\{ \begin{array}{l} a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \dots \end{array} \right\}$$

13. تصنف الآلة المنتهية الى الآلة المنتجة للمخرجات حيث يتم في هذا النوع من

الات الحالة المنتهية انتاج المخرجات وتعتمد قيمة الخرج المنتج على الحالة

الحالية للالة وقيمة الدخل المقروء وكذلك الحال بالنسبة للحالة القادمة

والتي تعامل كاقتران يعتمد على الدخل والحالة الحالية كما هو الحال في

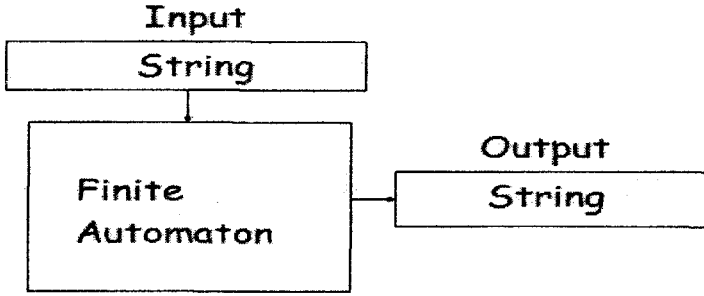
هذا وسوف نستعرض هذا النوع من الآلات لاحقا وسوف نستعرض نماذج من

هذه الآلات مثل الة موور والتي يعتمد فيها الخرج على الدخل والحالة

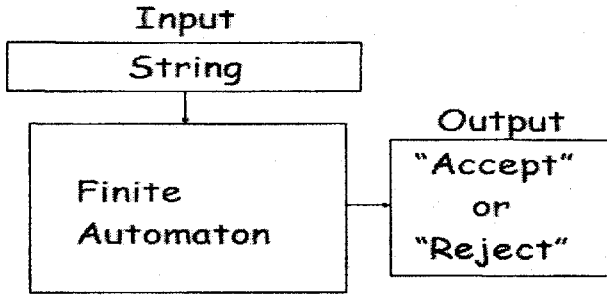
الحالية والة ميلي والتي يعتمد فيها الخرج على الحالة الحالية ومن اشهر

انواع المعدات المثلة لهذا النوع من الآلات المتحسسات واشكل التالي يبين

المخطط الصندوقي للالة المنتهية المنتجة للمخرجات:

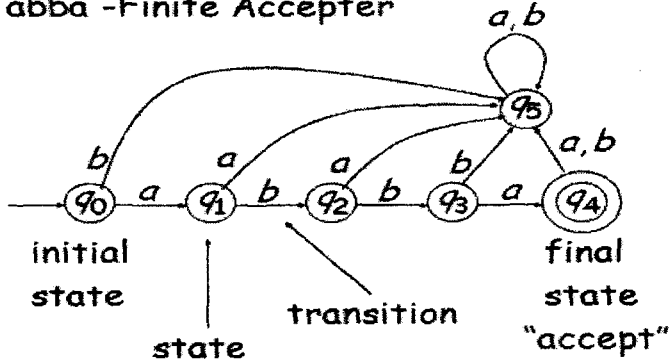


اما النوع الثاني من الآلات المنتهية فهو الذي نحن بصددده في هذه الوحدة الا وهو الآلة المميزة وهي آلة حالة منتهية تعمل على تمييز مجموعة من الرموز تشكل لغة الآلة بحيث تقود قراءتها الى حالة نهائية او تعمل على رفض مجموعة من الرموز والتي تقود قراءتها الى الانتقال الى حالة غير نهائية والشكل التالي يبين هذا النوع من الآلات:



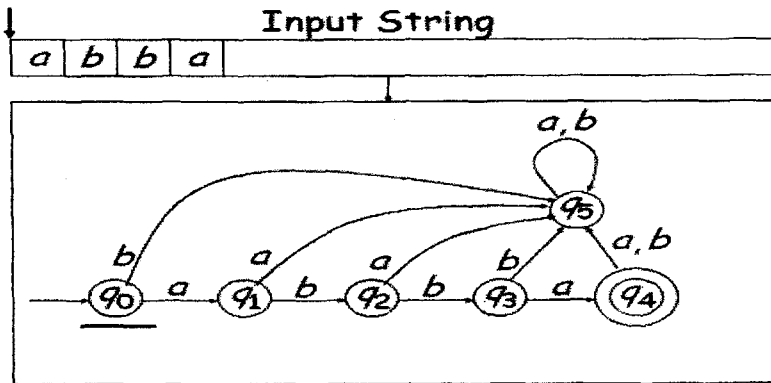
14. تستخدم مجموعة من الرموز لتمثيل الآلة المنتهية بواسطة مخطط الحالات وهذه الرموز مبينة في الشكل التالي:

abba - Finite Acceptor

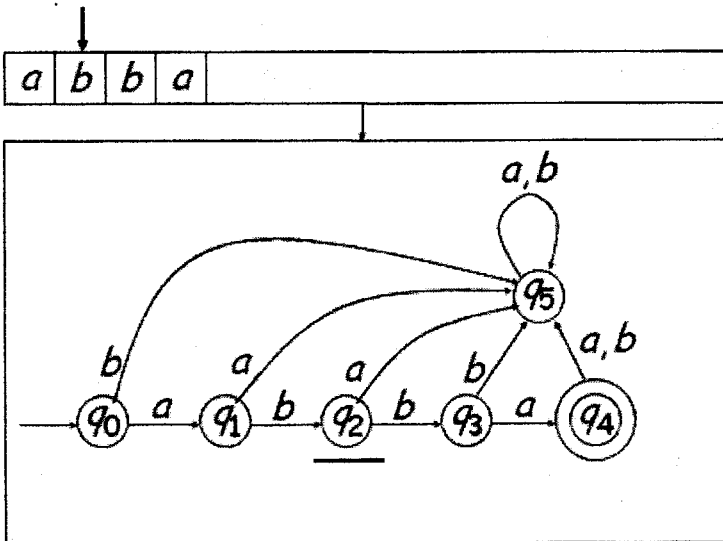
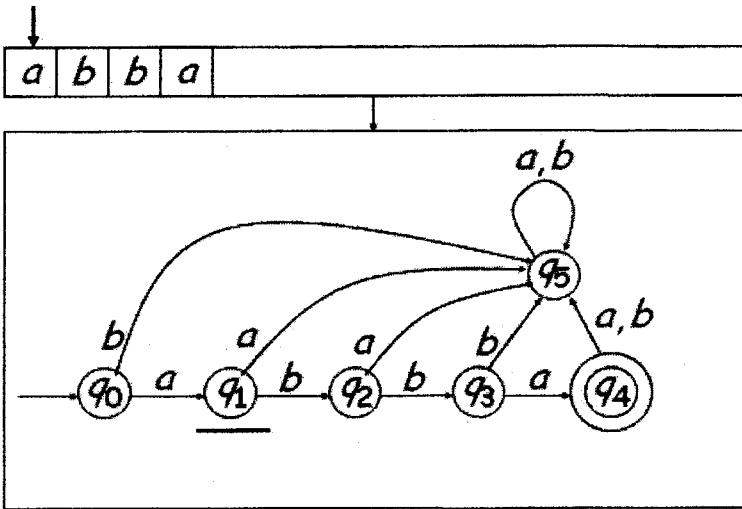


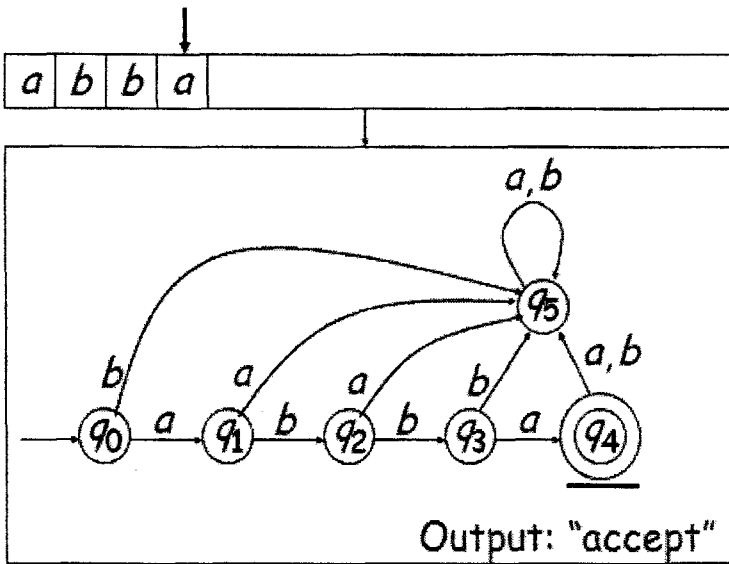
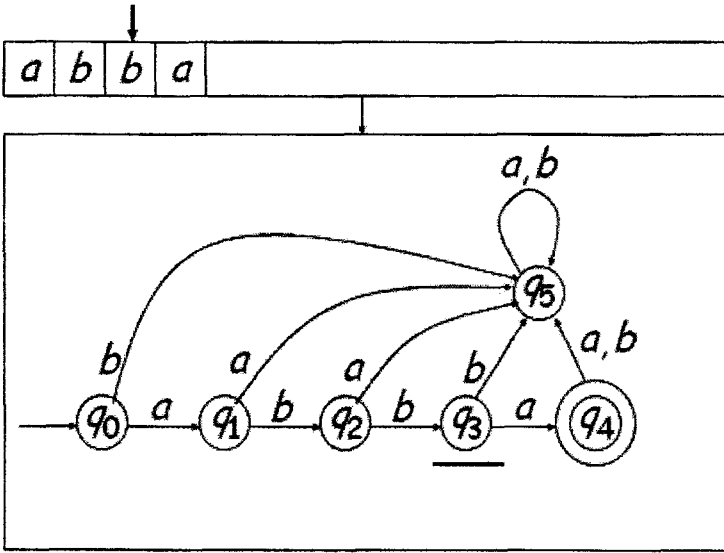
15. تعبر هيئة الآلة Configuration عن الحالة الحالية ومجموعة الرموز التي لم تقرا بعد ويمكن للآلة أن تنتقل من هيئة إلى أخرى بعد قراءة رمز من مجموعة الرموز المسجلة على الشريط هذا ويمكن التعبير عن الهيئة باستخدام المخطط أو باستخدام عملية التمثيل الرياضي وفيما يلي مجموعة من الأشكال والتي تبين كيفية الانتقال من هيئة إلى أخرى في حالة قبول اللغة وتمييز مجموعة من الرموز:

Initial Configuration



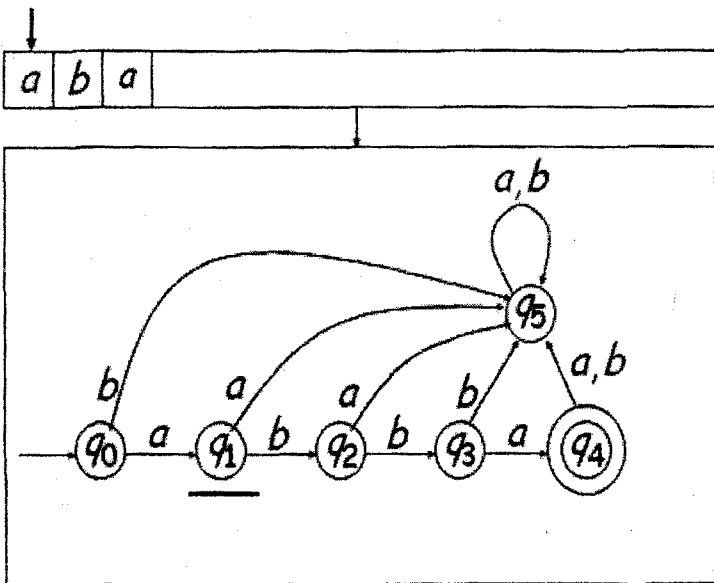
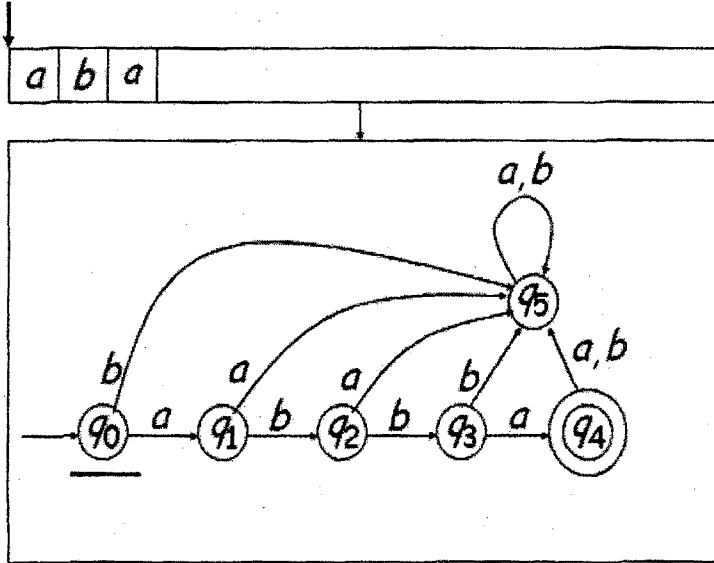
Reading the Input



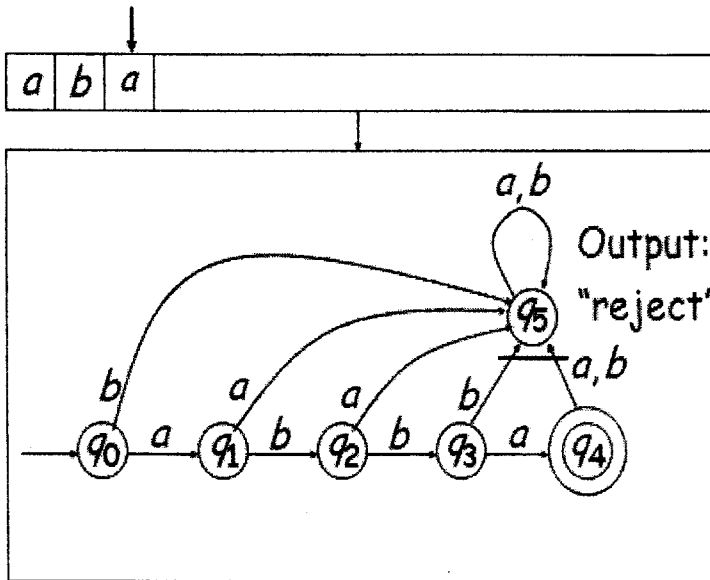
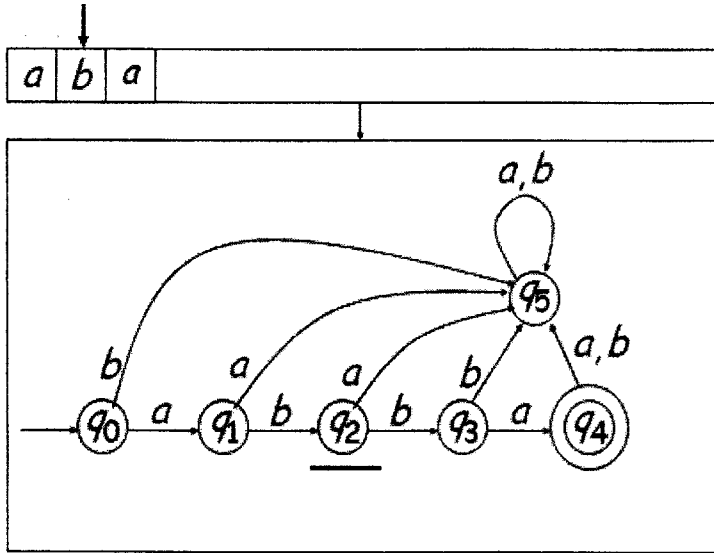


والخطوات التالية تبين عملية رفض اللغة او عدم قبولها:

Rejection

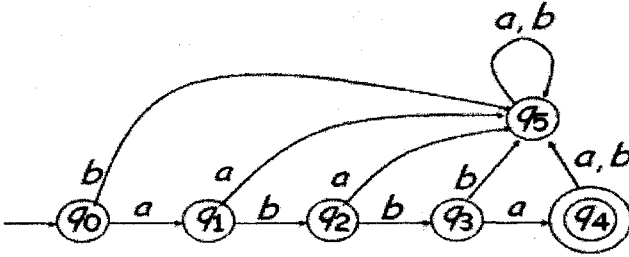




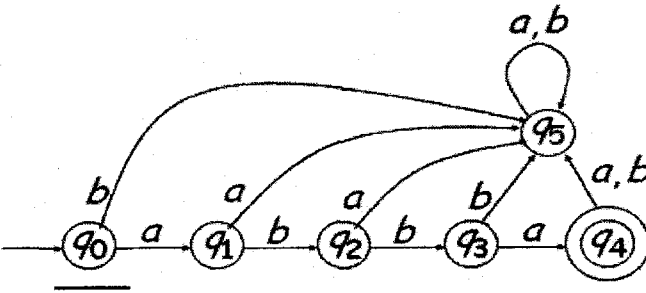


16. يتم وصف آلة الحالة المنتهية باستخدام مخطط الحالات او جدول الانتقال او الهيئة وفيما يلي بعض الامثلة التوضيحية على تمثيل الآلة المنتهية:

$$\Sigma = \{a, b\}$$

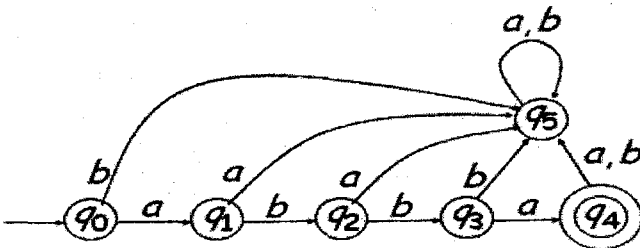


Initial State  $q_0$



Set of Final States  $F$

$$F = \{q_4\}$$

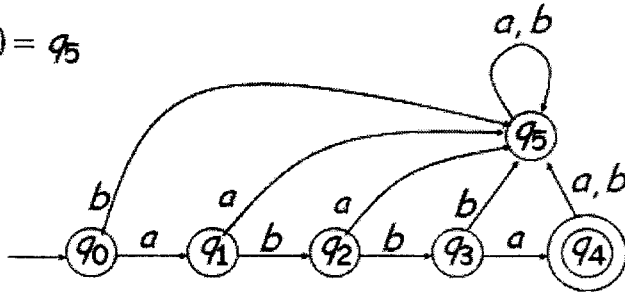


Transition Function  $\delta$ 

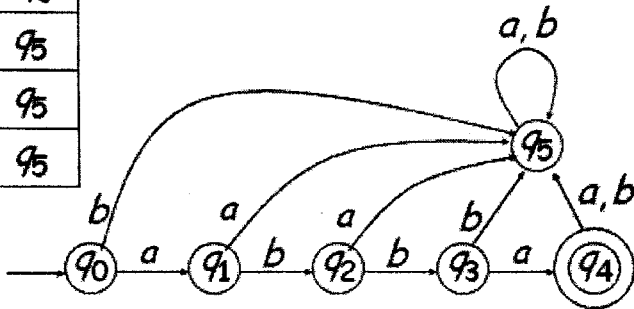
$$\delta: Q \times \Sigma \rightarrow Q$$

$$\delta(q_0, a) = q_1$$

$$\delta(q_0, b) = q_5$$

Transition Function  $\delta$ 

$\delta$	a	b
q0	q1	q5
q1	q5	q2
q2	q2	q3
q3	q4	q5
q4	q5	q5
q5	q5	q5



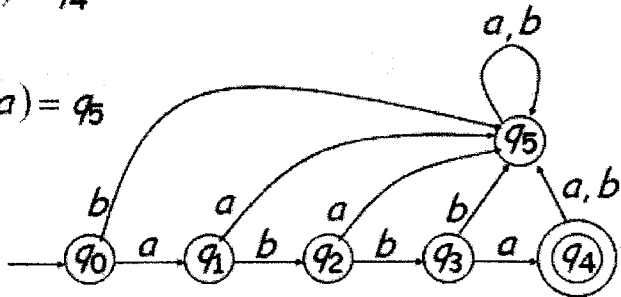
Extended Transition Function  $\delta^*$ 

$$\delta^* : Q \times \Sigma^* \rightarrow Q$$

$$\delta^*(q_0, ab) = q_2$$

$$\delta^*(q_0, abba) = q_4$$

$$\delta^*(q_0, abbaa) = q_5$$



## Recursive Definition

$$\delta^*(q, \lambda) = q$$

$$\delta^*(q, wa) = \delta(\delta^*(q, w), a)$$

$$\delta^*(q_0, ab) =$$

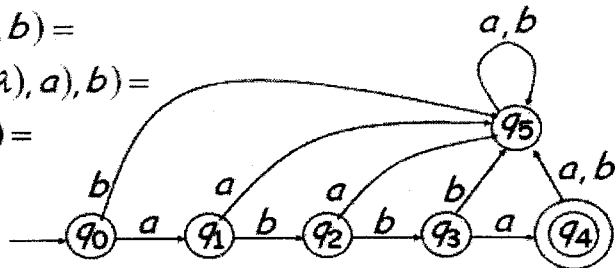
$$\delta(\delta^*(q_0, a), b) =$$

$$\delta(\delta(\delta^*(q_0, \lambda), a), b) =$$

$$\delta(\delta(q_0, a), b) =$$

$$\delta(q_1, b) =$$

$$q_2$$



## 4.2 اللغة المقبولة من آلة الحالة المنتهية

تكتب دالة الانتقال بالصورة التالية:

$$\delta(q, a) = p$$

والتي تعني الانتقال من الحالة  $q$  الى الحالة  $p$  عند قراءة الرمز  $a$  والحرف  $a$  وهناك اصطلاح اخر لوصف هيئة الآلة ويكتب هذا الاصطلاح كما يلي:

$$[q_i, aw] \vdash [q_j, w]$$

حيث يشير الطرف الايسر الى الحالة الحالية والرمز الذي يقف عنده راس القراءة ومجموعة الرموز المتبقية والتي لم تقرا بعد اما الطرف الايمن فيشير الى الحالة بعد قراءة الرمز ومجموعة الرموز المتبقية.

والصورة السابقة للهيئة هي مكافئة للصيغة التالية:

$$[q_i, aw] \vdash [\delta(q_i, a), w], \text{ where } \delta(q_i, a) = q_j$$

تقبل آلة الحالة المنتهية مجموعة الرموز:

$$w = a_1 a_2 \dots a_n$$

اذا كان هناك تتابع من الانتقالات بحيث:

- تبدأ بحالة ابتدائية.
- تنتهي بحالة مقبولة.
- تمتلك مجموعة من الرموز من المجموعة الكلية للرموز مساوي ل

$$a_1, a_2, \dots, a_n = w.$$

هذا ويمكن توسعة دالة الانتقال لتشير الى الحالة النهائية كما يلي:

$$\hat{\delta}(q, w),$$

والتي تعني نقل الالة من الحالة المحددة بقراءة مجموعة الرموز المحددة واذا كانت مجموعة الرموز خالية فان الانتقال يتم فوريا وهنا يمكن ان نستخدم لامدا او ايبسلون للتعبير عن عملية الانتقال دون الحاجة الى وجود مدخلات:

$$\text{If } |w| = 0, \text{ then } \hat{\delta}(q, \lambda) = q$$

اما اذا كانت مجموعة الرموز تشكل رمزا واحدا فتكتب دالة الانتقال كما يلي:

$$\text{If } |w| = 1, \text{ then } \hat{\delta}(q, a) = \delta(q, a).$$

اما اذا كانت مجموعة الرموز اكث من رمز فيمكن التعبير عن دالة الانتقال كما يلي:

$$\text{Let } |w| \text{ be } n > 1.$$

Then  $w = ua$  and  $\hat{\delta}(q, ua) = \delta(\hat{\delta}(q, u), a)$ ,  
where  $a$  is a single symbol

وبهذا فانه لالة المنتهية التالية:

$$M = (Q, \Sigma, \delta, q_0, F)$$

تكون مجموعة الرموز المقبولة:

String  $w$  if  $\hat{\delta}(q_0, w)$  is in  $F$ .

اما اللغة المقبولة من الالة فهي مجموعة الرموز المقروءة والتي تؤدي فراءتها لنقل الالة من حالة الى حالة مقبولة او نهائية ويعبر عن اللغة كما يلي:

$$L(M) = \{w \mid \delta^*(q_0, w) \in F\}.$$

وفيما يلي التعريف الرياضي للغة المقبولة من قبل الوحدة المنتهية:

For a DFA  $M = (Q, \Sigma, \delta, q_0, F)$

Language accepted by  $M$  :

$$L(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \in F\}$$

alphabet
transition function
initial state
final states

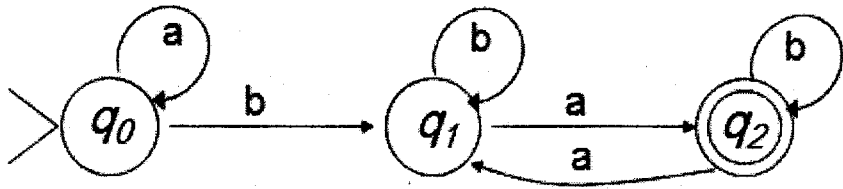
يمكن استخدام ايضا جدول الانتقال او دالة الانتقال لتمثيل الة الحالة المنتهية ولناخذ المثال التوضيحي التالي:

استخدم البيانات التالية لرسم مخطط الحالات لالة المنتهية:

- Let  $M = \{Q, \Sigma, \delta, q_0, F\}$  be a DFA.
- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = \{a, b\}$
- $F = \{q_2\}$

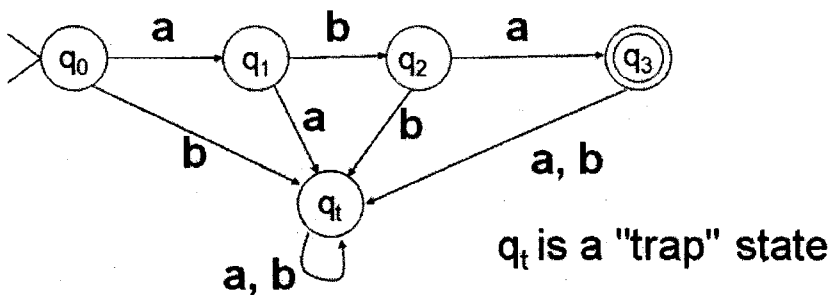
$\delta$	a	b
$q_0$	$q_0$	$q_1$
$q_1$	$q_2$	$q_1$
$q_2$	$q_1$	$q_2$

الحل:



كما اشرنا سابقا فان اللغة المقبولة من الآلة هي مجموعة الرموز التي تؤدي قراءتها الى الانتقال الى حالة نهائية وفي بعض الاحيان فان بعض الرموز يمكن ان تستثنى من اللغة وفي هذه الحالة تؤدي قراءة مثل هذه الرموز الى الوقوع في حالة تسمى حالة الفخ او حالة ما يسمى رفض الرموز وكما هو مبين في الشكل

التالي:





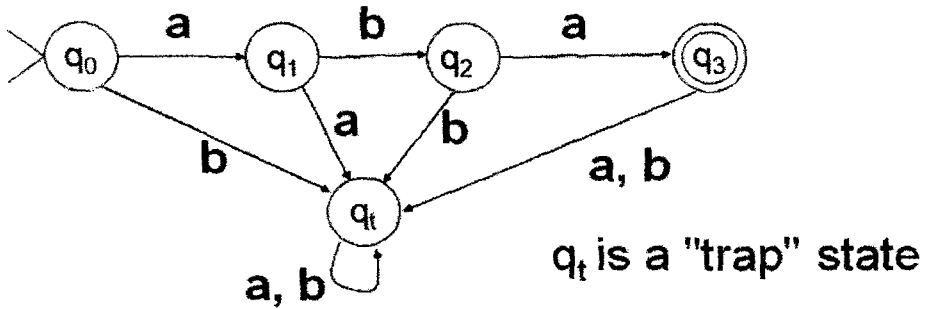
ولإيجاد اللغة المقبولة من قبل آلة الحالة المنتهية يمكن اتباع الخطوات التالية:

- اوجد التعبيرات المنتظمة  $u_1, \dots, u_n$  لمجموعة الرموز التي تؤدي قراءتها للانتقال من الحالة الابتدائية آلة حالة نهائية.
- اوجد التعبيرات المنتظمة لكافة مسارات الخروج من كل حالة نهائية والرجوع إليها  $v_1, \dots, v_m$
- ستكون عند هذا الرموز المقبولة هي:

$$(u_1 \cup \dots \cup u_n)(v_1 \cup \dots \cup v_m)^*$$

مثال:

اوجد اللغة المقبولة من قبل آلة الحالة المنتهية المثلة بالمخطط التالي:



الحل:

مجموعة الرموز التي تقود الى حالة نهائية من الحالة الابتدائية هي aba

وعليه فان اللغة المقبولة من قبل هذه الآلة هي:

$$L(M) = aba$$

مثال:

ابن مخطط الحالات لثلاثة المعبر عنها بما يلي ثم اوجد اللغة المقبولة من هذه الآلة:

Let  $M = \{Q, \Sigma, \delta, q_0, F\}$  be a DFA.

$$Q = \{q_0, q_1, q_2\}$$

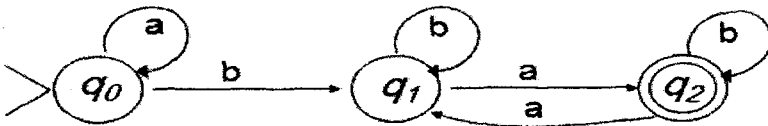
$$\Sigma = \{a, b\}$$

$$F = \{q_2\}$$

$\delta$	a	b
$q_0$	$q_0$	$q_1$
$q_1$	$q_2$	$q_1$
$q_2$	$q_1$	$q_2$

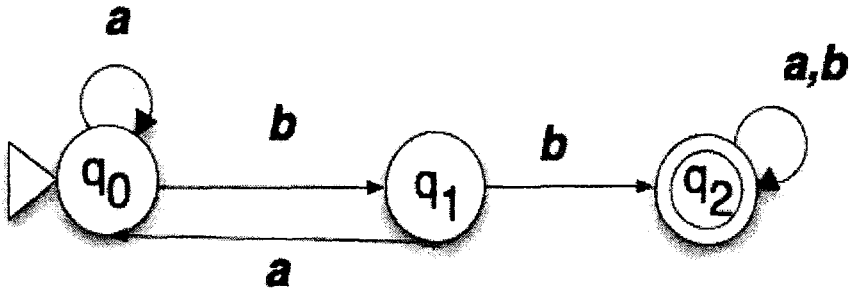
الحل:

$$RE = a^*b^*a(b \cup ab^*a)^*$$



مثال:

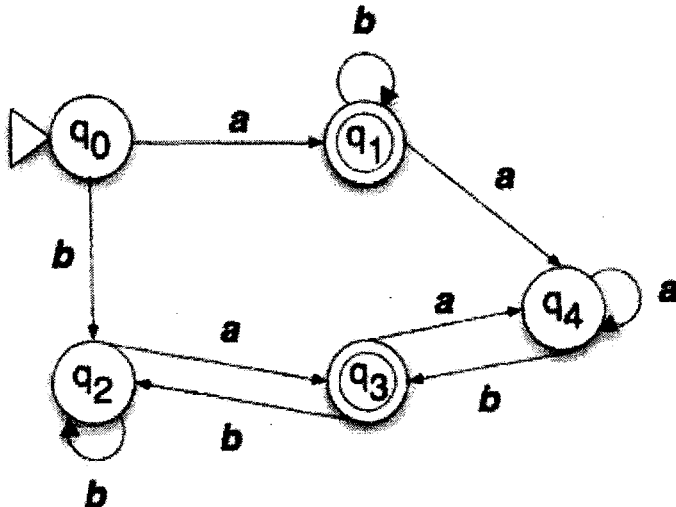
اوجد اللغة المقبولة لثلاثة التالية:



$$a^*b(aa^*b)^*b(a \cup b)^* = a^*b(a^*b)^*b(a \cup b)^*$$

مثال:

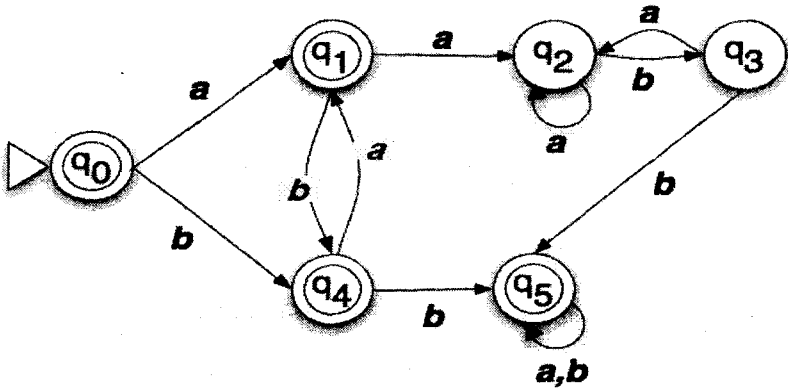
اوجد اللغة المقبولة لثلاثة التالية:



$ab^*$  and  $(ab^*aa^*b \cup bb^*a)(aa^*b \cup bb^*a)^*$ , so  $L(M)$  is  $ab^* \cup (ab^*a^+b \cup b^+a)(a^+b \cup b^+a)^*$

مثال:

اوجد اللغة المقبولة لثلاثة التالية:

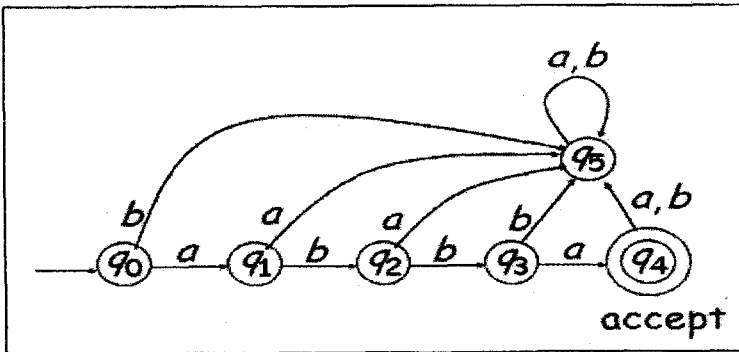


$\lambda \cup a(ba)^* \cup b(ab)^* \cup (a(ba)^*((a^+b)^+b \cup bb) \cup b(ab)^*(a(a^+b)^+b \cup bb))(a \cup b)^*$

مثال:

$L(M) = \{abba\}$

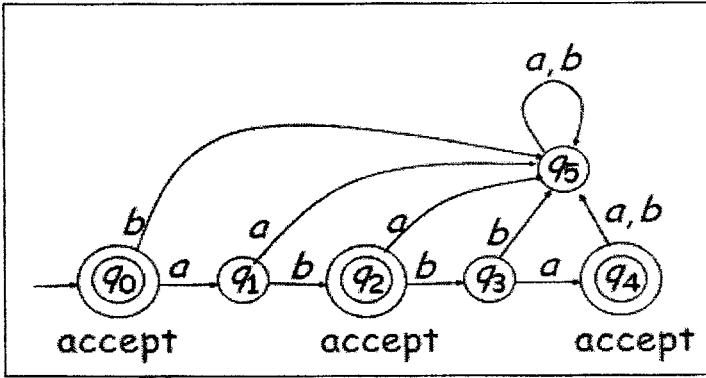
M



مثال:

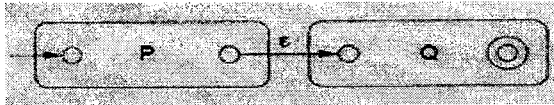
$$L(M) = \{\lambda, ab, abba\}$$

M



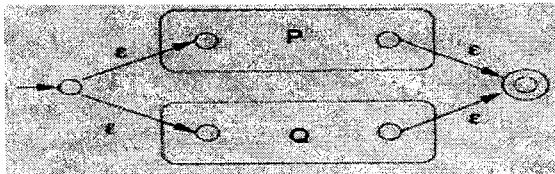
عند استخراج اللغة المقبولة للآلة المنتهية لا بد من الانتباه الى الحالات التالية:

- اذا قبلت آلة منتهية لغة محددة وكانت متبوعة بالة منتهية اخرى فان اللغة الناتجة هي حاصل دمج اللغتين وكما هو مبين في الشكل التالي:



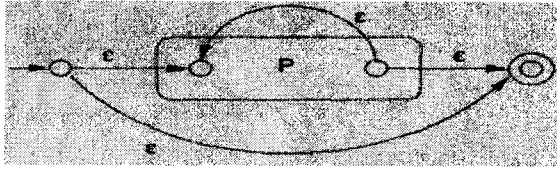
PQ

- اذا كان هناك تفرع لآلتين فان اللغة الناتجة هي لغة الآلة الاولى والثانية:



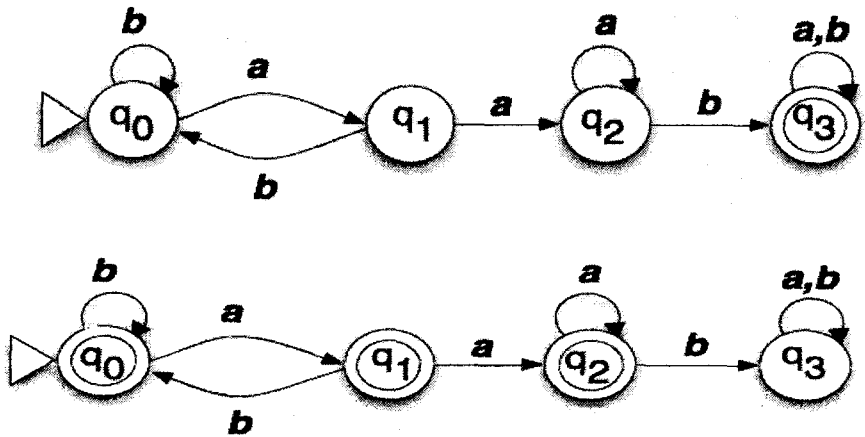
P+Q

- اذا كان هناك حلقة تشكل رجوع داخل الآلة فان اللغة المقبولة هي تكرر لمجموعة الرموز  $P^*$ .



### 5.2 اللغة المكملة Complement Language

الآلة المنتهية المكملة لآلة منتهية أخرى هي نفس هذه الآلة ولكن بعكس الحالات والمثال التالي يبين الآلة المنتهية تقبل الأحرف  $ab$  ومكملها والذي لا يقبل مجموعة هذه الأحرف:



يمثل مكمل اللغة لآلة الحالة المنتهية مجموعة الرموز المرفوضة من قبل الآلة:

Language accepted by  $M$  :

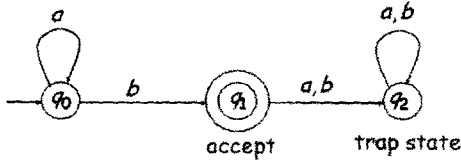
$$L(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \in F\}$$

Language rejected by  $M$  :

$$\bar{L}(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \notin F\}$$

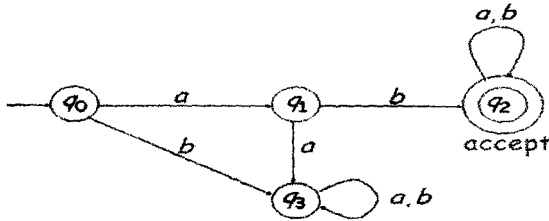
مثال:

$$L(M) = \{a^n b : n \geq 0\}$$



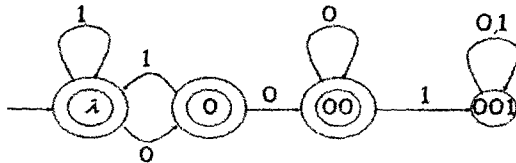
مثال:

$$L(M) = \{ \text{all substrings with prefix } ab \}$$



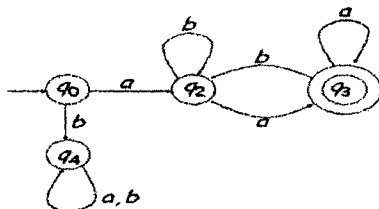
مثال:

$$L(M) = \{ \text{all strings without substring } 001 \}$$



مثال:

$$L = \{awa : w \in \{a,b\}^*\}$$



الوحدة الثالثة

آلة الحالة المنتهية

غير المحدودة

*Non-Deterministic  
Finite State Machine (NDFSM)*

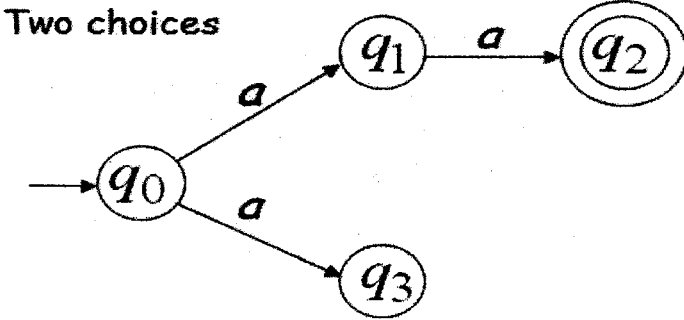
3



## 3. تعريف آلة الحالة المنتهية غير المحدودة:

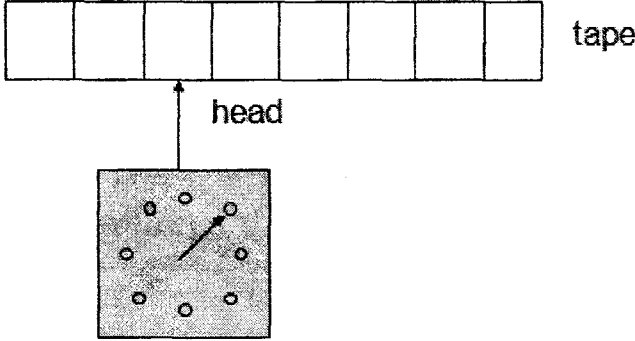
تشبه آلة الحالة المنتهية غير المحدودة الآلة المحدودة والتي تعرضنا إليها في الوحدة السابقة ولكن بخلاف بسيط الا وهو امكانية الانتقال الى اكثر من حالة عند قراءة الرمز المحدد والمثال التالي يبين نموذجا لآلة منتهية غير محدودة

$$\text{Alphabet} = \{a\}$$



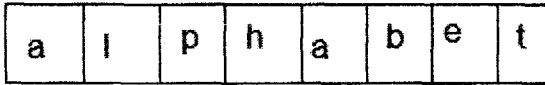
لاحظ امكانية الانتقال من الحالة الابتدائية الى حالتين باستخدام نفس الرمز والذي يدل على وجود خيارين.

تتكون الآلة المنتهية غير المحدودة من شريط من المدخلات ووحدة تحكم تحتفظ بالحالات وكما هو مبين في الشكل التالي:

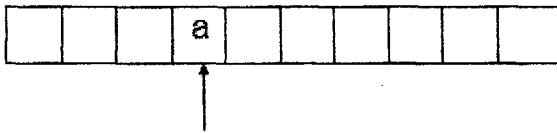


Finite Control

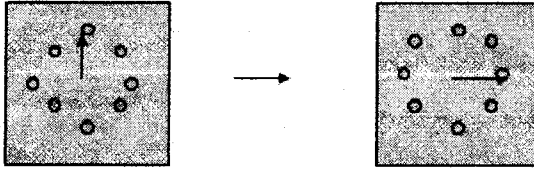
يقسم الشريط الى عدد محدود من الخلايا بحيث يتم تخصيص كل خلية لرمز من الرموز المشكلة لمجموعة الرموز المراد قراءتها من قبل الالة المنتهية غير المحدودة:



يستخدم راس القراءة لقراءة الرمز من الشريط ونقل الراس الى اليمين ولخطوة واحدة بعد قراءة الرمز او يمكن ابقاء الراس في مكانه في حالة قراءة لامدا او ايبسلون.

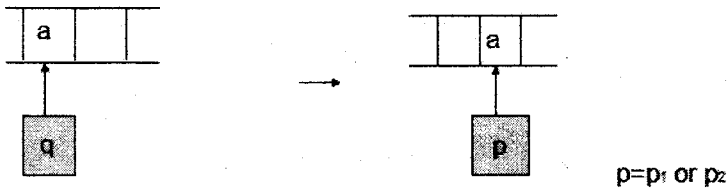
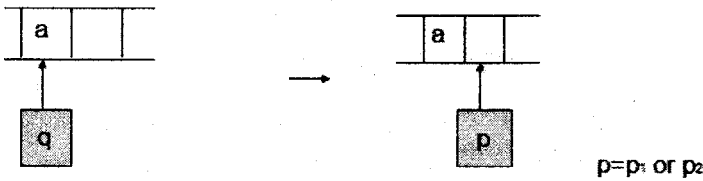


تنتقل الالة المنتهية غير المحدودة من حالة الى اخرى بعد قراءة الرمز ويمكن ان تبقى في نفس الحالة.



يمكن وصف الآلة المنتهية غير المحدودة باستخدام مخطط الحالات واستخدام جدول الانتقال ودالة الانتقال والتي تحدد الحالة المقبلة عند قراءة الرمز ووجود الآلة في حالة محددة مثل:

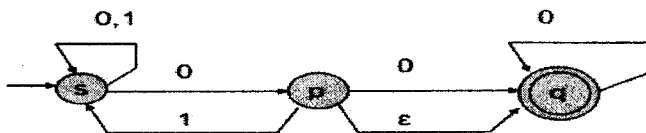
$$\delta(q, a) = \{p_1, p_2\}$$



والمثال التالي يبين جدول الانتقال ومخطط الحالات لآلة منتهية

غير محدودة:

$\delta$	0	1	$\epsilon$
s	p, s	s	q
p	q	s	q
q	q		



يتم وصف الآلة المنتهية غير المحدودة والتعبير عنها رياضيا بخمسة امور هي:

- مجموعة الحالات.
- مجموعة رموز المدخلات.
- دالة الانتقال.
- الحالة الابتدائية والتي تنتمي الى مجموعة الحالات.
- الحالة (اواكثر) النهائية والتي تنتمي هي الاخرى لمجموعة الحالات:

$$M = (Q, \Sigma, \delta, q_0, F)$$

$Q$  : set of states

$\Sigma$  : input alphabet

$\delta$  : transition function

$q_0$  : initial state

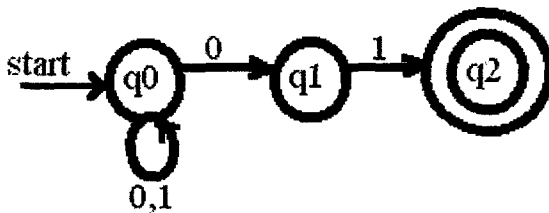
$F$  : set of accepting states

وفيما يلي نستعرض بعض الامثلة:

مثال

ابن الآلة الحالة المنتهية غير المحدودة والذي يقبل مجموعة من الازهار

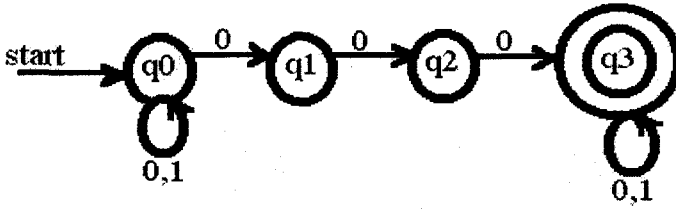
والوحدات والمنتهاية بصفر وواحد:



مثال:

ابن آلة الحالة المنتهية غير المحدودة لقبول سلسلة تحتوي على 3 اصفار

متتابة:



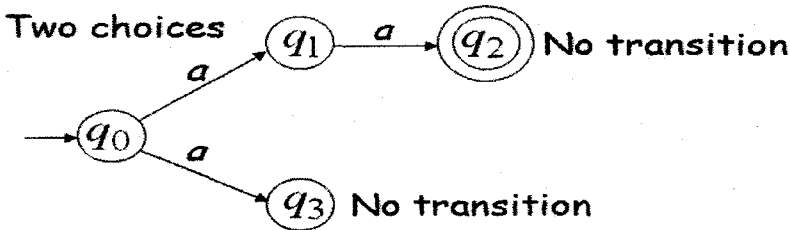
كما اشرنا سابقا فان سابقا فان الآلة المنتهية غير المحدودة يتوفر فيها

اكثر من خيار للتنقل عند قراءة الرمز بحيث يؤدي السير في الخيار الاول الى

النتهاء دون الانتقال الى الخيار الثاني وكذلك الحال فيما لو سلكت الآلة الخيار

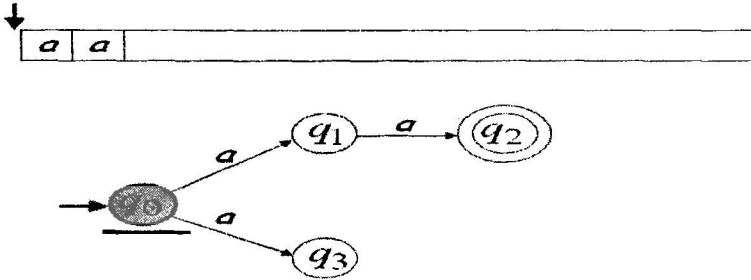
الثاني اولا وكما هو مبين في الشكل التالي:

Alphabet = {a}

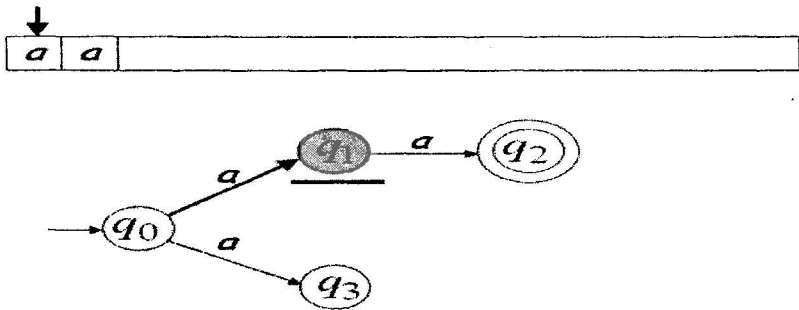


وفيما لو سلكت الآلة المسار الأول فان تتابع التنفيذ سيكون كما يلي:

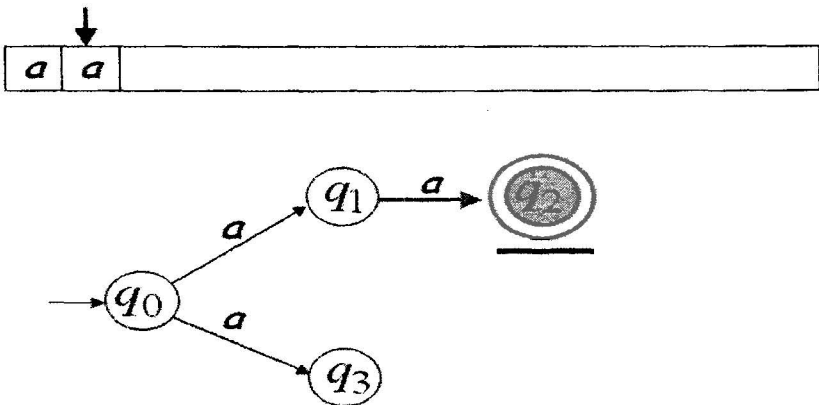
1.



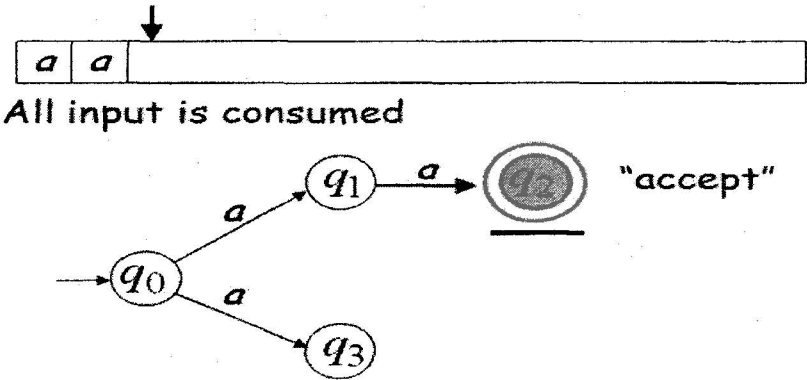
2.



3.

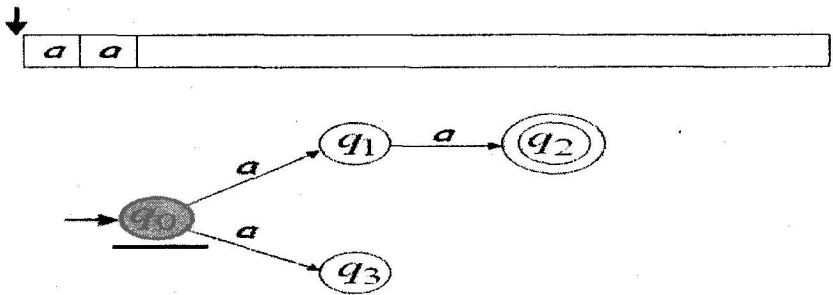


.4

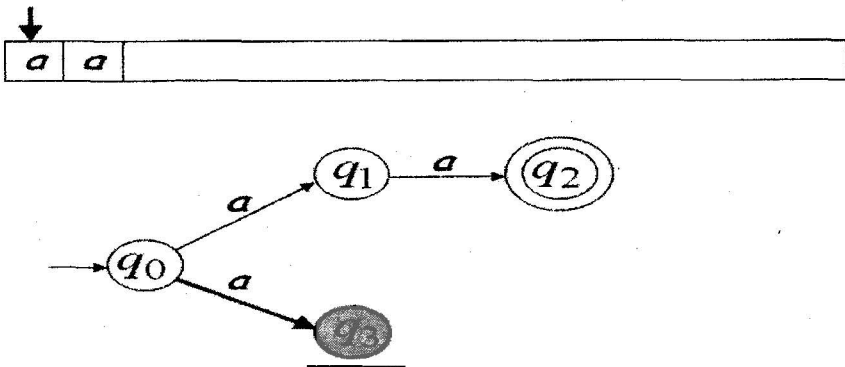


اما اذا سلكت الالة المسار الثاني فان الية التنفيذ ستكون كما يلي:

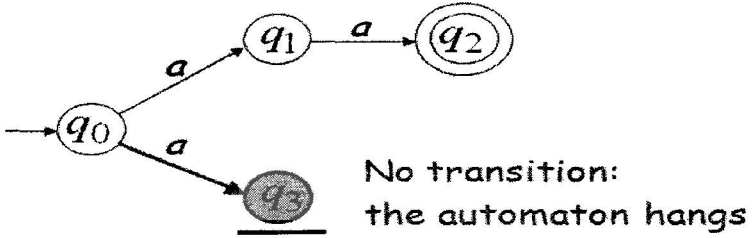
.1



.2



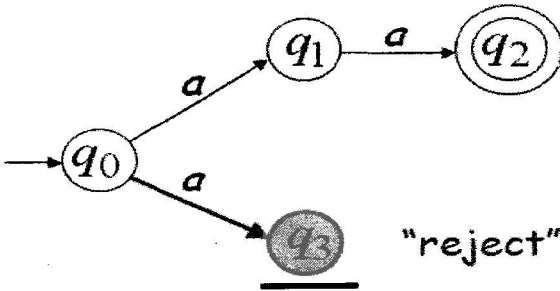
.3



.4



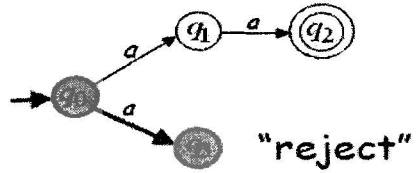
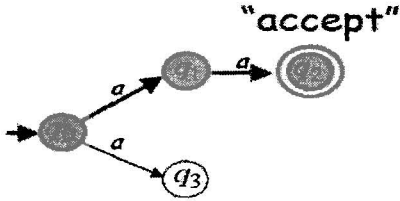
Input cannot be consumed





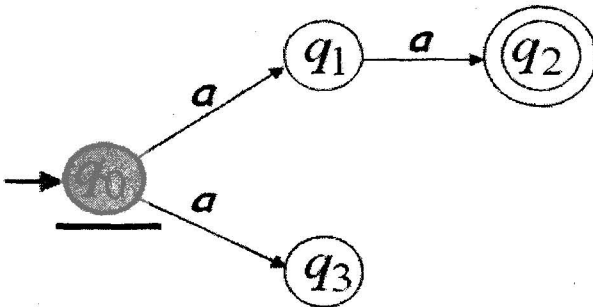
وعليه فان الالة السابقة تقبل اللغة aa وترفض a وكما هو مبين في الشكل التالي:

aa is accepted by the NFA:



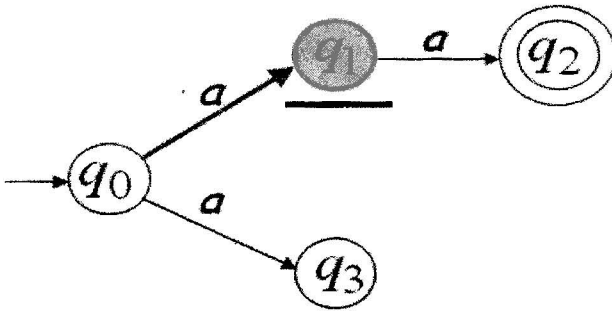
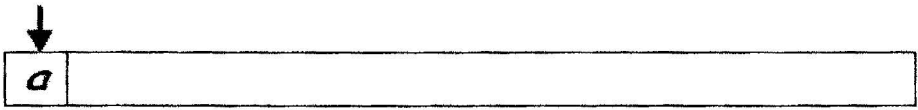
because this computation accepts aa

مما تقدم يتبين ان الالة ترفض سلسلة معينة من الرموز اذا كانت نتيجة الالة بعد قراءة هذه السلسلة غير محسوبة اي اذا انتهت عملية قراءة الرموز دون الوصول الى حالة نهائية فمثلا الالة التالية ترفض السلسلة a وفي المسارين:

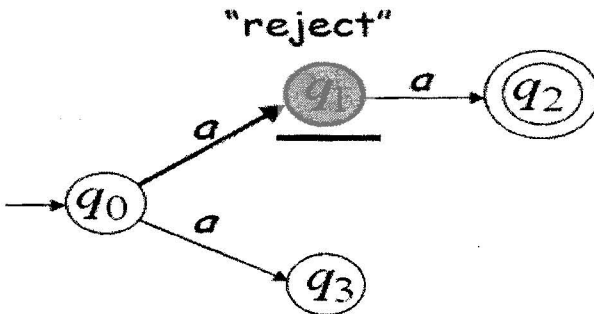
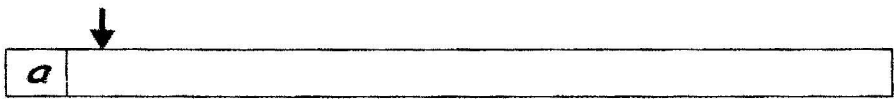


المسار الاول:

1.

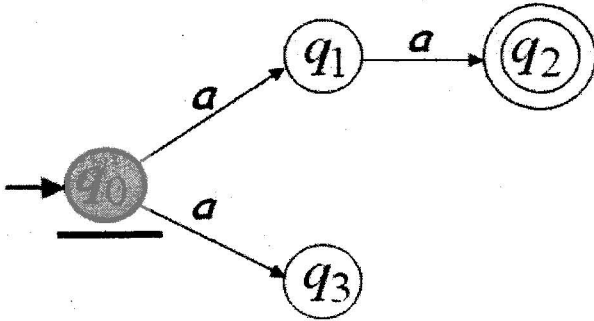
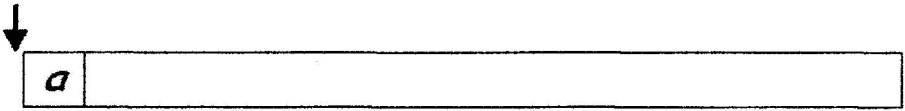


2

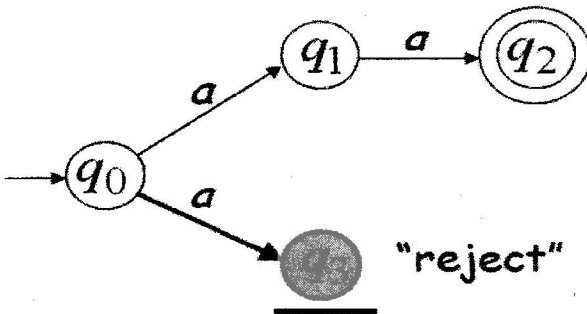
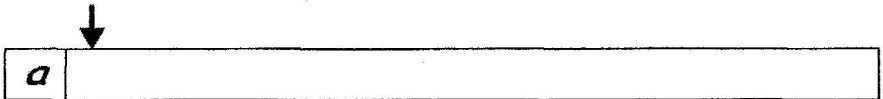


المسار الثاني:

.1

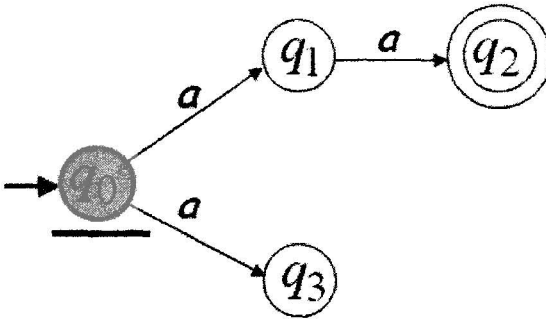


.2



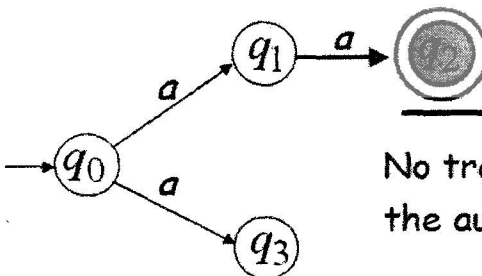
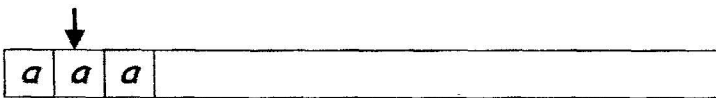
تعتبر آلة الحالة المنتهية غير المحدودة غير محسوبة في الحالات التالية:

1. إذا انتهت مجموعة الرموز دون الوصول إلى حالة نهائية.
2. إذا أدت عملية قراءة الرموز إلى الوصول إلى حالة غير نهائية.
3. إذا تم الوصول إلى حالة نهائية قبل انتهاء اللغة المطلوبة فإن السلسلة الرمزية المؤلفة للغة سيتم رفضها وما هو مبين في المثال التالي:



إذا سلكت الآلة المسار الأول فإن النتيجة النهائية ستكون رفض السلسلة

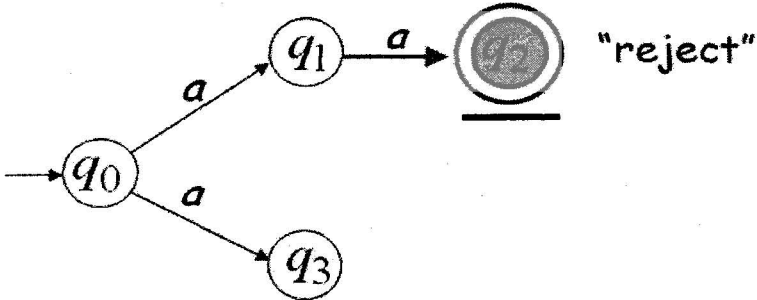
وكما هو مبين أدناه:



No transition:  
the automaton hangs

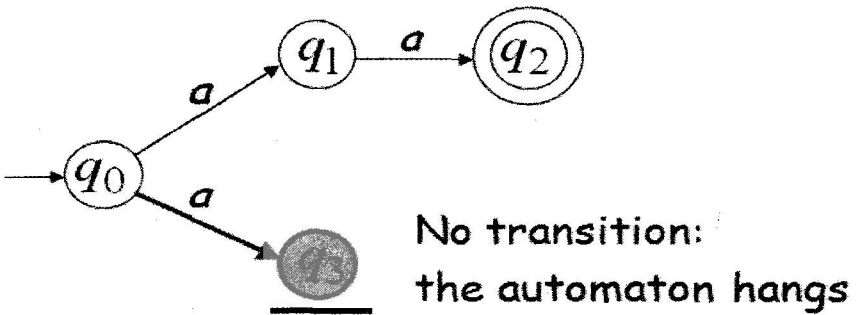
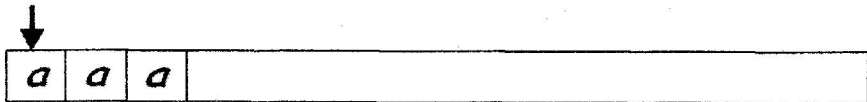


Input cannot be consumed

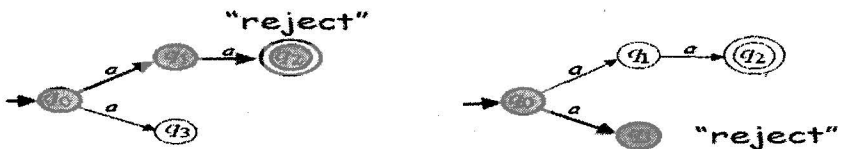


اما اذا سلكت الالة المسار الثاني فان النتيجة ستكون رفض السلسلة

للوصول الى حالة غير متفرعة وليست نهائية قبل اكمال السلسلة



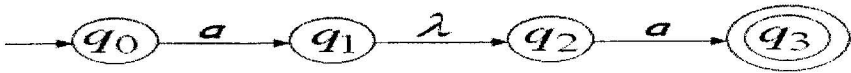
aaa is rejected by the NFA:



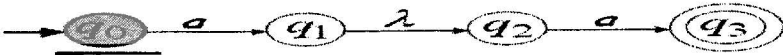
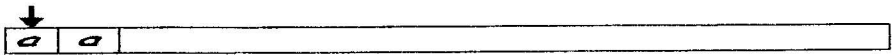
All possible computations lead to rejection

يمكن التخلص من بعض مشاكل رفض اللغة في الآلة المنتهية غير المحدودة وتحويلها من آلة غير محسوبة إلى آلة محسوبة وذلك باستخدام لامدا أو ايبسلون لتمثيل عملية الانتقال من حالة إلى أخرى حيث يمثل هذا الرمز فارغا وعند المرور به فإن رأس القراءة يبقى ثابتا على الرمز الذي كان واقفا عليه قبل الانتقال إلى الحالة الجديدة باستخدام لامدا.

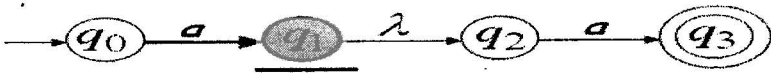
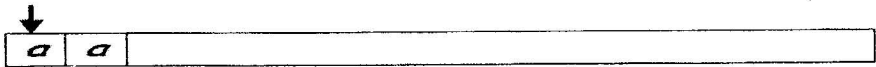
وفيما يلي مثلا يوضح كيفية استخدام لامدا لجعل الآلة المنتهية محسوبة:



1.

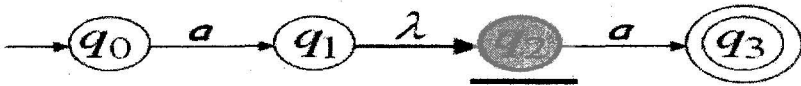
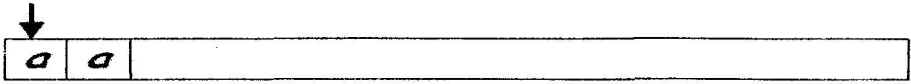


2.

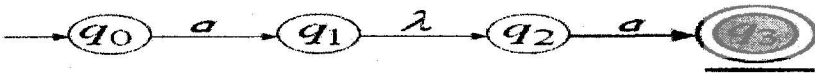
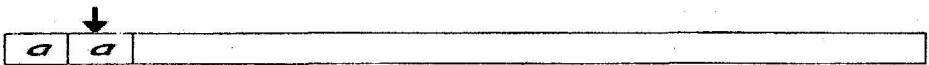


.3

(read head does not move)

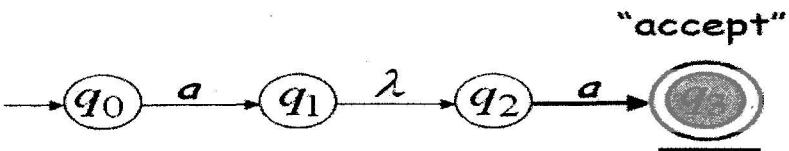


.4



.5

all input is consumed

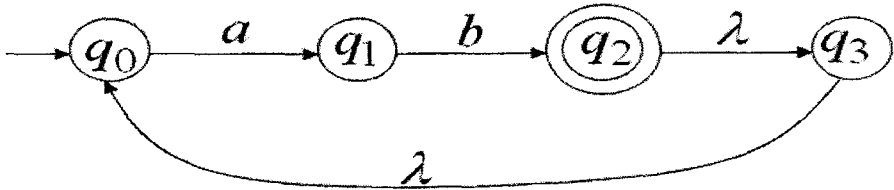


String  $aa$  is accepted

هذا ويمكن تكرار مجموعة الرموز المشكلة للغة المقبولة من قبل الآلة باستخدام لامدا وكما هو مبين في المثال التالي:

$$L = \{ab, abab, ababab, \dots\}$$

$$= \{ab\}^+$$

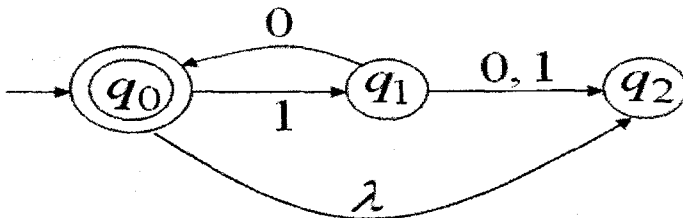


2.3 لغة الآلة المنتهية غير المحدودة:

كما اشرنا سابقا فان اللغة المقبولة من قبل الآلة المنتهية غير المحدودة هي مجموعة الرموز التي لو قررات فانها تقود الى حالة نهائية أو تصبح بعدها الآلة محسوبة والشكل التالي يبين مخطط لالة منتهية واللغة المقبولة من قبل هذه الآلة:

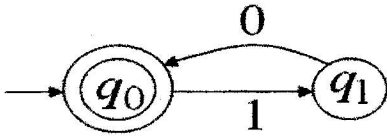
$$L(M) = \{\lambda, 10, 1010, 101010, \dots\}$$

$$= \{10\}^*$$





لاحظ في المثال السابق ان الحالة الثانية هي فائضة وعليه يمكن الاستغناء عنها لتحقيق أو تنفيذ نفس اللغة.



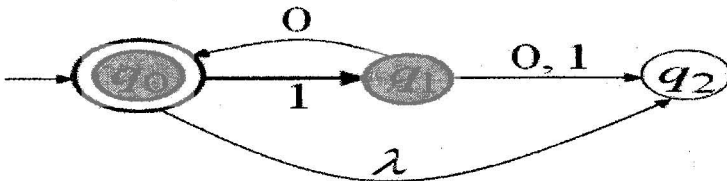
(q2 was a redundant state)

هذا ويمكن استخدام دالة النقل لتحديد لغة الآلة المنتهية غير المحدودة.

تعتبر دالة النقل عن الحالة التي يمكن الانتقال اليها بمعرفة الحالة الحالية والرمز المراد قراءته والامثلة التالية تبين بعض دوال الانتقال لآلة منتهية غير محدودة:

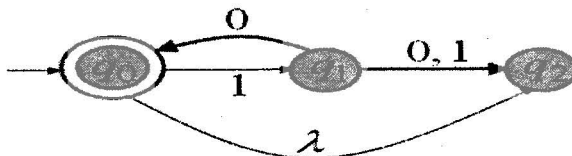
.1

$$\delta(q_0, 1) = \{q_1\}$$



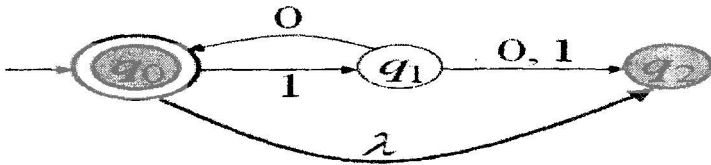
.2

$$\delta(q_1, 0) = \{q_0, q_2\}$$



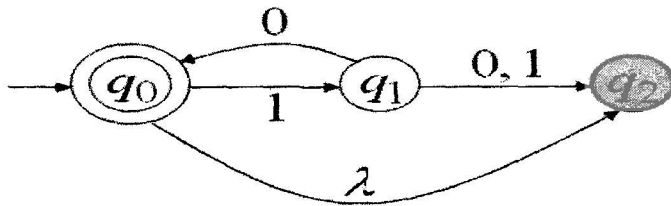
.3

$$\delta(q_0, \lambda) = \{q_0, q_2\}$$



.4

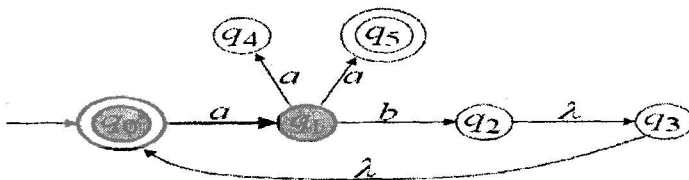
$$\delta(q_2, 1) = \emptyset$$



هذا ويستخدم مفهوم دالة الانتقال الموسعة والتي تشير الى الحالة أو الحالات التي يمكن الانتقال اليها من حالة حالية بعد قراءة سلسلة من الرموز والامثلة التالية تبين أمثلة على دالة الانتقال الموسعة:

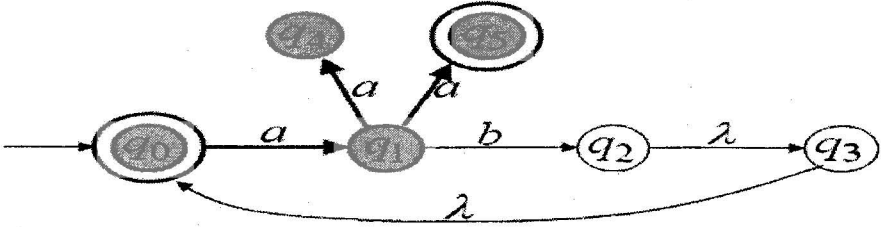
.1

$$\delta^*(q_0, a) = \{q_1\}$$



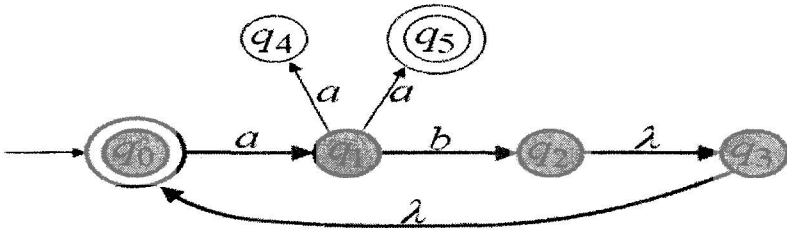
2.

$$\delta^*(q_0, aa) = \{q_4, q_5\}$$



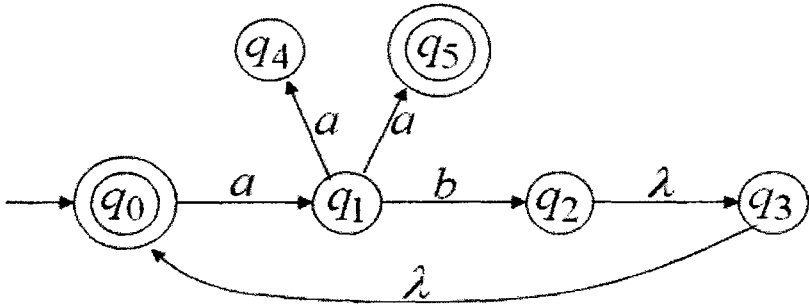
3.

$$\delta^*(q_0, ab) = \{q_2, q_3, q_0\}$$



ولتحديد اللغة المقبولة من الآلة المنتهية غير المحدودة يمكن استخدام دوال الانتقال الموسعة فإذا كان الطرف الأيمن أو أحد عناصره يشكل حالة نهائية في مجموعة الرموز الواقعة في الطرف الأيسر من الدالة تشكل جزءاً من اللغة المقبولة كونها تقود إلى حالة نهائية.

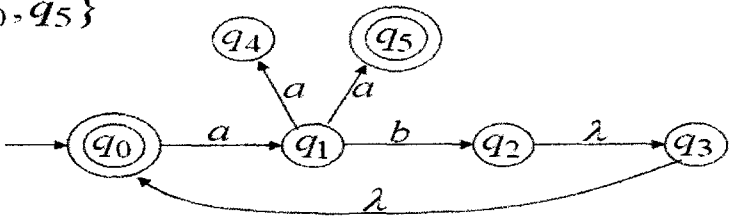
لنأخذ المثال التالي:



نحدد الحالات النهائية ودوال الانتقال الموسعة الى كل حالة من هذه الحالات:

1.

$$F = \{q_0, q_5\}$$

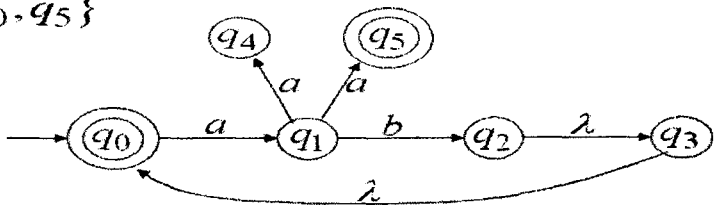


$$\delta^*(q_0, aa) = \{q_4, \underline{q_5}\} \quad aa \in L(M)$$

$\searrow \in F$

2.

$$F = \{q_0, q_5\}$$

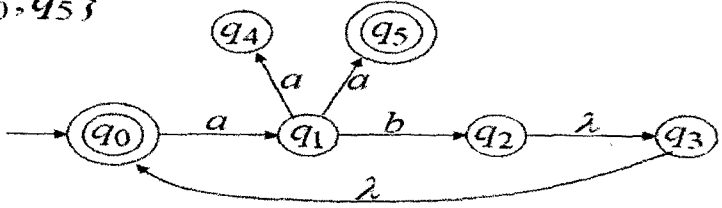


$$\delta^*(q_0, ab) = \{q_2, q_3, \underline{q_0}\} \quad ab \in L(M)$$

$\searrow \in F$

.3

$$F = \{q_0, q_5\}$$

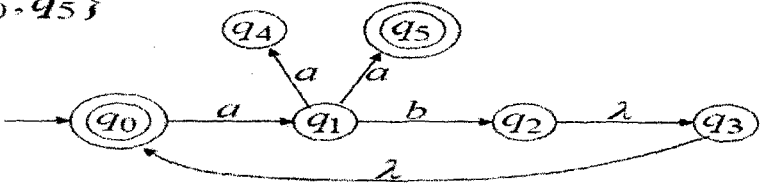


$$\delta^*(q_0, abaa) = \{q_4, q_5\} \quad aaba \in L(M)$$

$\searrow \in F$

.4

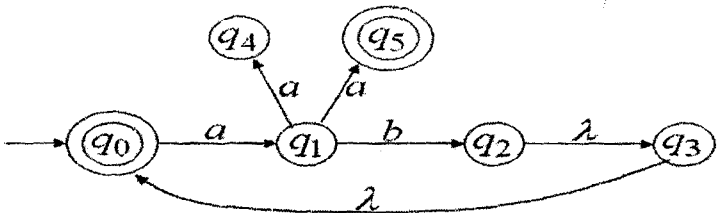
$$F = \{q_0, q_5\}$$



$$\delta^*(q_0, aba) = \{q_1\} \quad aba \notin L(M)$$

$\searrow \notin F$

.5



$$L(M) = \{\lambda\} \cup \{ab\}^* \{aa\}$$

### 3.3 تحويل آلة الحالة المنتهية غير المحدودة إلى آلة محدودة:

أشرنا سابقا إلى أن الآلة غير المحدودة تختلف عن الآلة المحدودة في احتواء الآلة غير المحدودة على أكثر من مسار عند تنفيذ لغة مؤلفة من مجموعة من الرموز المحددة وفيما يلي نبين أوجه الخلاف بين هاتين الآلتين وبشكل أكثر تفصيلا:

1. في الآلة المحدودة تكون كل عملية انتقال محددة أما في الآلة غير المنتهية فيمكن أن يكون لهيئة الآلة  $(q, a)$  أكثر من مسار للانتقال إلى حالة جديدة وباستخدام نفس الرمز (بدون انتقال، مسار أو مساران أو أكثر).
2. في الآلة المنتهية تقود عملية قراءة مجموعة الرموز المحددة إلى السلوك في مسار واحد محدد أما في الآلة غير المنتهية فيمكن أن تسلك الآلة أكثر من مسار بعد قراءة مجموعة الرموز المحددة.
3. لا تحتوي الآلة المنتهية على مسارات بالرمز ايبسلون أو لامدا فيمكن أن تحتوي الآلة غير المنتهية على مسارات من هذا النوع.
4. تختلف دالة الانتقال في الآلة غير المحدودة عنها في الآلة المحدودة وذلك باستخدام الرمز ايبسلون وعليه يمكن بيان أوجه الخلاف هذه رياضيا كما يلي:

#### • DFA

$M = (K, \Sigma, \delta, s, F)$ ,  
 $K: \{q_0, q_1, \dots, q_{m-1}\}$ , finite set of states,  
 $\Sigma$ : alphabet,  
 $s$ : start state,  
 $F$ : set of final states,  
 $\delta: K \times \Sigma \rightarrow K$ , transition function

#### • NFA

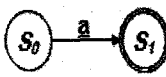
$M = (K, \Sigma, \Delta, s, F)$ ,  
 $K: \{q_0, q_1, \dots, q_{m-1}\}$ , finite set of states,  
 $\Sigma$ : alphabet,  
 $s$ : start state,  
 $F$ : set of final states,  
 $\Delta: K \times (\Sigma \cup \{e\}) \rightarrow K$ , transition relation

5. يتم قبول مجموعة الرموز في الآلة غير المنتهية اذا توفر على الاقل مسار واحد مشكل نتيجة لقراءة هذه الرموز وانتهى المسار بحالة نهائية.
6. اذا رفضت مجموعة الرموز في الآلة المنتهية فان هذا يعني عدم توفر مسار ينتهي بحالة نهائية ومشكل نتيجة لقراءة هذه الرموز.
7. بالرغم من تعدد المسارات في الآلة المنتهية غير المحدودة الا انه يمكن البدء بتصميمها اولا وبعد ذلك يمكن ان تحول الى آلة منتهية محدودة.

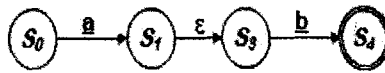
بناء على ما تقدم يمكن القول ان:

1. اية آلة منتهية غير محدودة يمكن ان تحول الى آلة مكافئة محدودة.
2. اللغة المقبولة من الآلة المكافئة يجب ان تكون مطابقة للغة المقبولة من الآلة غير المحدودة.
3. اذا كان عدد الحالات في الآلة غير المحدودة مساويا ل س فان عدد الحالات في الآلة المكافئة سيكون قريبا من 2 مرفوعة لاس س.

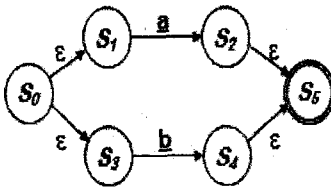
وقبل البدء بعملية التحويل من الآلة غير المنتهية الى الآلة المنتهية لنستعرض بعض الامثلة على التعابير المنتظمة وكيفية بناء الآلة غير المنتهية لقبولها:



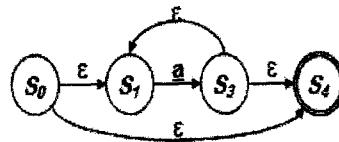
NFA for  $a$



NFA for  $ab$

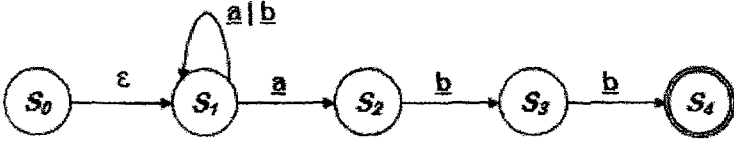


NFA for  $a|b$



NFA for  $a^+$

$$(\underline{a} \mid \underline{b})^* \underline{a} \underline{b} \underline{b}$$



كما اشرنا سابقا فان استخدام ايبسلون يؤدي الى تفادي بعض المشاكل في الآلة المنتهية غير المحدودة بتحويلها الى آلة محسوبة تقرا مجموعة من الرموز للوصول الى حالة نهائية لناخذ المثال التالي والمتضمن ببناء آلة منتهية غير محدودة لقبول التعبير المنتظم التالي:  $\underline{a} (\underline{b} \mid \underline{c})^*$

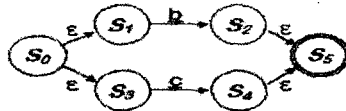
1. نبدء اولاً بكل رمز من الرموز الداخلة في التعبير.
2. الرمز الثاني مربوط مع الرمز الثالث بعلاقة أو اي التفرع باستخدام الرمز الثاني أو استخدام الرمز الثالث.
3. تكرار الخطوة 2 باستخدام عملية الانتقال بايبسلون.
4. دمج عملية التعرف على الرمز الاول بنتائج الخطوة الثالثة.

وفيما يلي توضيح لهذه النقاط باستخدام المخطط:

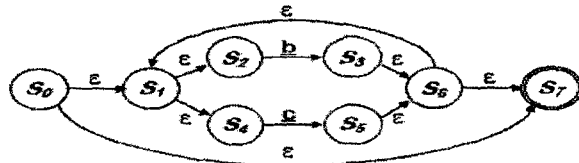
1.  $\underline{a}$ ,  $\underline{b}$ , &  $\underline{c}$



2.  $\underline{b} \mid \underline{c}$

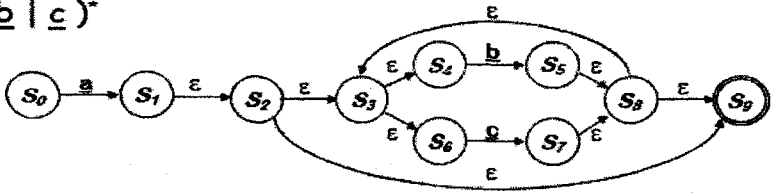


3.  $(\underline{b} \mid \underline{c})^*$

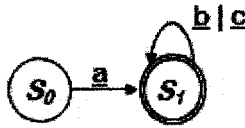




4.  $a(b|c)^*$



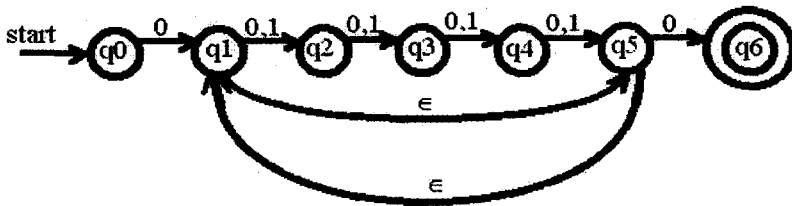
لاحظ انه بدون استخدام ايبسلون يمكن وصف المخطط كما يلي:



يمكن استخدام مسار ايبسلون لتنفيذ عملية التكرار لتنفيذ أو قراءة مجموعة من الرموز واللائحة التالية توضح كيفية استخدام مسار ايبسلون لتنفيذ عملية التكرار:

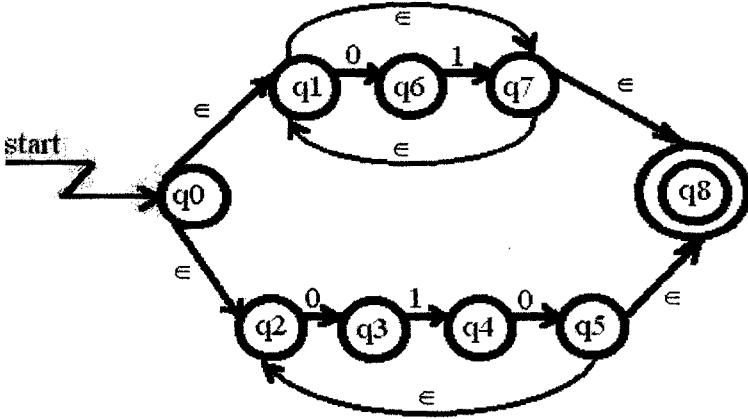
مثال:

ابن الآلة المنتهية غير المحدودة والتي تقبل اي سلسلة (مكونة من 0 و 1) تحتوي على صفرين مفصولين بسلسلة طولها 4 س حيث س اكبر أو تساوي الواحد:



مثال:

اين آلة منتهية غير محدودة تقبل سلسلة تحتوي على 01 مكررة صفر مرة او اكثر او تحتوي على 010 مكررة مرة واحدة او اكثر:



هناك طرق متعددة للتحويل من آلة منتهية غير محدودة الى آلة منتهية محدودة ومن هذا الطرق استخدام مفهوم التغطية باستخدام ايبسلون والتي تعني ايجاد جميع الحالات التي يمكن الوصول اليها باستخدام ايبسلون.

تستخدم هذه الطريقة اقترانين مفتاحين هما:

1. مجموعة الحالات التي يمكن الوصول اليها من حالة محددة بعد قراءة رمز او اكثر ويرمز لهذه الدالة ب:

Move ( $s_i, a$ )

2. مجموعة حالات التغطية بايبسلون وهي مجموعة الحالات التي يمكن الوصول اليها من حالة محددة باستخدام الرمز ايبسلون ويرمز لها ب:

$\epsilon$ -closure ( $s_i$ )

تنفذ خوارزمية التحويل حسب الخطوات التالية:

- Start state derived from  $s_0$  of the NFA
- Take its  $\epsilon$ -closure  $S_0 = \epsilon\text{-closure}(s_0)$
- Take the image of  $S_0$ , Move  $(S_0, \alpha)$  for each  $\alpha \in \Sigma$ , and take its  $\epsilon$ -closure
- Iterate until no more states are added

**The algorithm:**

```

 $s_0 \leftarrow \epsilon\text{-closure}(q_{0n})$ 
while (  $S$  is still changing )
  for each  $s_i \in S$ 
    for each  $\alpha \in \Sigma$ 
       $s_j \leftarrow \epsilon\text{-closure}(\text{Move}(s_i, \alpha))$ 
      if (  $s_j \notin S$  ) then
        add  $s_j$  to  $S$  as  $s_j$ 
         $T[s_i, \alpha] \leftarrow s_j$ 
    
```

**The algorithm halts:**

1.  $S$  contains no duplicates (test before adding)
2.  $2^{Q_n}$  is finite
3. while loop adds to  $S$ , but does not remove from  $S$  (monotone)

$\Rightarrow$  the loop halts

$S$  contains all the reachable NFA states

It tries each character in each  $s_i$ .

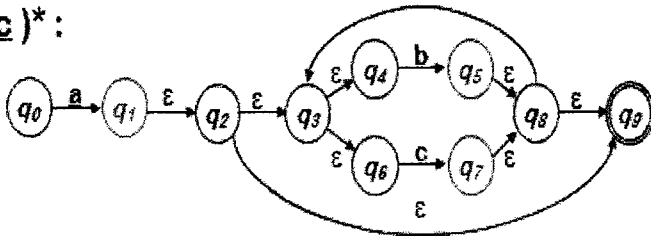
It builds every possible NFA configuration.

$\Rightarrow S$  and  $T$  form the DFA

وفيما يلي نستعرض مثالا نستخدم فيه هذه الخوارزمية لتحويل الآلة

المنتهية غير المحدودة الى آلة منتهية محدودة:

$a(b|c)^*$ :

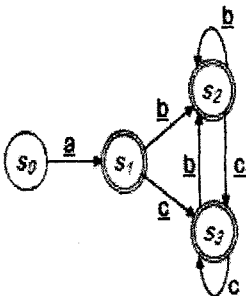


Applying the subset construction:

		ε-closure (move(s, *))		
		a	b	c
	NFA states			
$s_0$	$q_0$	$q_1, q_2, q_3, q_4, q_6, q_8$	none	none
$s_1$	$q_1, q_2, q_3, q_4, q_6, q_8$	none	$q_5, q_7, q_8, q_9$	$q_7, q_8, q_9$
$s_2$	$q_5, q_7, q_8, q_9$	none	$s_2$	$s_3$
$s_3$	$q_7, q_8, q_9$	none	$s_2$	$s_3$

Final states

The DFA for  $a(b|c)^*$



$\delta$	a	b	c
$s_0$	$s_1$	-	-
$s_1$	-	$s_2$	$s_3$
$s_2$	-	$s_2$	$s_3$
$s_3$	-	$s_2$	$s_3$

## 4.3 التخلص من ايبسلون في الآلة المنتهية غير المحدودة:

كما اشرنا سابقا الى فائدة استخدام ايبسلون في الآلة المنتهية غير المحدودة وذلك لتحويل الآلة الى آلة محسوبة تصل الى حالة نهائية بقراءة مجموعة من الرموز هذا ويمكن ايضا التخلص من ايبسلون مع المحافظة على اللغة وبقاء الآلة محسوبة ولتنفيذ عملية التخلص من ايبسلون يمكن تنفيذ الخوارزمية التالية والمثلة بمجموعة الخطوات التالية:

1. Compute  $\epsilon^*$  for the current state, resulting in a set of states  $S$ .
2.  $\delta(S,a)$  is computed for all  $a$  in  $\Sigma$  by
  - a. Let  $S = \{p_1, p_2, \dots, p_k\}$
  - b. Compute  $I=1 \rightarrow k$   $(p_i, a)$  and call this set  $\{r_1, r_2, r_3, \dots, r_m\}$ . This set is achieved by following input  $a$ , not by following any  $\epsilon$  transitions
  - c. Add the  $\epsilon$  transitions in by computing  $(S, a) = I=1 \rightarrow m$   $\epsilon^*(r_1)$
3. Make a state an accepting state if it includes any final states in the -NFA.

وفيما يلي نستعرض مثالا لاستخدام هذه الخوارزمية من اجل التخلص من

ايبسلون

لناخذ الآلة المنتهية غير المحدودة والمثلة كما يلي:

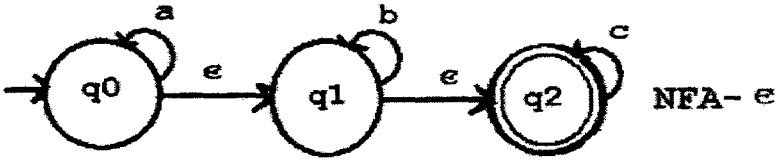
$$M = \{Q, \Sigma, \delta, q_0, F\}$$

$$Q = \{q_0, q_1, q_2\}$$

$$\Sigma = \{a, b, c\} \text{ and } \epsilon \text{ moves}$$

$$q_0 = q_0$$

$$F = \{q_2\}$$



دالة الانتقال لهذه الآلة هي كما يلي:

$\delta$	a	b	C	$\epsilon$
q0	{q0}	$\phi$	$\phi$	{q1}
q1	$\phi$	{q2}	$\phi$	{q2}
q2	$\phi$	$\phi$	{q2}	$\phi$

نجد الحالات الجديدة كما يلي:

$$Q' = \{\{q_0, q_1, q_2\}, \{q_1, q_2\}, \{q_2\}\} \text{ or renamed } \{q_x, q_y, q_z\}$$

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

$$F' = \{\{q_0, q_1, q_2\}, \{q_1, q_2\}, \{q_2\}\} \text{ or renamed } \{q_x, q_y, q_z\}$$

$$q_0 = \{q_0, q_1, q_2\} \text{ or renamed } q_x$$

نبني جدول الانتقال باستخدام هذه الحالات:

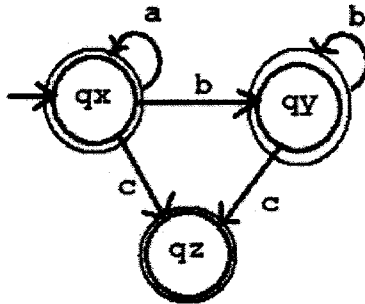
$\delta'$	a	b	c
$q_x \text{ or } \{q_0, q_1, q_2\}$			
$q_y \text{ or } \{q_1, q_2\}$			
$q_z \text{ or } \{q_2\}$			

$\delta'$	a	b	c
qx or {q0,q1,q2}	{qx} or {{q0,q1,q2}}	{qy} or {{q1,q2}}	{qz} or {{q2}}
qy or {q1,q2}	$\phi$	{qy} or {{q1,q2}}	{qz} or {{q2}}
qz or {q2}	$\phi$	$\phi$	{qz} or {{q2}}

$\delta'$	a	b	c
qx	{qx}	{qy}	{qz}
qy	$\phi$	{qy}	{qz}
qz	$\phi$	$\phi$	{qz}

$\rightarrow Q'$

وبهذا نحصل على الآلة المنتهية غير المنتهية وبدون استخدام ايبسلون:



## 5.3 تطبيقات الآلات الحالة المنتهية:

تستخدم الآلات المنتهية في كثير من التطبيقات ز من أهم هذه التطبيقات نورد:

1. تمثيل الدارات المنطقية التابعة المختلفة وسوف نستعرض نماذج من هذه الآلات في نهاية هذا الكتاب ان شاء الله ونخص بالذكر هنا آلة مور والة ميلي.
2. تمثيل وقراءة التعابير المنتظمة واللغات المؤلفة من مجموعة من الرموز.
3. تصميم المترجمات.
4. بناء البرمجيات الخاصة والمعتمدة على عملية تمييز الانماط مثل برامج معالجة الصور والصوت.

استعرضنا في الوحدة هذه والوحدة السابقة كيفية استخدام الآلة لتمثيل مجموعة من الرموز المشكلة لتعبير منتظم اولفة معينة بحيث تعمل الآلة المنتهية على قبول هذه المجموعة والتعرف عليها وفيما يلي وللتذكير فقط نورد بعض الامثلة على كيفية تمييز التعابير المنتظمة.

مثال: الآلة المنتهية والتي تقبل عددا زوجيا من الاحرف علما بان سلسلة

الاحرف مؤلفة من حرفين:

$$\delta(q_0, a) = q_1$$

$$\delta(q_0, b) = q_2$$

$$\delta(q_1, a) = q_0$$

$$\delta(q_1, b) = q_3$$

$$\delta(q_2, a) = q_3$$

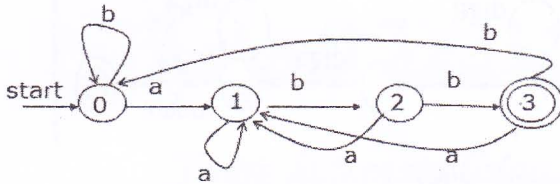
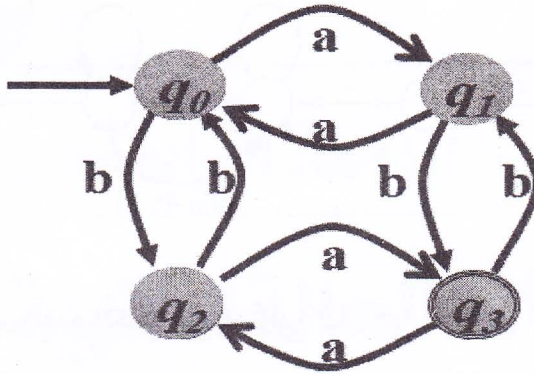
$$\delta(q_2, b) = q_0$$

$$\delta(q_3, a) = q_2$$

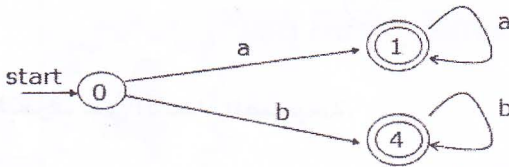
$$\delta(q_3, b) = q_1$$



$\delta$	a	b
$q_0$	$q_1$	$q_2$
$q_1$	$q_0$	$q_3$
$q_2$	$q_3$	$q_0$
$q_3$	$q_2$	$q_1$



Accepted language:  $(a|b)^*abb$

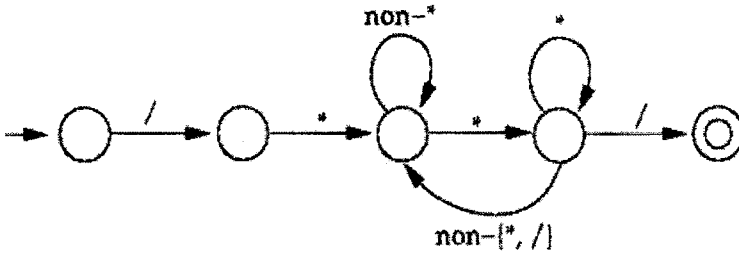


Accepted language:  $a^+ | b^+$

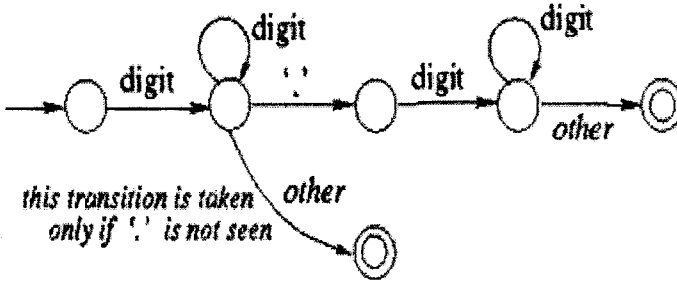
تستخدم الآلات المنتهية أيضا في عملية التحليل اللغوي وهي من من المراحل الأساسية لبناء المترجمات والتي تعمل على ترجمة البرامج المصدرية المكتوبة بلغة كلفة سي بلس باس ويشبه دور الآلات المنتهية هنا دور الآلات المنتهية في

التعرف على التعبيرات المنتظمة وقبولها وفيما نستعرض بعض مهمات المترجم وكيف يمكن تمثيلها باستخدام الآلات المنتهية:

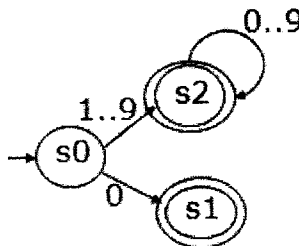
- تمييز الملاحظة في لغة سي بلس بلس والتعرف وقبول هذه الملاحظة اذا كانت بالصيغة المحددة أو رفضها اذا لم تتطابق مع لغة الآلة:



- التعرف على القيم الكسرية مثل 123.45



- التعرف على الاعداد الصحيحة.



الوحدة الرابعة

آلة الحالة المنتهية

المستخدمة للحزمة

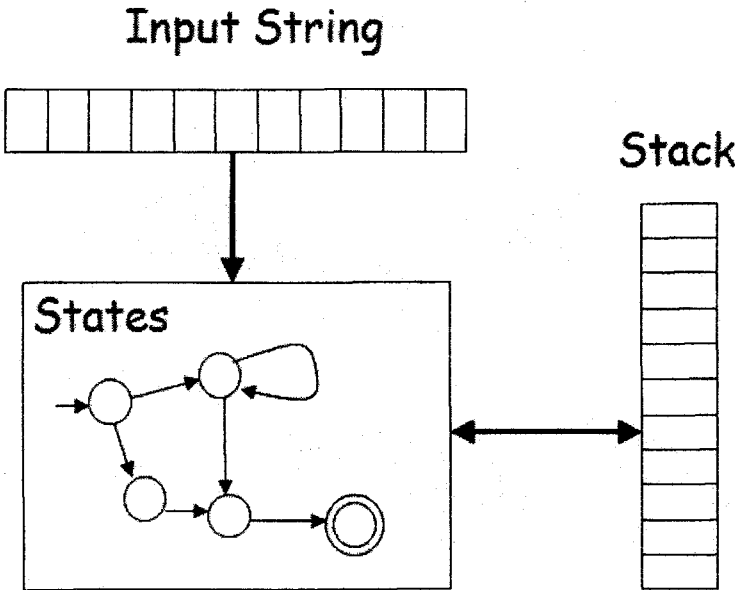
*Finite Automata Using Stack*

*(Push Down Automata)*

4

## 1.4 المفهوم العام لآلة الحالة المستخدمة للحزمة:

تتكون الآلة المنتهية ذات الدفع في الحزمة وكما هو مبين في الشكل التالي من شريط المدخلات والذي يحتوي على مجموعة الرموز المراد قراءتها ووحدة تحكم تحتفظ بحالات الآلة وحزمة تستخدم لعملية حفظ الرموز واسترجاعها:

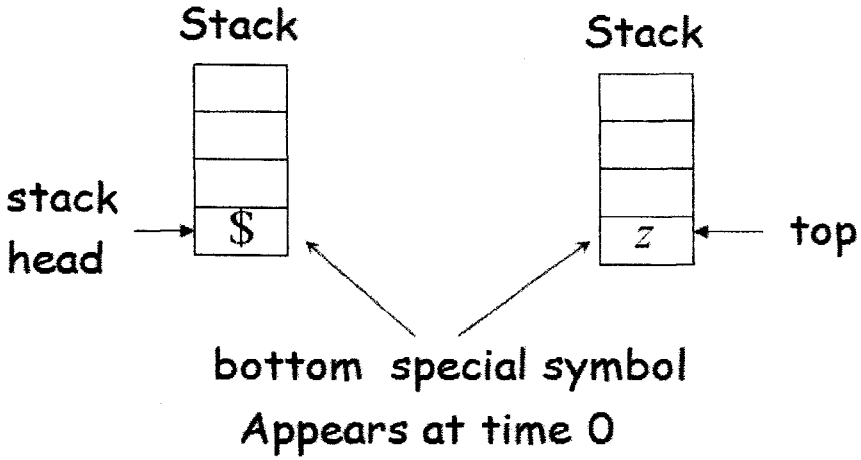


تشبه هذه الآلة إلى حد كبير الآلات المنتهية التي استعرضناها في الوحدة الثانية والثالثة والخلاف الوحيد هنا هو استخدام هذه الآلة لذاكرة الحزمة والتي تستخدم لحفظ الرموز المقروءة واسترجاعها.

تتألف الحزمة من مجموعة من المواقع وتنفذ عليها عمليتا الدفع (الحفظ) في الحزمة والاسترجاع.

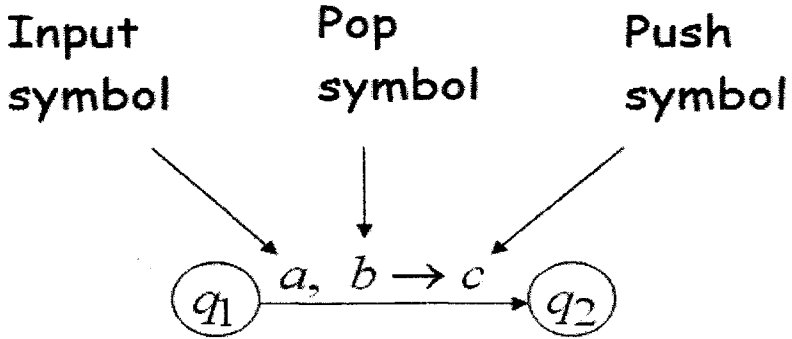
تمتاز الحزمة بخاصية الداخل أولاً خارج أولاً أي أن آخر رموز حفظ في الحزمة يتم استرجاعه وحذفه من الحزمة أولاً.

تبدء الحزمة بحالة ابتدائية يتم وضع رمز خاص فيها للإشارة الى انتهاء الرموز المقروءة وكما هو مبين في الشكل التالي فاذا نفذت الآلة المنتهية وقرات مجموعة من الرموز ونتج عن عمليات القراءة تنفيذ عمليات اضافة وحذف من الحزمة واصبحت الحزمة بعد تنفيذ هذه العمليات فارغة فان هذا سيدل الى الوصول الى حالة منتهية أو قبول مجموعة الرموز والتعرف عليها.



كما اشرنا تنفيذ عملية الاضافة والحذف على الحزمة من طرف واحد ويستخدم مؤشر يشير الى هذا الطرف وعند تنفيذ عملية الاضافة يزداد المؤشر بقيمة واحد ثم يوضع الرمز في الحزمة وعند الحذف يسترجع العنصر المشار اليه بالمؤشر من الحزمة ويطرح واحد من المؤشر.

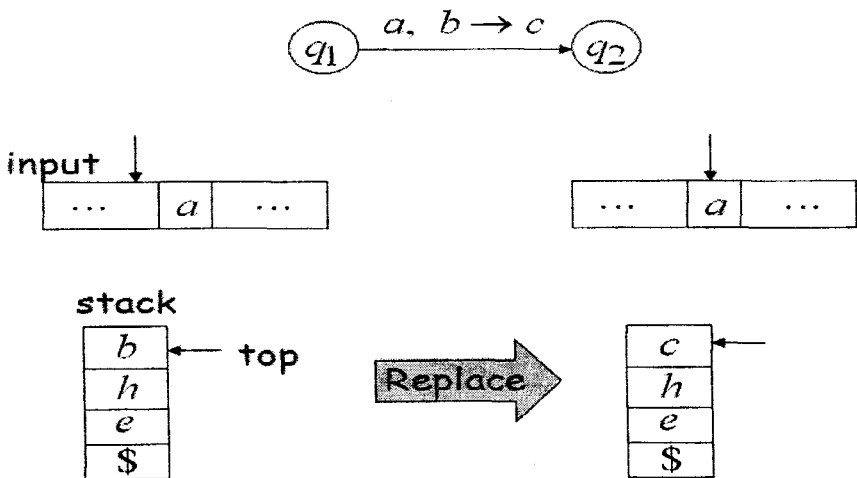
تستخدم الدائرة لتمثيل الحالة والخط المستقيم المنتهي بالسهم يخصص لتمثيل عملية الانتقال من حالة الى اخرى على ان تكتب على الخط نواتج عملية الانتقال وكما هو موضح في الشكل التالي:



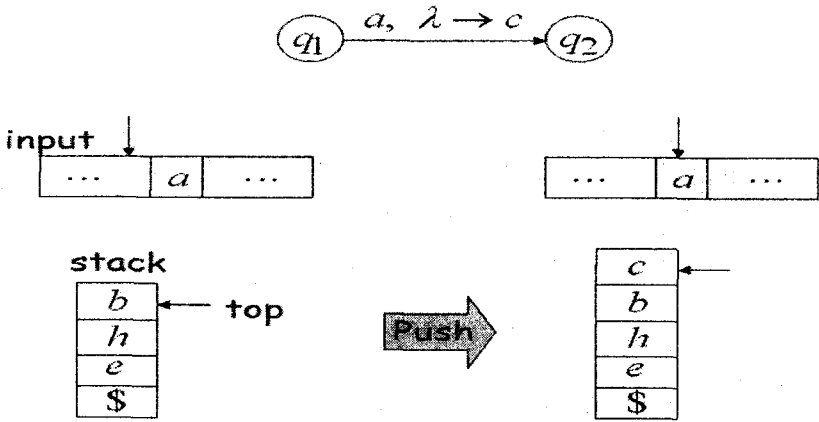
اي ان عملية الانتقال مرتبطة بالحالة الحالية والرمز المراد قراءته والرمز المراد استرجاعه من الحزمة والرمز المراد حفظه في الحزمة والحالة (او الحالات اذا كانت الالة غير محدودة) المراد الانتقال اليها (بعد القراءة ينتقل راس القراءة الى اليمين ولموقع واحد على شريط المدخلات).

والاشكال التالية توضح الية تنفيذ عملية الانتقال من حالة الى حالة اخرى وتنفيذ عمليتي الاضافة الى الحزمة والحذف منها.

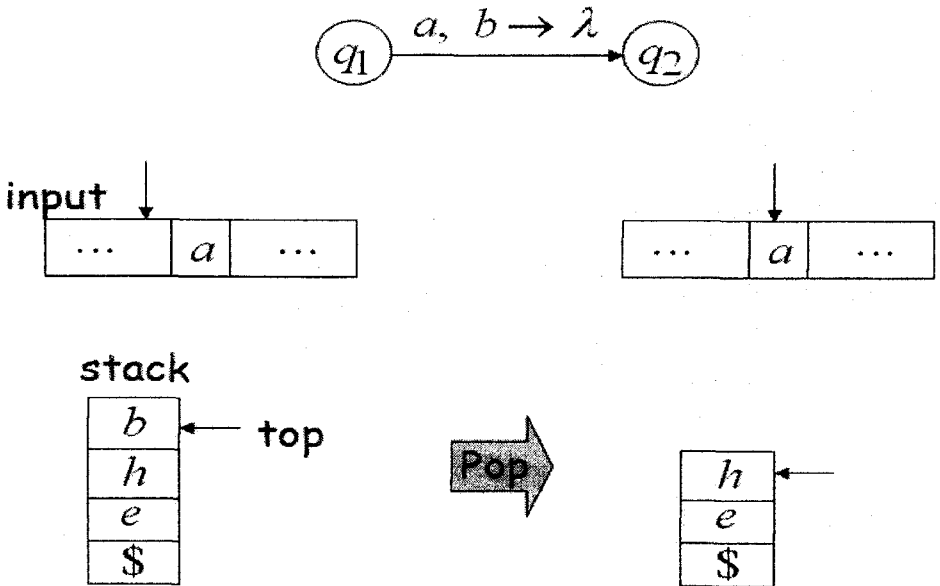
1.

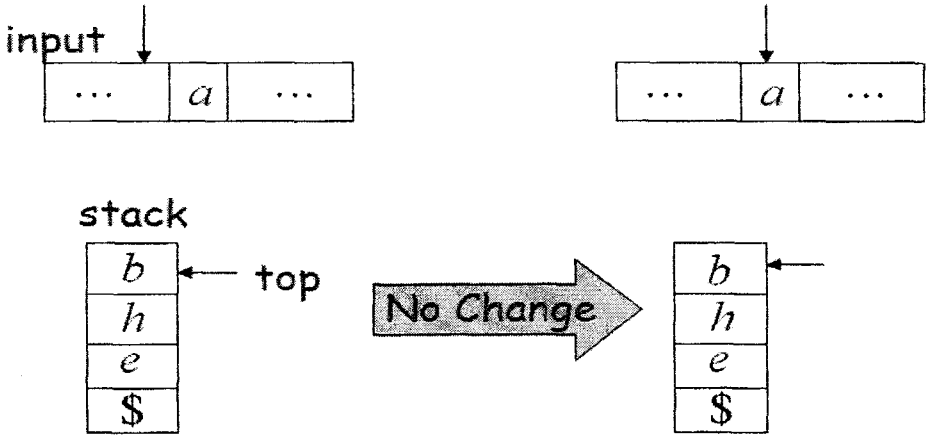
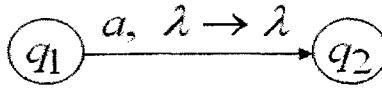


.2

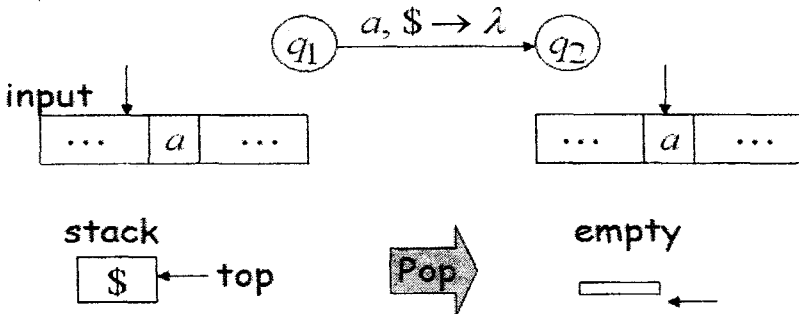


.3





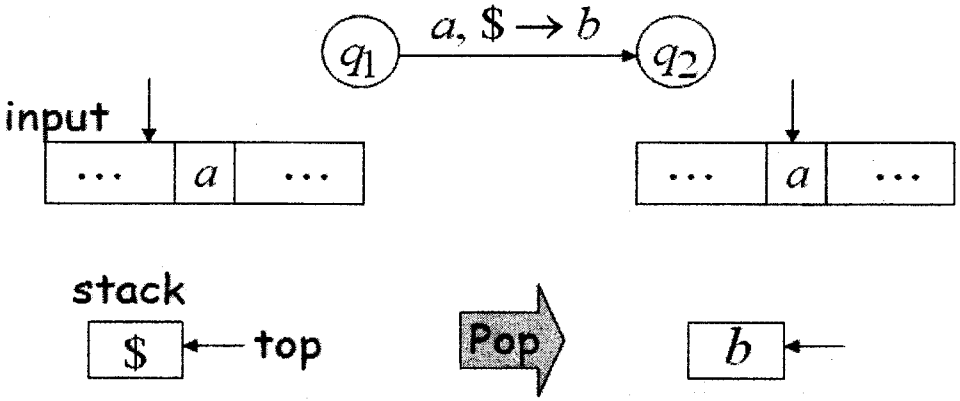
إذا أصبحت الحزمة فارغة فهذا مؤشر إلى انتهاء عملية الانتقال وانتهاء الرموز المقروءة فإذا كانت الحالة التي تم الوصول إليها نهائية فإن الآلة تكون قد ترفت على الرموز وقبلتها وإلا ستكون مجموعة الرموز المقروءة مرفوضة من قبل هذه الآلة والشكل التالي يبين الوصول إلى نهاية الرموز بالحصول على حزمة فارغة:



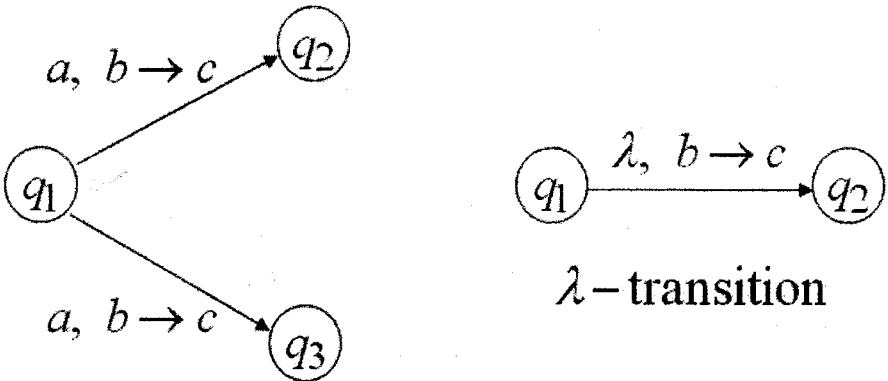
The automaton HALTS  
No possible transition after  $q_2$



والشكل التالي يبين عملية انتقال مسموحة.

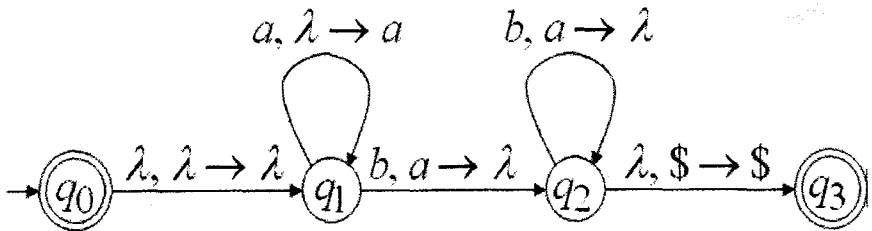


تشبه الآلة المنتهية المستخدمة للحزمة آلة الحالة المنتهية والتي يمكن أن تكون محدودة أو غير محدودة وعليه فإن الآلة المنتهية المستخدمة للحزمة يمكن أن تكون محدودة أو غير محدودة ولحل عملية الفرع في المسارات بقراءة نفس الرمز يمكن استخدام عملية الانتقال بلا مدام (الانتقال بالفراغ أو بلا شيء ودون تحريك رأس القراءة تماما كما في الآلة المنتهية غير المحدودة) والشكل التالي يبين هذا:

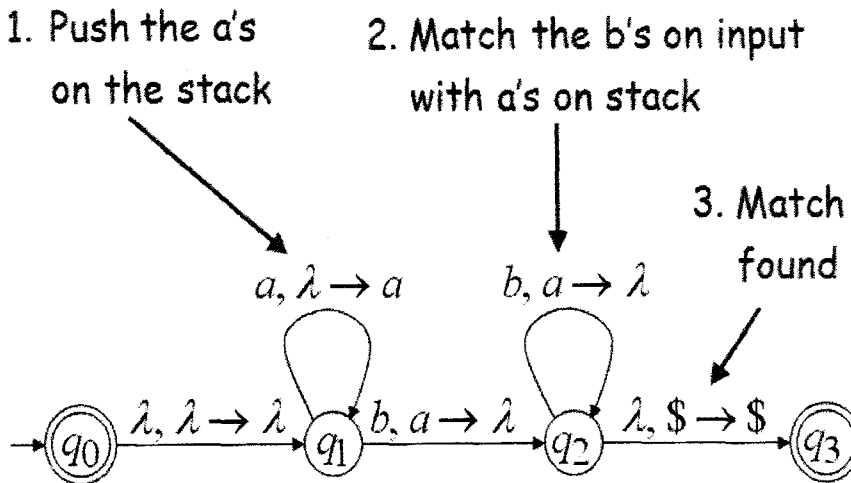


والآن لناخذ مثال ونبين كيفية تنفيذ هذه الآلة:

$$L(M) = \{a^n b^n : n \geq 0\}$$



والشكل التالي يوضح مدلولات عملية الانتقال من حالة لاخرى:

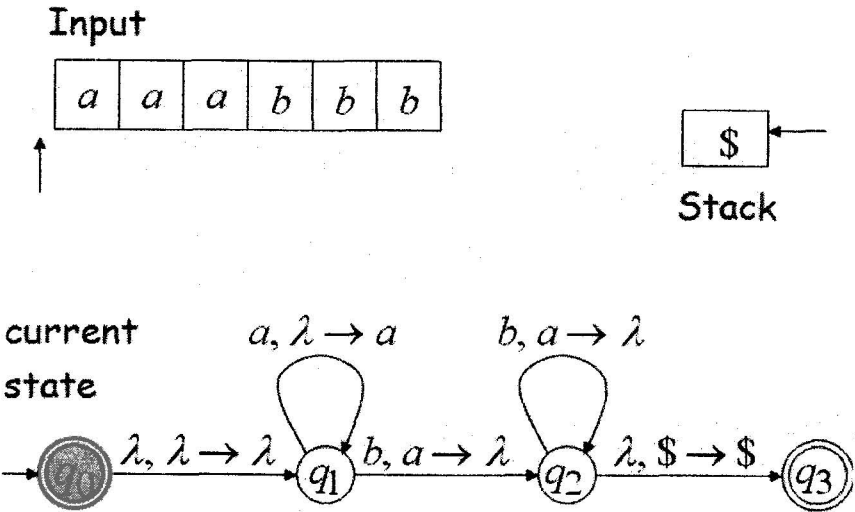


وفيما يلي خطوات تنفيذ هذه الآلة وبشكل مفصل أملا في إيضاح مفهوم

هذا النوع من الآلات:

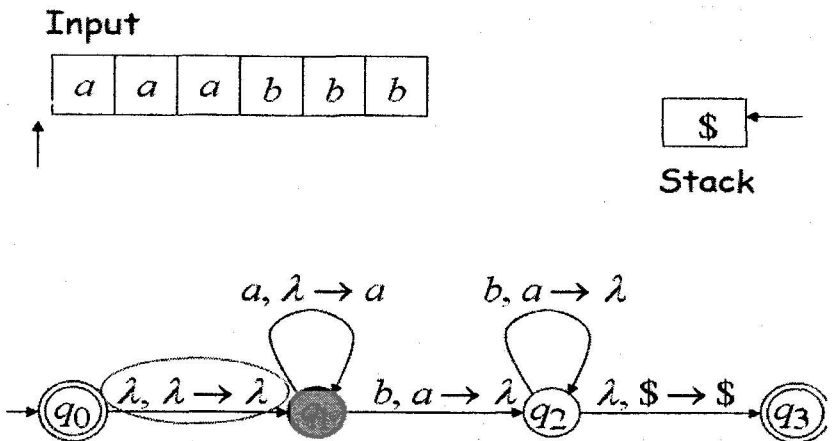
.1

Execution Example: Time 0



.2

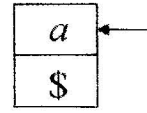
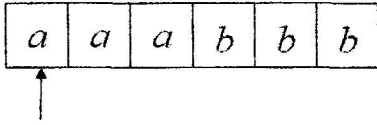
Time 1



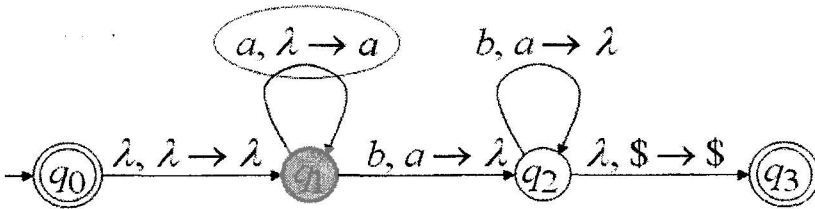
.3

Time 2

Input



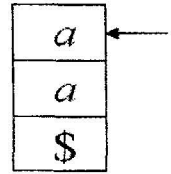
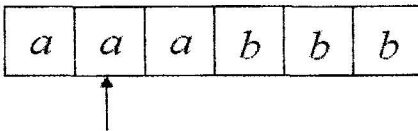
Stack



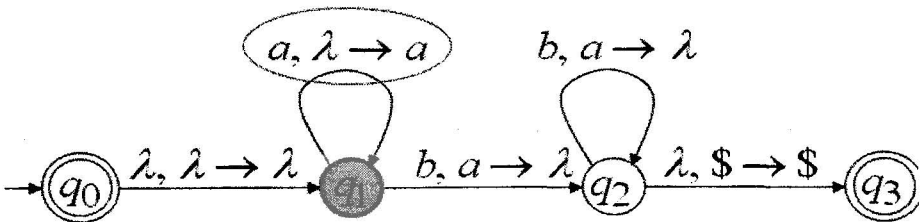
.4

Time 3

Input



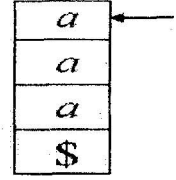
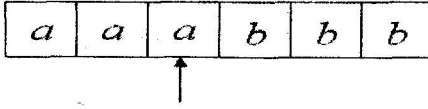
Stack



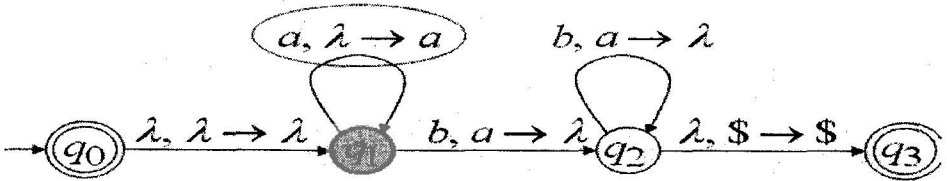
.5

Time 4

Input



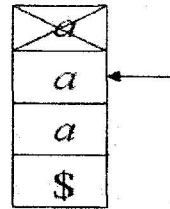
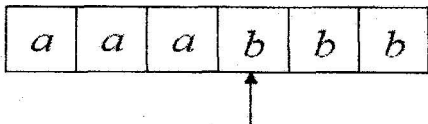
Stack



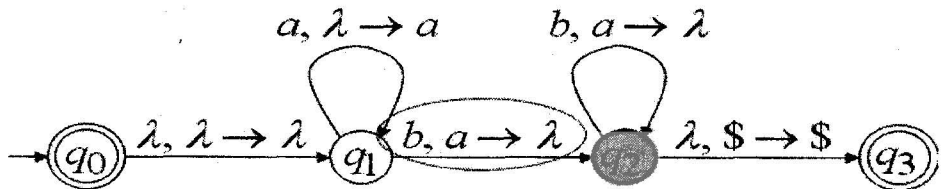
.6

Time 5

Input



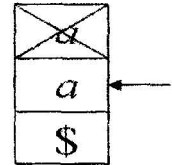
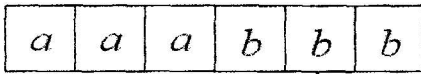
Stack



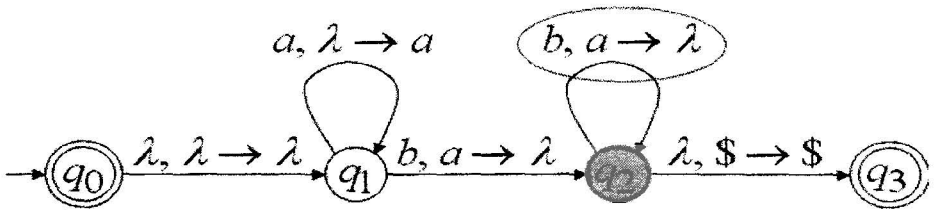
.7

Time 6

Input



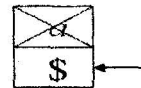
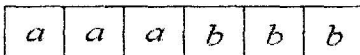
Stack



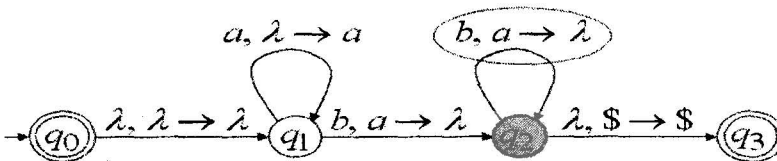
.8

Time 7

Input

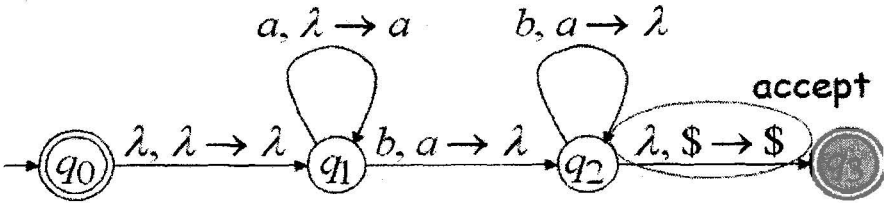
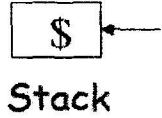
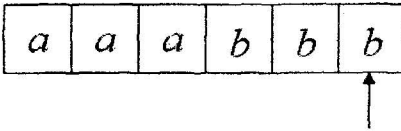


Stack



Time 8

Input



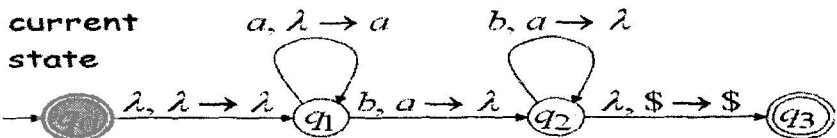
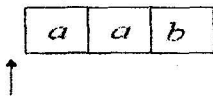
مثال:

عملية رفض الرموز

هل اللغة التالية مقبولة من قبل الآلة؟

Rejection Example: Time 0

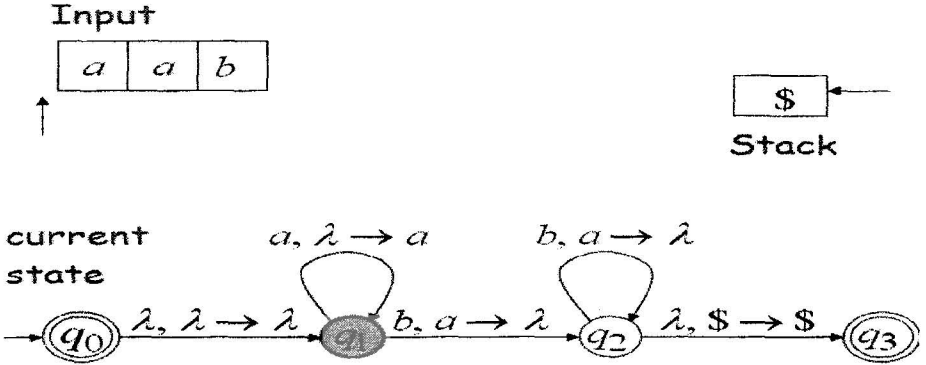
Input



لنتبع هذه الآلة خطوة خطوة:

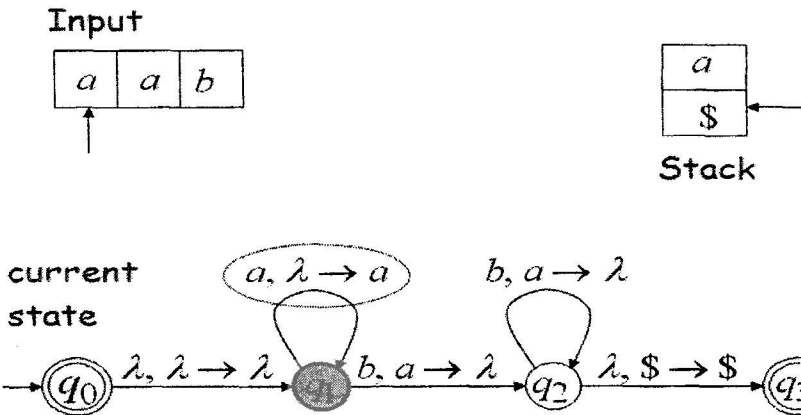
1.

Time 1



2.

Time 2

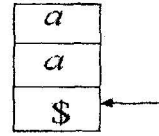
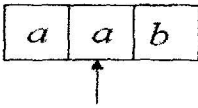




.3

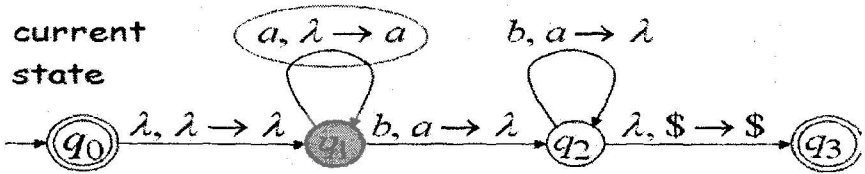
Time 3

Input



Stack

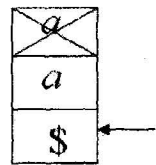
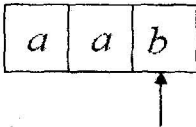
current state



.4

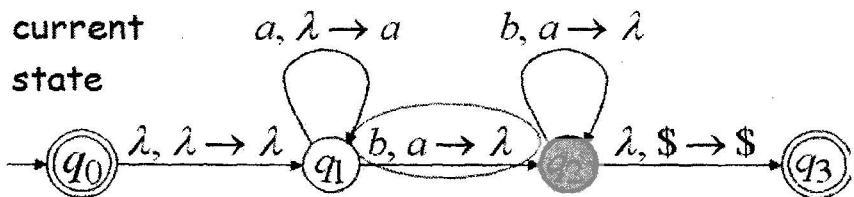
Time 4

Input



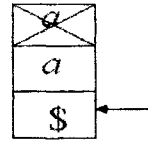
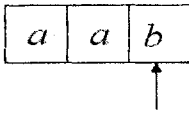
Stack

current state



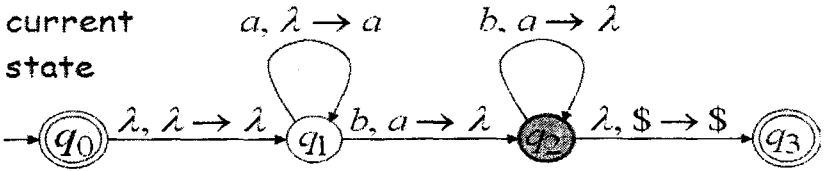
Time 4

Input



Stack

reject



مثال:

لتأخذ الآلة التالية:

$$L(M) = \{vv^R : v \in \{a,b\}^*\}$$

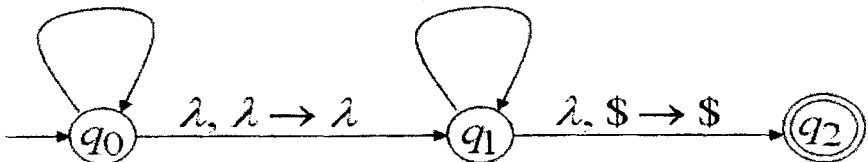
PDA  $M$

$a, \lambda \rightarrow a$

$a, a \rightarrow \lambda$

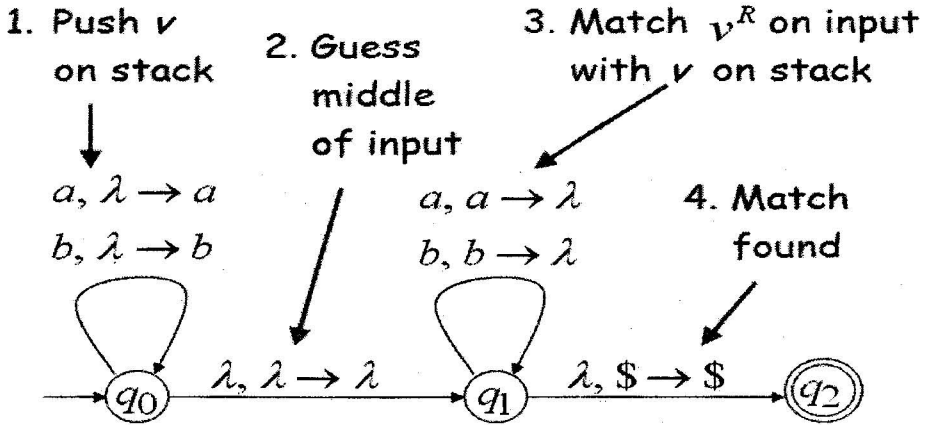
$b, \lambda \rightarrow b$

$b, b \rightarrow \lambda$



والشكل التالي يوضح مدلولات عملية الانتقال من حالة لأخرى:

Basic Idea:  $L(M) = \{vv^R : v \in \{a, b\}^*\}$

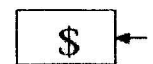
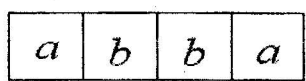


لنتتبع تنفيذ هذه الآلة خطوة خطوة

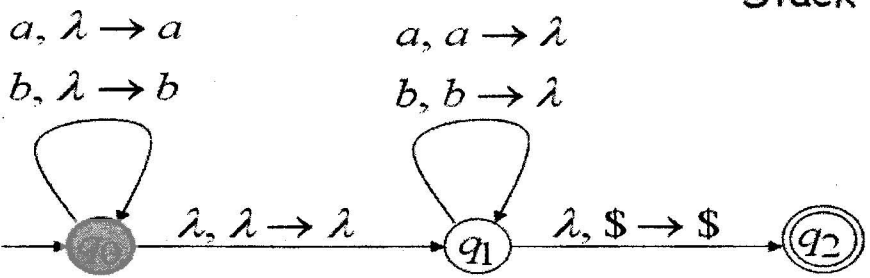
.1

Time 0

Input

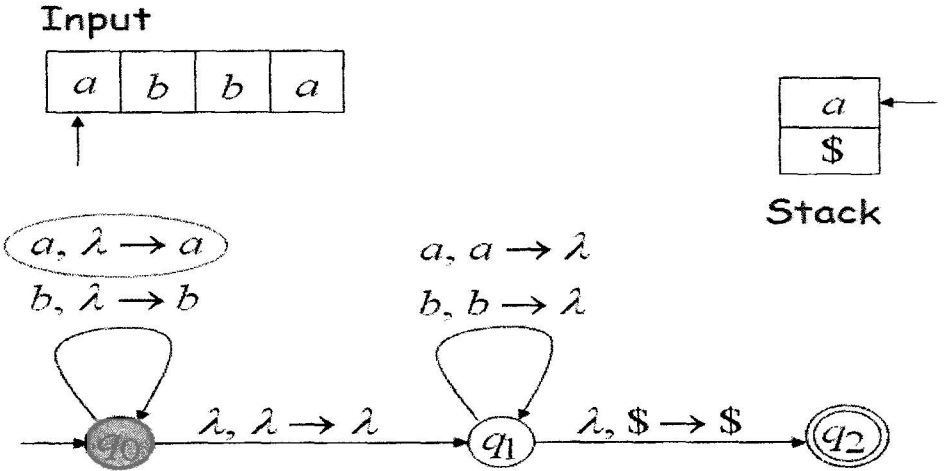


Stack



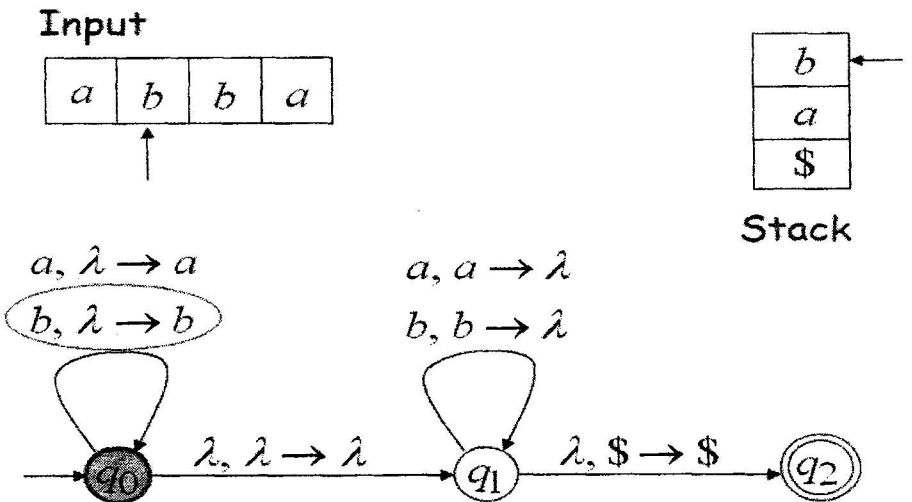
.2

Time 1



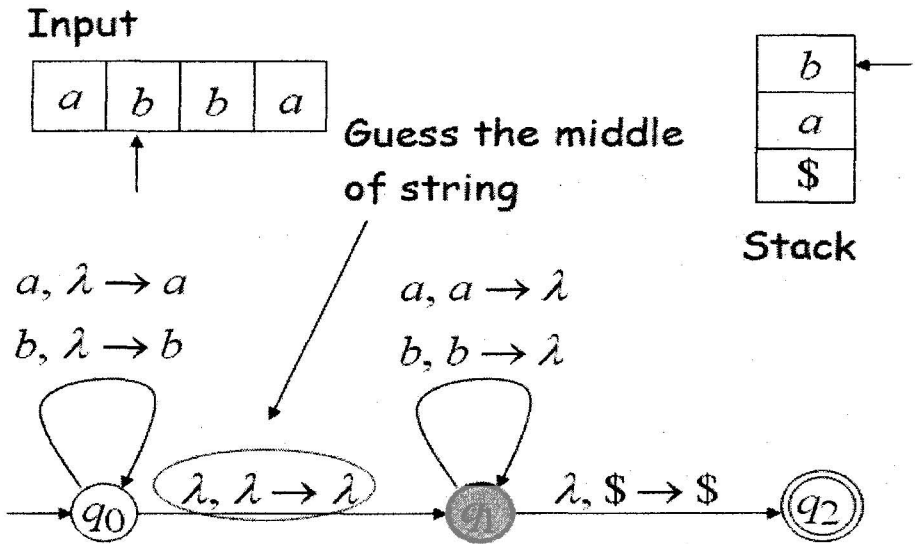
.3

Time 2



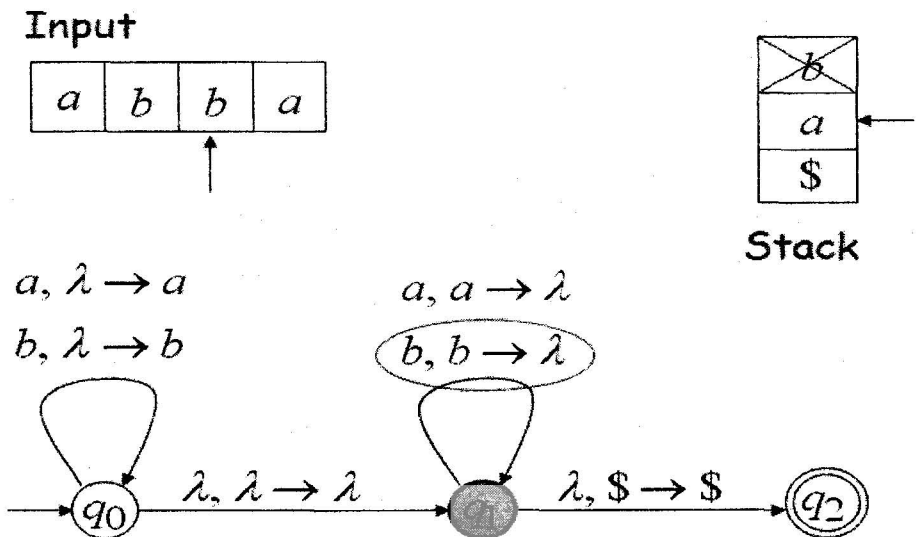
.4

Time 3



.5

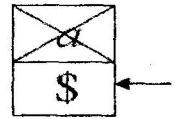
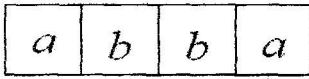
Time 4



.6

Time 5

Input



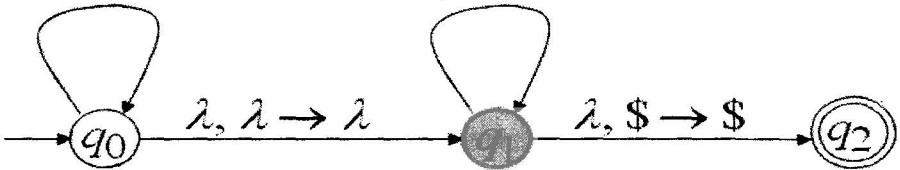
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

$a, a \rightarrow \lambda$

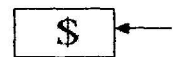
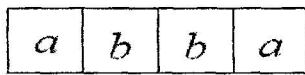
$b, b \rightarrow \lambda$



.7

Time 6

Input



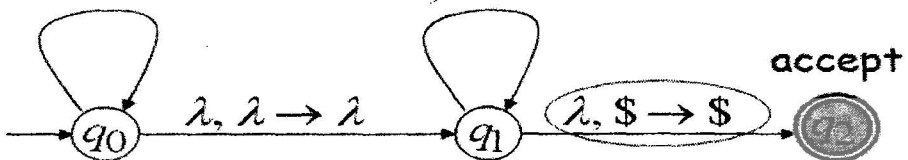
Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

$a, a \rightarrow \lambda$

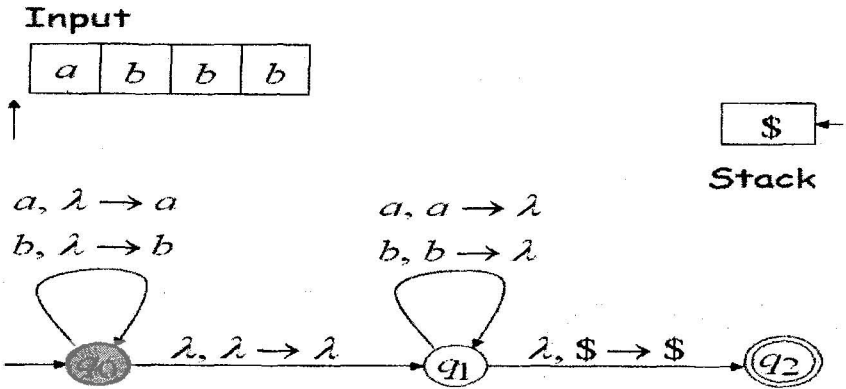
$b, b \rightarrow \lambda$



مثال:

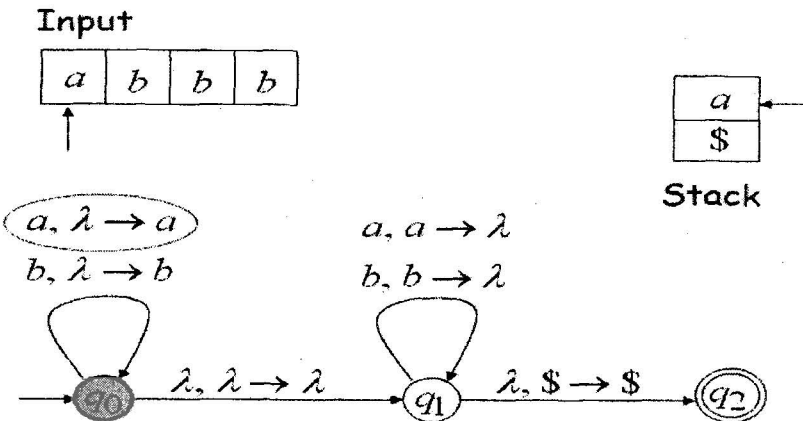
عملية رفض الرموز

Rejection Example: Time 0



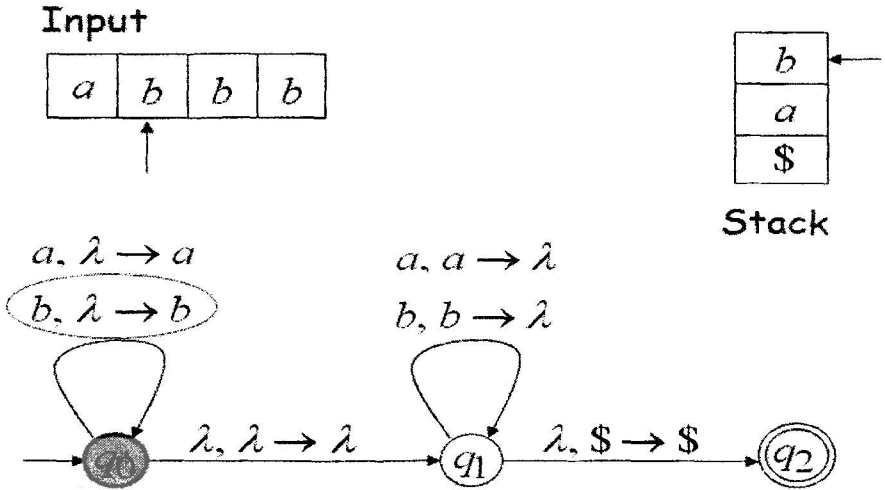
.1

Time 1



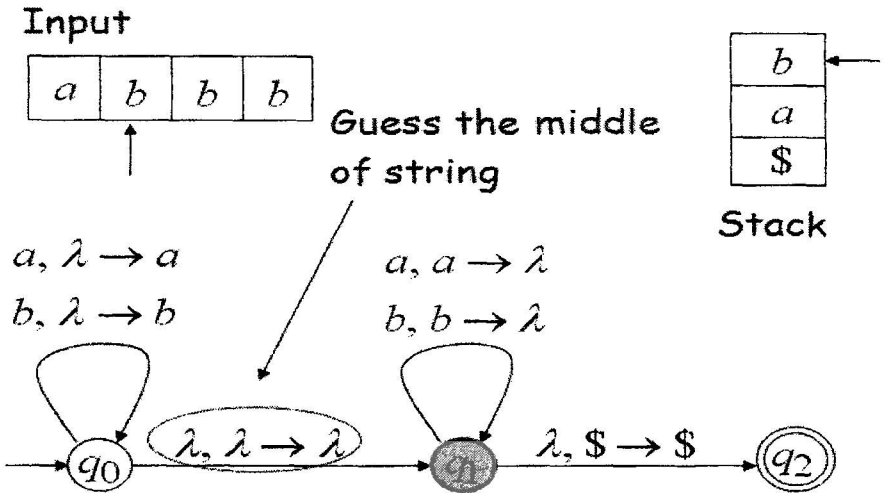
.2

Time 2



.3

Time 3

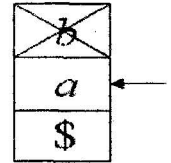
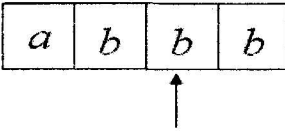




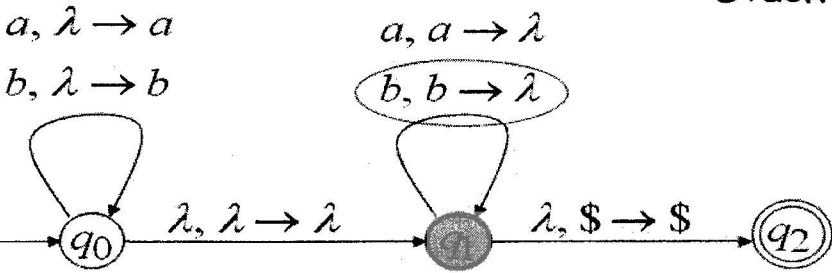
.4

Time 4

Input



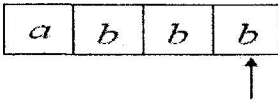
Stack



.5

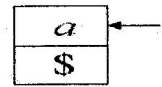
Time 5

Input

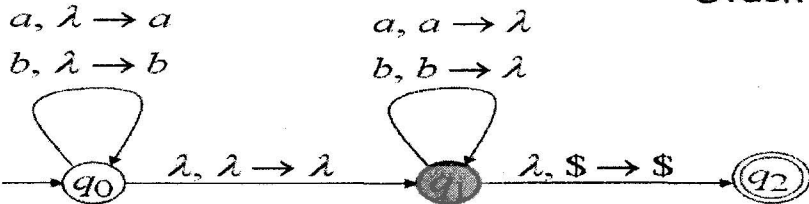


There is no possible transition.

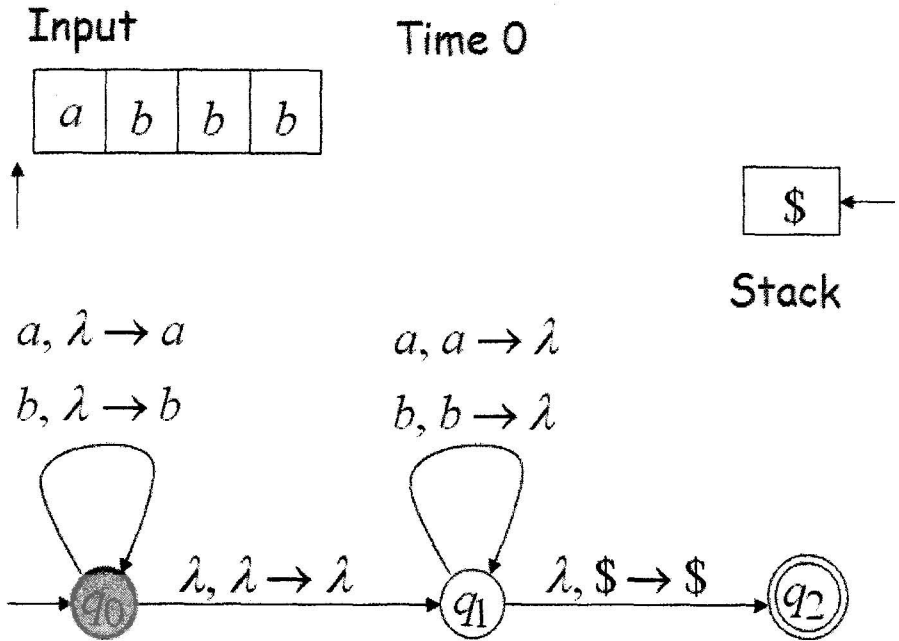
Input is not consumed



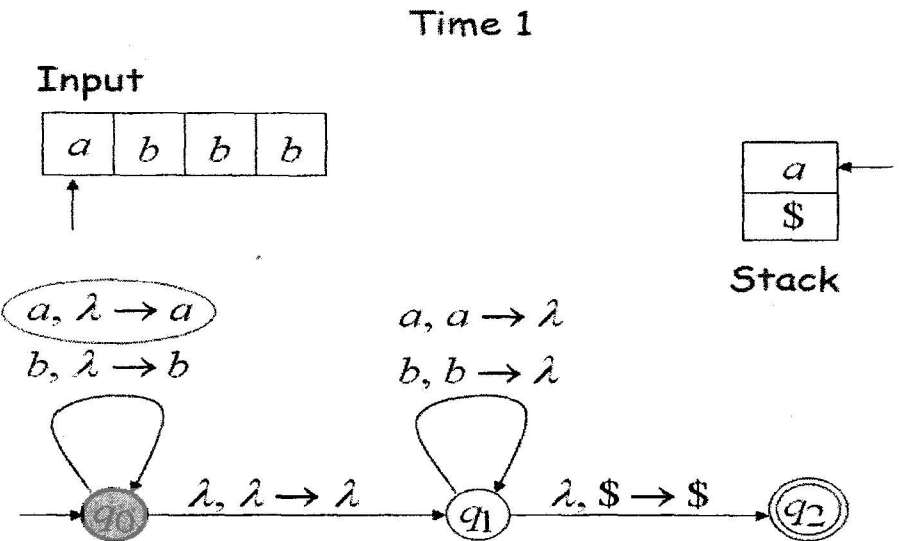
Stack



مثال آخر:

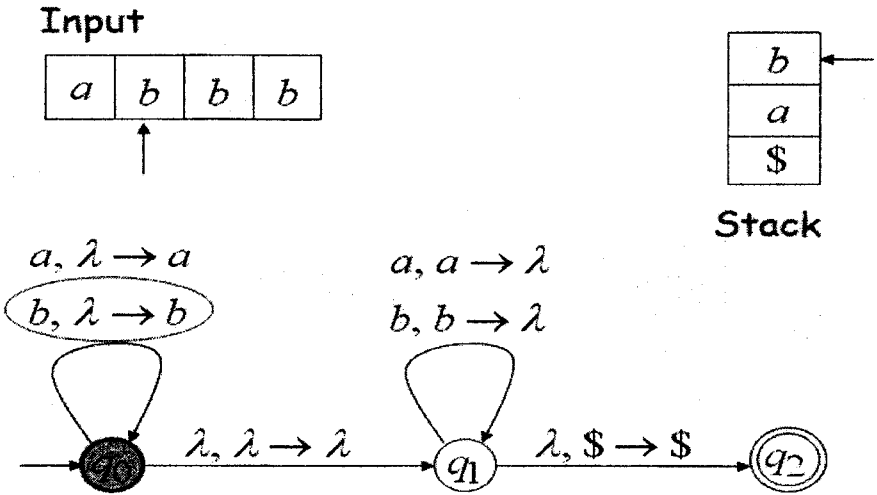


.1



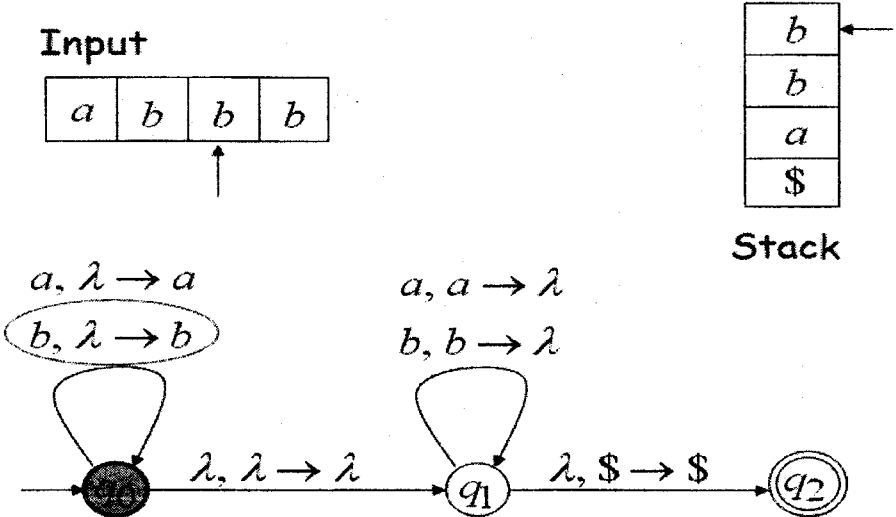
.2

Time 2

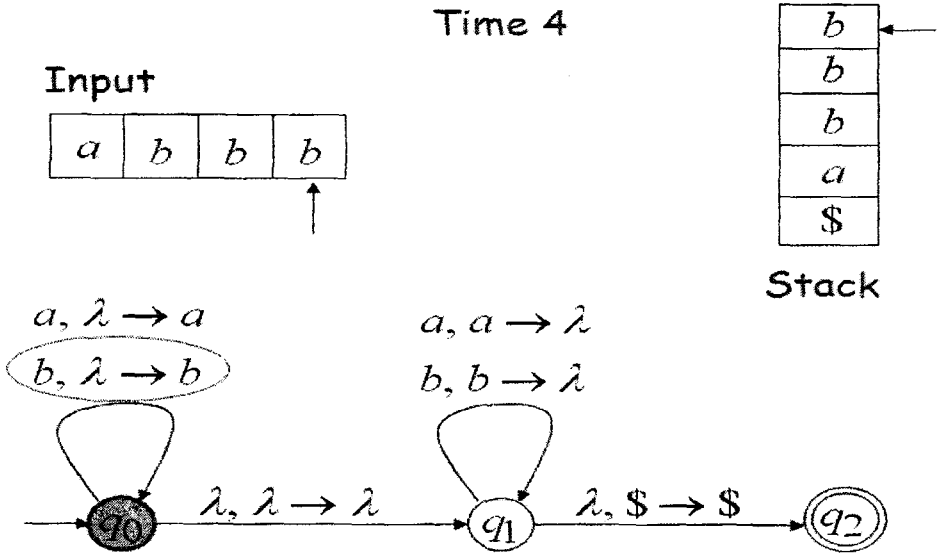


.3

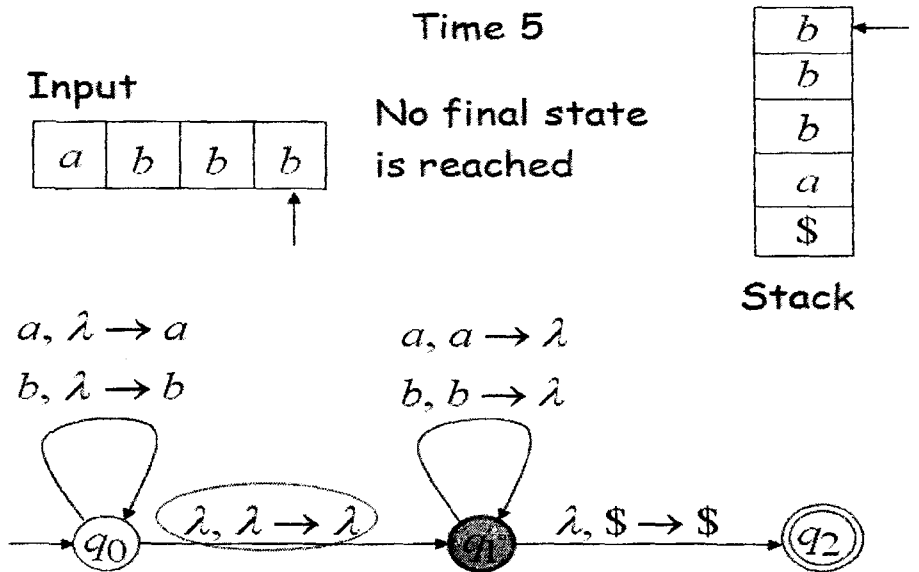
Time 3



.4

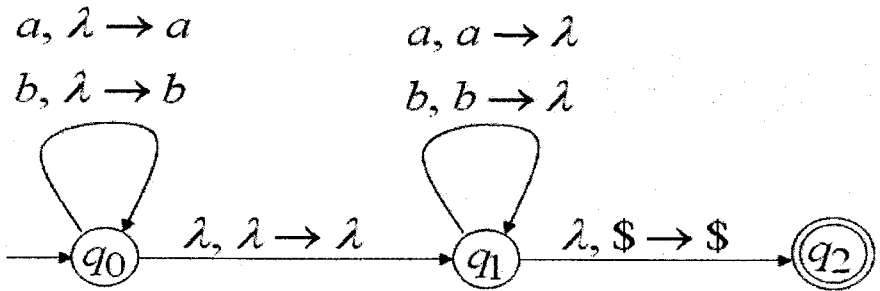


.5



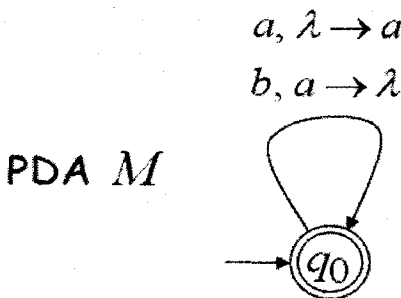
There is no computation  
that accepts string  $abbb$

$$abbb \notin L(M)$$



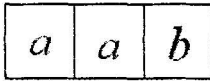
مثال:

$$L(M) = \{w \in \{a, b\}^* : \text{in every prefix } v, n_a(v) \geq n_b(v)\}$$



Execution Example: Time 0

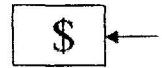
Input



$a, \lambda \rightarrow a$

$b, a \rightarrow \lambda$

$b, \$ \rightarrow \lambda$

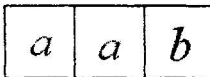


Stack

.1

Time 1

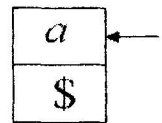
Input



$a, \lambda \rightarrow a$

$b, a \rightarrow \lambda$

$b, \$ \rightarrow \lambda$

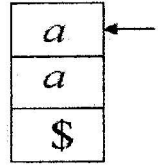
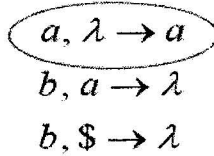
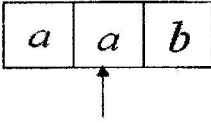


Stack

.2

Time 2

Input

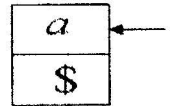
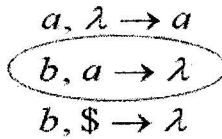
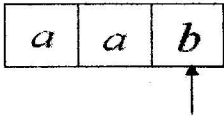


Stack

.3

Time 3

Input



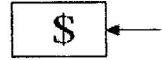
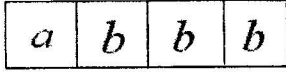
Stack

accept

عملية الرفض:

Rejection example: Time 0

Input



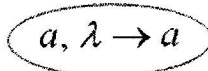
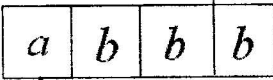
Stack



.1

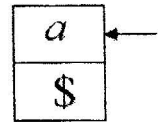
Time 1

Input



$b, a \rightarrow \lambda$

$b, \$ \rightarrow \lambda$



Stack

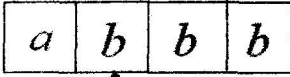




.2

Time 2

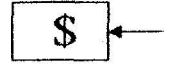
Input



$a, \lambda \rightarrow a$

$b, a \rightarrow \lambda$

$b, \$ \rightarrow \lambda$

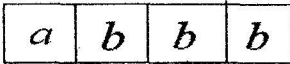


Stack

.3

Time 3

Input



$a, \lambda \rightarrow a$

$b, a \rightarrow \lambda$

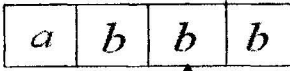
$b, \$ \rightarrow \lambda$



Stack

Time 4

Input

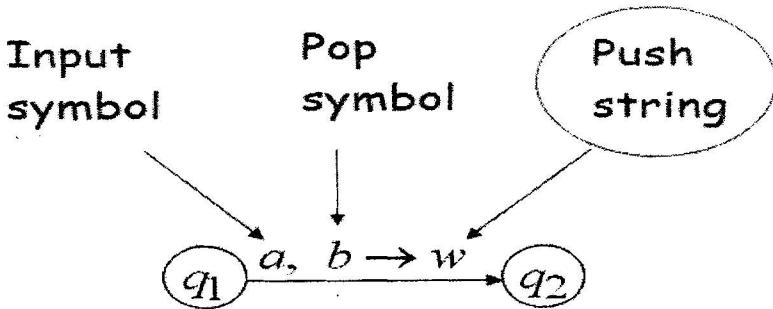


$a, \lambda \rightarrow a$   
 $b, a \rightarrow \lambda$   
 $b, \$ \rightarrow \lambda$

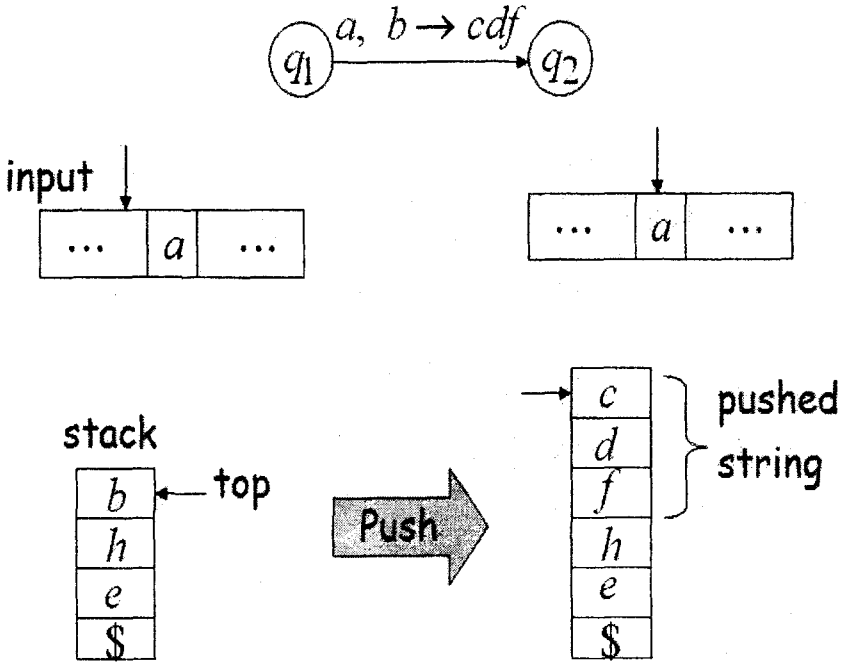


Halt and Reject

اشرنا سابقا الى ان عملية الإنتقال من حالة لحالة تصاحبها عملية اضافة رمز الى الحزمة هذا ويمكن تنفيذ عملية الاضافة لكثر من رمز عند الانتقال من حالة لأخرى بحيث تخزن هذه الرموز في الحزمة تباعا ويبين الشكل التالي تمثيل عملية الانتقال من حالة لاخرى بتسجيل أو اضافة مجموعة من الرموز الى الحزمة:

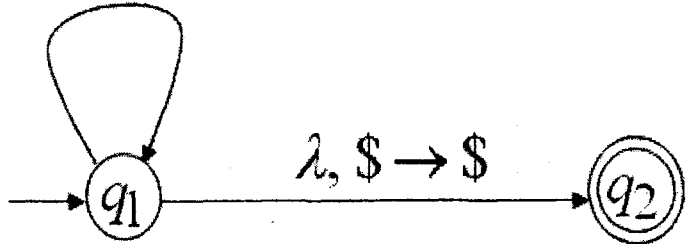


والمثال التالي يوضح هذه العملية:



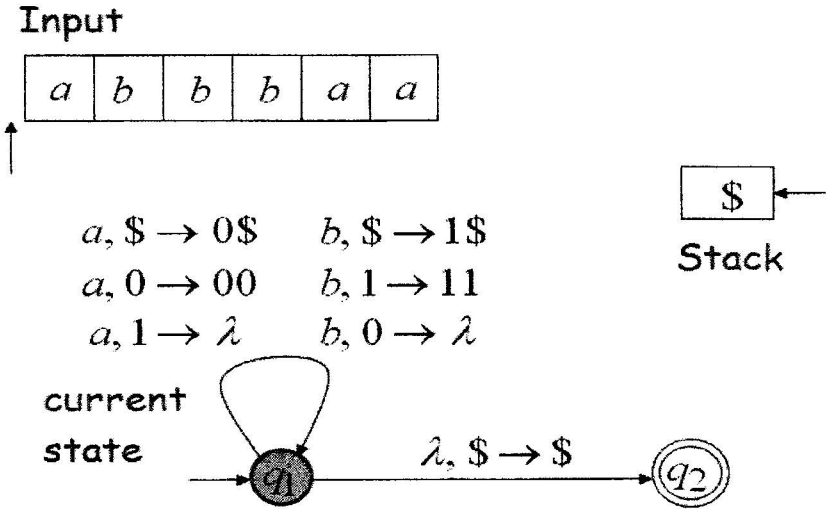
مثال:

- $a, \$ \rightarrow 0\$$       $b, \$ \rightarrow 1\$$
- $a, 0 \rightarrow 00$       $b, 1 \rightarrow 11$
- $a, 1 \rightarrow \lambda$       $b, 0 \rightarrow \lambda$



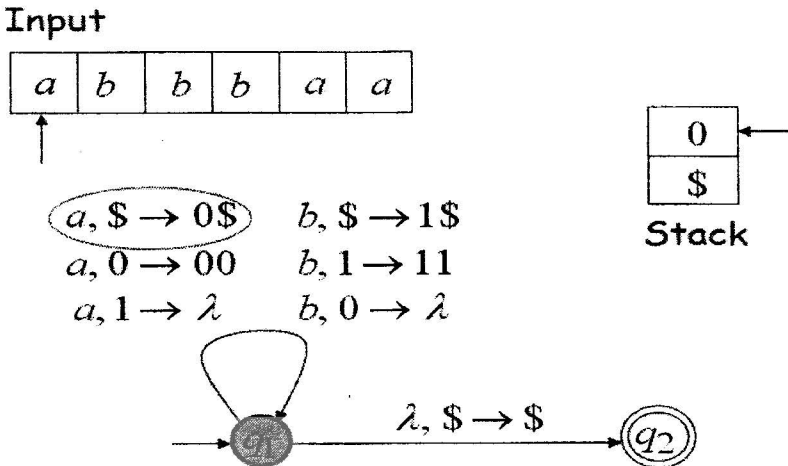
لنتبع هذه الآلة باخذ مجموعة من الرموز

Time 0



.1

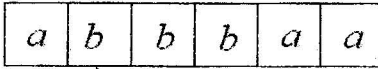
Time 1



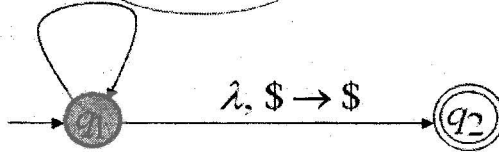
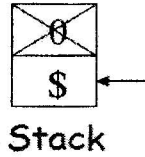
.2

Time 2

Input



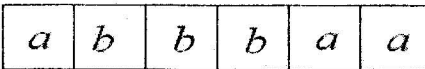
$a, \$ \rightarrow 0\$$      $b, \$ \rightarrow 1\$$   
 $a, 0 \rightarrow 00$      $b, 1 \rightarrow 11$   
 $a, 1 \rightarrow \lambda$      $b, 0 \rightarrow \lambda$



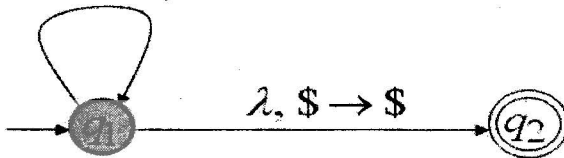
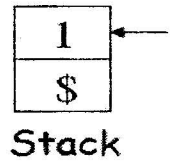
.3

Time 3

Input



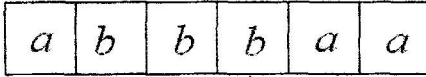
$a, \$ \rightarrow 0\$$      $b, \$ \rightarrow 1\$$   
 $a, 0 \rightarrow 00$      $b, 1 \rightarrow 11$   
 $a, 1 \rightarrow \lambda$      $b, 0 \rightarrow \lambda$



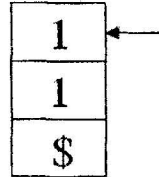
.4

Time 4

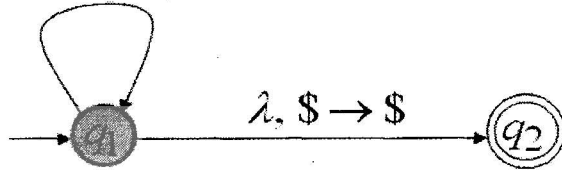
Input



$a, \$ \rightarrow 0\$$        $b, \$ \rightarrow 1\$$   
 $a, 0 \rightarrow 00$        $b, 1 \rightarrow 11$   
 $a, 1 \rightarrow \lambda$        $b, 0 \rightarrow \lambda$



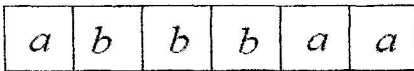
Stack



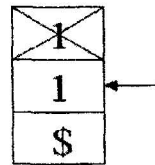
.5

Time 5

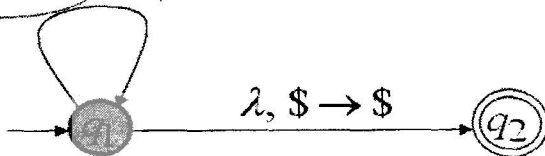
Input



$a, \$ \rightarrow 0\$$        $b, \$ \rightarrow 1\$$   
 $a, 0 \rightarrow 00$        $b, 1 \rightarrow 11$   
 $a, 1 \rightarrow \lambda$        $b, 0 \rightarrow \lambda$



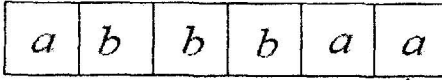
Stack



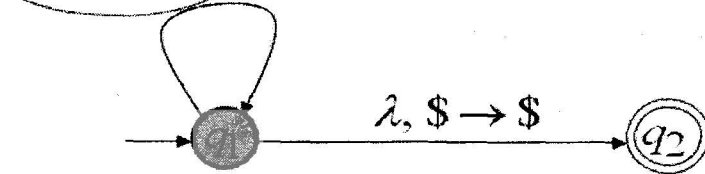
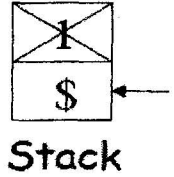
.6

Time 6

Input



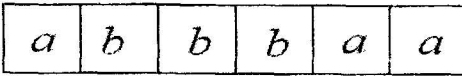
$a, \$ \rightarrow 0\$$      $b, \$ \rightarrow 1\$$   
 $a, 0 \rightarrow 00$      $b, 1 \rightarrow 11$   
 $a, 1 \rightarrow \lambda$      $b, 0 \rightarrow \lambda$



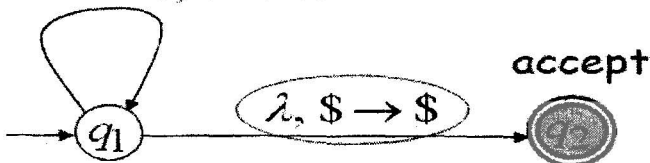
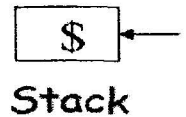
.7

Time 7

Input



$a, \$ \rightarrow 0\$$      $b, \$ \rightarrow 1\$$   
 $a, 0 \rightarrow 00$      $b, 1 \rightarrow 11$   
 $a, 1 \rightarrow \lambda$      $b, 0 \rightarrow \lambda$

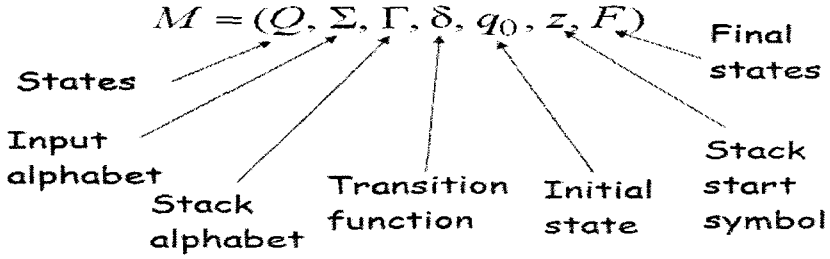


2.4 التعريف الشكلي لآلة الحالة المنتهية المستخدمة للحزمة:

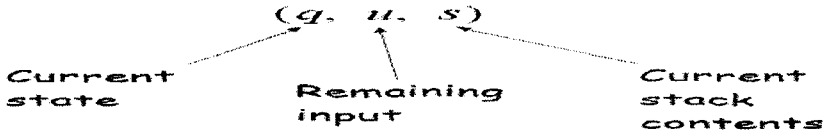
يضم النموذج الرياضي لآلة الحالة المنتهية المستخدمة للحزمة وكما هو

مبين في الشكل التالي مجموعة من المكونات هي:

- مجموعة الحالات بما فيها الحالة الابتدائية والحالات النهائية والمرحلية.
- مجموعة رموز المدخلات والمراد التعرف عليها أو رفضها.
- مجموعة رموز الحزمة.
- دالة الانتقال.
- الحالة الابتدائية.
- رمز البداية للحزمة.
- مجموعة الحالات النهائية.



اما هبة الآلة أو عملية الوصف الآنية فتضم:

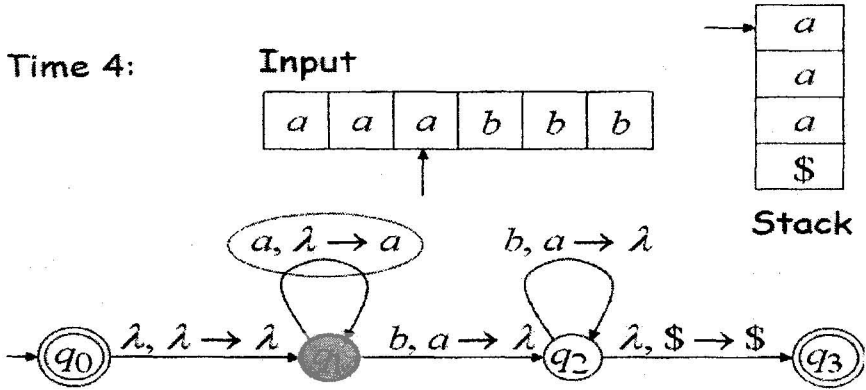


- الحالة الحالية.
- الرموز المتبقية للقراءة.
- محتوى الحزمة الحالي من الرموز.



والشكل التالي يبين هيئة الآلة في لحظة زمنية معينة:

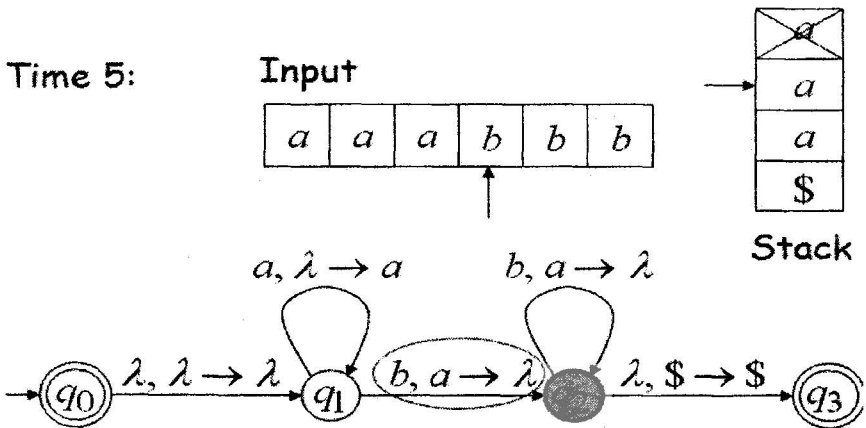
$(q_1, bbb, aaa\$)$



وتغيير الهيئة من عملية انتقال الى اخرى لتغيير الحالة وحالة الشريط

وحالة الحزمة ويبين الشكل التالي هيئة الآلة اللاحقة للهيئة المبينة اعلاه:

$(q_2, bb, aa\$)$



وتمثل عملية الانتقال من هيئة في الحظة الرابعة الى هيئة جديدة في

الحظة الخامسة كما يلي:

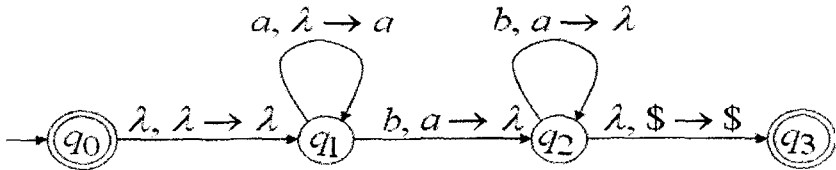
$(q_1, bbb, aaa\$) \rightarrow (q_2, bb, aa\$)$

Time 4

Time 5

ويمكن استخدام الهيئات لحساب الآلة كما يلي:

$$\begin{aligned} (q_0, aaabbb, \$) &\succ (q_1, aaabbb, \$) \succ \\ (q_1, aabbb, a\$) &\succ (q_1, abbb, aa\$) \succ (q_1, bbb, aaa\$) \succ \\ (q_2, bb, aa\$) &\succ (q_2, b, a\$) \succ (q_2, \lambda, \$) \succ (q_3, \lambda, \$) \end{aligned}$$



ولتسهيل عملية التمثيل يمكن اختصار عملية الحساب هذه كما يلي:

$$(q_0, aaabbb, \$) \overset{*}{\succ} (q_3, \lambda, \$)$$

وبهذا يمكن تعريف الآلة المنتهية باستخدام مفهوم الهيئة على انها آلة تبء

بهيئة ابتدائية وتنتهي بهيئة نهائية وكما هو مبين أدناه حيث تشكل عملية

الانتقال هذه لغة الآلة:

$$L(M) = \{w : (q_0, w, s) \overset{*}{\succ} (q_f, \lambda, s')\}$$

Initial state

Final state

مثال:

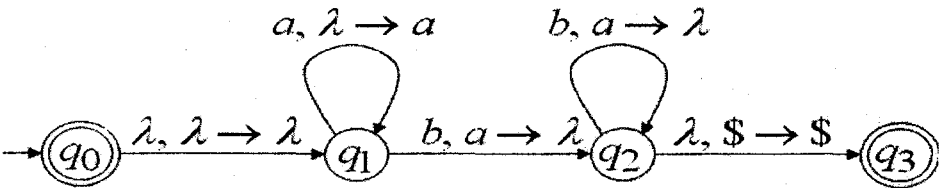
Example:

$$(q_0, aaabbb, \$) \xrightarrow{*} (q_3, \lambda, \$)$$

↓

$$aaabbb \in L(M)$$

PDA  $M$ :

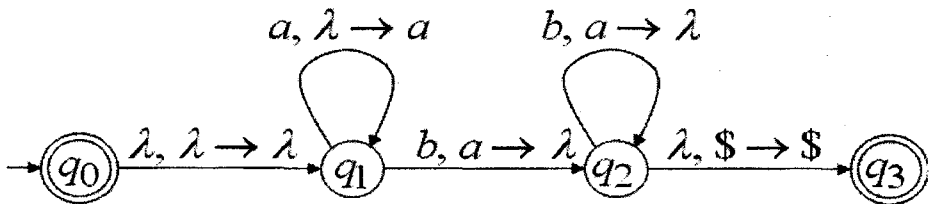


$$(q_0, a^n b^n, \$) \xrightarrow{*} (q_3, \lambda, \$)$$

↓

$$a^n b^n \in L(M)$$

PDA  $M$ :



3.4 تمارين:

1. صمم الآلة المنتهية لموازنة الأقواس الدائرية المفتوحة والأقواس الدائرية المغلقة.

الحل:

$$M = (\{q_1\}, \{“(“,”)\}, \{L, \#\}, \delta, q_1, \#, \emptyset)$$

$\delta$ :

$$(1) \quad \delta(q_1, (, \#) = \{(q_1, L\#)\}$$

$$(2) \quad \delta(q_1, ), \#) = \emptyset$$

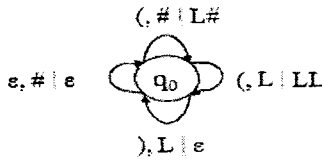
$$(3) \quad \delta(q_1, (, L) = \{(q_1, LL)\}$$

$$(4) \quad \delta(q_1, ), L) = \{(q_1, \epsilon)\}$$

$$(5) \quad \delta(q_1, \epsilon, \#) = \{(q_1, \epsilon)\}$$

$$(6) \quad \delta(q_1, \epsilon, L) = \emptyset$$

• Transition Diagram:



• Example Computation:

Current Input	Stack	Transition	
(()	#		
()	L#	(1)	- Could have applied rule (5), but it would have done no good
)	LL#	(3)	
)	L#	(4)	
$\epsilon$	#	(4)	
$\epsilon$	-	(5)	

## • Example Computation:

- (1)  $\delta(q_1, 0, R) = \{(q_1, BR)\}$  (9)  $\delta(q_1, 1, R) = \{(q_1, GR)\}$   
 (2)  $\delta(q_1, 0, B) = \{(q_1, BB)\}$  (10)  $\delta(q_1, 1, B) = \{(q_1, GB)\}$   
 (3)  $\delta(q_1, 0, G) = \{(q_1, BG)\}$  (11)  $\delta(q_1, 1, G) = \{(q_1, GG)\}$   
 (4)  $\delta(q_1, c, R) = \{(q_2, R)\}$   
 (5)  $\delta(q_1, c, B) = \{(q_2, B)\}$   
 (6)  $\delta(q_1, c, G) = \{(q_2, G)\}$   
 (7)  $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$  (12)  $\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}$   
 (8)  $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$

State	Input	Stack	Rule Applied	Rules Applicable
$q_1$	01c10	R		(1)
$q_1$	1c10	BR	(1)	(10)
$q_1$	c10	GBR	(10)	(6)
$q_2$	10	GBR	(6)	(12)
$q_2$	0	BR	(12)	(7)
$q_2$	$\varepsilon$	R	(7)	(8)
$q_2$	$\varepsilon$	$\varepsilon$	(8)	-

## • Example Computation:

- (1)  $\delta(q_1, 0, R) = \{(q_1, BR)\}$  (9)  $\delta(q_1, 1, R) = \{(q_1, GR)\}$   
 (2)  $\delta(q_1, 0, B) = \{(q_1, BB)\}$  (10)  $\delta(q_1, 1, B) = \{(q_1, GB)\}$   
 (3)  $\delta(q_1, 0, G) = \{(q_1, BG)\}$  (11)  $\delta(q_1, 1, G) = \{(q_1, GG)\}$   
 (4)  $\delta(q_1, c, R) = \{(q_2, R)\}$   
 (5)  $\delta(q_1, c, B) = \{(q_2, B)\}$   
 (6)  $\delta(q_1, c, G) = \{(q_2, G)\}$   
 (7)  $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$  (12)  $\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}$   
 (8)  $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$

State	Input	Stack	Rule Applied
$q_1$	1c1	R	
$q_1$	c1	GR	(9)
$q_2$	1	GR	(6)
$q_2$	$\varepsilon$	R	(12)
$q_2$	$\varepsilon$	$\varepsilon$	(8)

مثال:

- **Example PDA :** For the language  $\{x \mid x = ww^r \text{ and } w \text{ in } \{0,1\}^*\}$

$$M = (\{q_1, q_2\}, \{0, 1\}, \{R, B, G\}, \delta, q_1, R, \emptyset)$$

$\delta$ :

- |     |  |      |  |
|-----|--|------|--|
| (1) | $\delta(q_1, 0, R) = \{(q_1, BR)\}$                  |      |  |
| (2) | $\delta(q_1, 1, R) = \{(q_1, GR)\}$                  |      |  |
| (3) | $\delta(q_1, 0, B) = \{(q_1, BB), (q_2, \epsilon)\}$ | (6)  | $\delta(q_1, 1, G) = \{(q_1, GG), (q_2, \epsilon)\}$ |
| (4) | $\delta(q_1, 0, G) = \{(q_1, BG)\}$                  | (7)  | $\delta(q_2, 0, B) = \{(q_2, \epsilon)\}$            |
| (5) | $\delta(q_1, 1, B) = \{(q_1, GB)\}$                  | (8)  | $\delta(q_2, 1, G) = \{(q_2, \epsilon)\}$            |
|     |  | (9)  | $\delta(q_1, \epsilon, R) = \{(q_2, \epsilon)\}$     |
|     |  | (10) | $\delta(q_2, \epsilon, R) = \{(q_2, \epsilon)\}$     |

- **Notes:**

- Rules #3 and #6 are non-deterministic.
- Rules #9 and #10 are used to pop the final stack symbol off at the end of a computation.

- **Example Computation:**

- |     |  |      |  |
|-----|--|------|--|
| (1) | $\delta(q_1, 0, R) = \{(q_1, BR)\}$                  | (6)  | $\delta(q_1, 1, G) = \{(q_1, GG), (q_2, \epsilon)\}$ |
| (2) | $\delta(q_1, 1, R) = \{(q_1, GR)\}$                  | (7)  | $\delta(q_2, 0, B) = \{(q_2, \epsilon)\}$            |
| (3) | $\delta(q_1, 0, B) = \{(q_1, BB), (q_2, \epsilon)\}$ | (8)  | $\delta(q_2, 1, G) = \{(q_2, \epsilon)\}$            |
| (4) | $\delta(q_1, 0, G) = \{(q_1, BG)\}$                  | (9)  | $\delta(q_1, \epsilon, R) = \{(q_2, \epsilon)\}$     |
| (5) | $\delta(q_1, 1, B) = \{(q_1, GB)\}$                  | (10) | $\delta(q_2, \epsilon, R) = \{(q_2, \epsilon)\}$     |

State	Input	Stack	Rule Applied	Rules Applicable
$q_1$	000000	R		(1), (9)
$q_1$	00000	BR	(1)	(3), both options
$q_1$	0000	BBR	(3) option #1	(3), both options
$q_1$	000	BBBR	(3) option #1	(3), both options
$q_2$	00	BBR	(3) option #2	(7)
$q_2$	0	BR	(7)	(7)
$q_2$	$\epsilon$	R	(7)	(10)
$q_2$	$\epsilon$	$\epsilon$	(10)	

• Example Computation:

- |     |  |      |  |
|-----|--|------|--|
| (1) | $\delta(q_1, 0, R) = \{(q_1, BR)\}$                  | (6)  | $\delta(q_1, 1, G) = \{(q_1, GG), (q_2, \epsilon)\}$ |
| (2) | $\delta(q_1, 1, R) = \{(q_1, GR)\}$                  | (7)  | $\delta(q_2, 0, B) = \{(q_2, \epsilon)\}$            |
| (3) | $\delta(q_1, 0, B) = \{(q_1, BB), (q_2, \epsilon)\}$ | (8)  | $\delta(q_2, 1, G) = \{(q_2, \epsilon)\}$            |
| (4) | $\delta(q_1, 0, G) = \{(q_1, BG)\}$                  | (9)  | $\delta(q_1, \epsilon, R) = \{(q_2, \epsilon)\}$     |
| (5) | $\delta(q_1, 1, B) = \{(q_1, GB)\}$                  | (10) | $\delta(q_2, \epsilon, R) = \{(q_2, \epsilon)\}$     |

<u>State</u>	<u>Input</u>	<u>Stack</u>	<u>Rule Applied</u>
$q_1$	010010	R	
$q_1$	10010	BR	(1) From (1) and (9)
$q_1$	0010	GBR	(5)
$q_1$	010	BGBR	(4)
$q_2$	10	GBR	(3) option #2
$q_2$	0	BR	(8)
$q_2$	$\epsilon$	R	(7)
$q_2$	$\epsilon$	$\epsilon$	(10)

مثال:

• Example : Consider the following CFG in GNF.

- (1)  $S \rightarrow aA$
- (2)  $S \rightarrow aB$
- (3)  $A \rightarrow aA$                        $G$  is in GNF
- (4)  $A \rightarrow aB$                        $L(G) = a^+b^+$
- (5)  $B \rightarrow bB$
- (6)  $B \rightarrow b$

Construct  $M$  as:

- $Q = \{q\}$
- $\Sigma = T = \{a, b\}$
- $\Gamma = V = \{S, A, B\}$
- $z = S$

- (1)  $\delta(q, a, S) = \{(q, A), (q, B)\}$                       From productions #1 and 2,  $S \rightarrow aA, S \rightarrow aB$
- (2)  $\delta(q, a, A) = \{(q, A), (q, B)\}$                       From productions #3 and 4,  $A \rightarrow aA, A \rightarrow aB$
- (3)  $\delta(q, a, B) = \emptyset$
- (4)  $\delta(q, b, S) = \emptyset$
- (5)  $\delta(q, b, A) = \emptyset$
- (6)  $\delta(q, b, B) = \{(q, B), (q, \epsilon)\}$                       From productions #5 and 6,  $B \rightarrow bB, B \rightarrow b$
- (7)  $\delta(q, \epsilon, S) = \emptyset$
- (8)  $\delta(q, \epsilon, A) = \emptyset$
- (9)  $\delta(q, \epsilon, B) = \emptyset$                       Recall  $\delta: Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow$  finite subsets of  $Q \times \Gamma^*$



مثال:

• Example : Consider the following CFG in GNF.

$$(1) S \rightarrow aABC$$

$$(2) A \rightarrow a \quad G \text{ is in GNF}$$

$$(3) B \rightarrow b$$

$$(4) C \rightarrow cAB$$

$$(5) C \rightarrow cC$$

Construct M as:

$$Q = \{q\}$$

$$\Sigma = T = \{a, b, c\}$$

$$\Gamma = V = \{S, A, B, C\}$$

$$z = S$$

$$(1) \delta(q, a, S) = \{(q, ABC)\} \quad S \rightarrow aABC$$

$$(2) \delta(q, a, A) = \{(q, \epsilon)\} \quad A \rightarrow a$$

$$(3) \delta(q, a, B) = \emptyset$$

$$(4) \delta(q, a, C) = \emptyset$$

$$>cAB\}cC$$

$$(5) \delta(q, b, S) = \emptyset$$

$$(6) \delta(q, b, A) = \emptyset$$

$$(7) \delta(q, b, B) = \{(q, \epsilon)\} \quad B \rightarrow b$$

$$(8) \delta(q, b, C) = \emptyset$$

$$(9) \delta(q, c, S) = \emptyset$$

$$(10) \delta(q, c, A) = \emptyset$$

$$(11) \delta(q, c, B) = \emptyset$$

$$(12) \delta(q, c, C) = \{(q, AB), (q, C)\} \quad C \rightarrow$$

$$(13) \delta(q, \epsilon, S) = \emptyset$$

$$(14) \delta(q, \epsilon, A) = \emptyset$$

$$(15) \delta(q, \epsilon, B) = \emptyset$$

$$(16) \delta(q, \epsilon, C) = \emptyset$$

مثال:

## • Example:

Consider  $L = \{\epsilon, b, ab, aab, aaab, \dots\}$ Then  $L' = \{b, ab, aab, aaab, \dots\}$ • The GNF CFG for  $L'$ :

(1)  $S \rightarrow aS$

(2)  $S \rightarrow b$

• The PDA M Accepting  $L'$ :

$Q = \{q\}$

$\Sigma = \Gamma = \{a, b\}$

$\Gamma = V = \{S\}$

$z = S$

$\delta(q, a, S) = \{(q, S)\}$

$\delta(q, b, S) = \{(q, \epsilon)\}$

$\delta(q, \epsilon, S) = \emptyset$

• If  $\delta(q, \epsilon, S) = \{(q, \epsilon)\}$  is added then:

$L(M) = \{\epsilon, a, aa, aaa, \dots, b, ab, aab, aaab, \dots\}$

الوحدة الخامسة

---

آلة تيورينج

*Turing Machine*

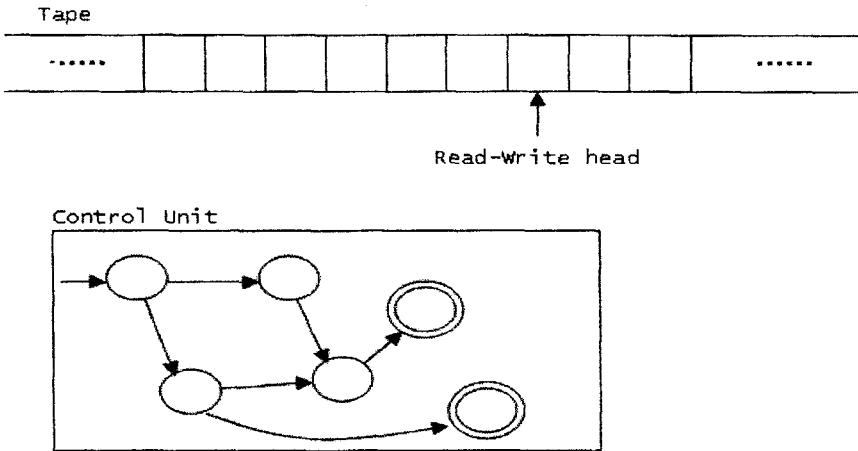
5

### 1.5 التعريف بالة تيورينج:

تعتبر الة تيورينج نموذجا من الالات الحالة المنتهية ولهذه الالة تطبيقات كثيرة خاصة في علوم الحاسوب حيث يمكن استخدام هذه الالة في بناء معالجات النصوص وذلك لتمثيل أهم العمليات في معالجات النصوص مثل عمليات النقل والتحريك والازاحة وغيرها من العمليات والتي سنستعرض بعضها منها في هذه الوحدة ان شاء الله.

تشبه الة تيورينج الات الحالة المنتهية والتي استعرضناها سابقا في الكتاب الا انها تختلف قليلا عنها وفيما يلي سوف نستعرض أهم خصائص هذه الالة:

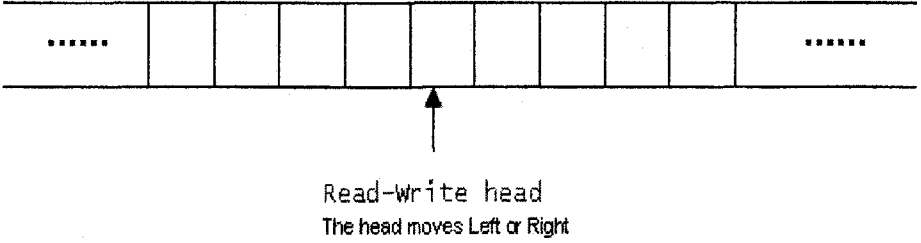
- تتكون الة تيورينج من وحدة تحكم وشريط مدخلات ورأس للقراءة والكتابة وكما هو مبين في الشمل التالي:



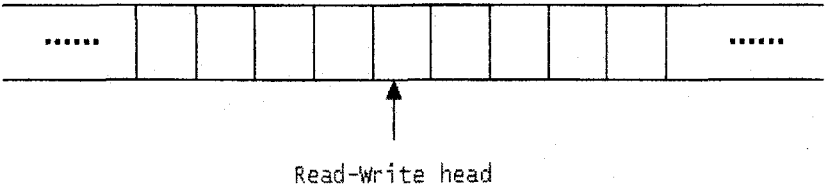
تستخدم وحدة التحكم للاحتفاظ بحالات الالة والناجمة عن عمليات القراءة أو الكتابة أو تحريك رأس القراءة والكتابة لليمين أو اليسار.

- سلسلة الرموز المخزنة على الشريط غير محدودة ويستطيع رأس القراءة والكتابة التحرك لليسار أو الى اليمين ويكون عدد هذه الحركات غير محدود.

No boundaries -- infinite length



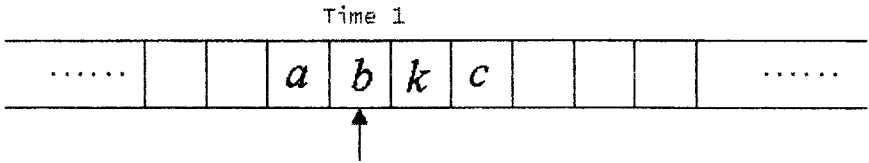
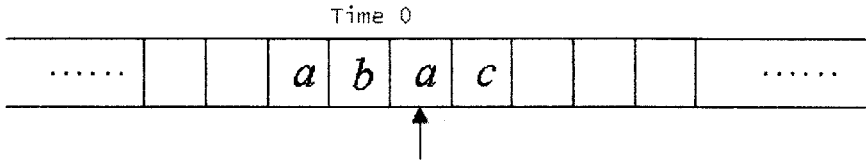
- في اللحظة الزمنية المعينة يمكن لرأس القراءة والكتابة قراءة رمز أو كتابة رمز أو التحرك لليسار أو اليمين ولخطوة واحدة.



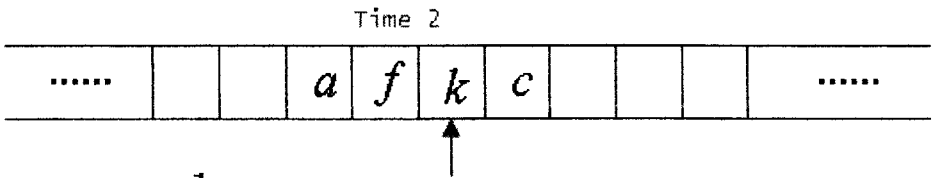
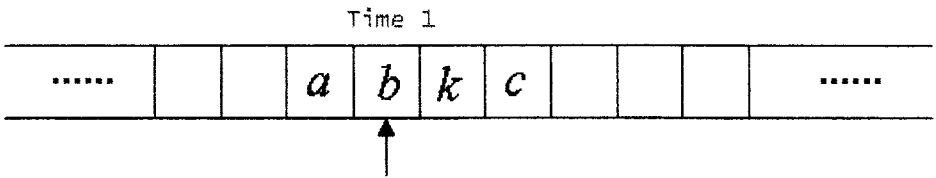
The head at each time step:

1. Reads a symbol
2. Writes a symbol
3. Moves Left or Right

- عملية القراءة أو الكتابة تتم في نفس موقع رأس القراءة والكتابة دون تحريك هذه الرأس اما عماية التحريك فتتم باستخدام الرمز R للحركة لليمين H والرمز L للحركة لليسار وعليه فان مجموعة الرموز يجب ان لا تتضمن هذين الحرفين.

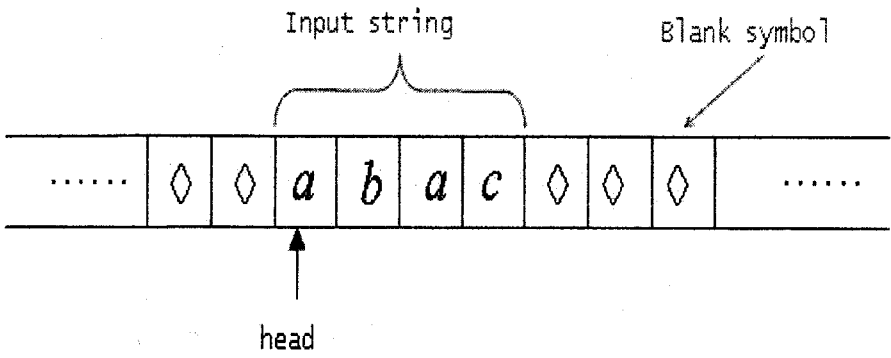


1. Reads *a*
2. Writes *k*
3. Moves left



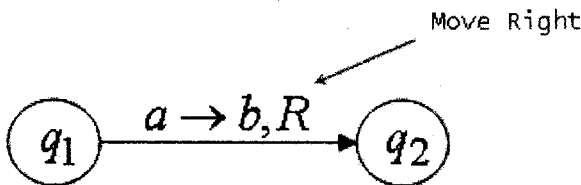
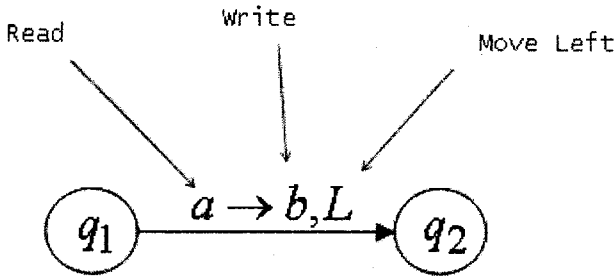
1. Reads *b*
2. Writes *f*
3. Moves right

- يجب أن تتضمن مجموعة الرموز رمز الفراغ وفي الغالب يكون الوضع الابتدائي لرأس القراءة والكتابة عند أول فراغ من اليسار.



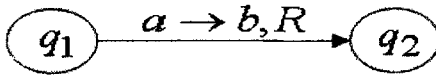
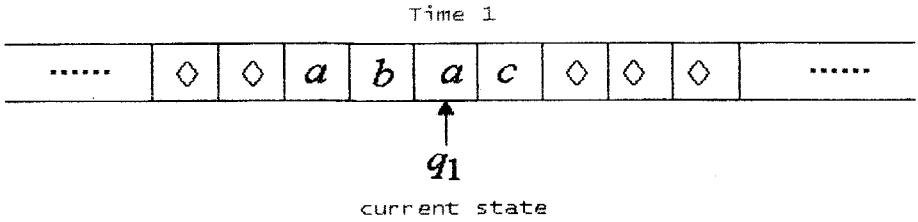
Head starts at the leftmost position of the input string

- تمثل الحالة بالدائرة وعملية الانتقال بالسهم على أن يوضع على محددات عملية الإنتقال من الحالة الحالية الى الحالة التالية وكما هو مبين في الشكل التالي:

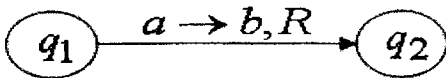
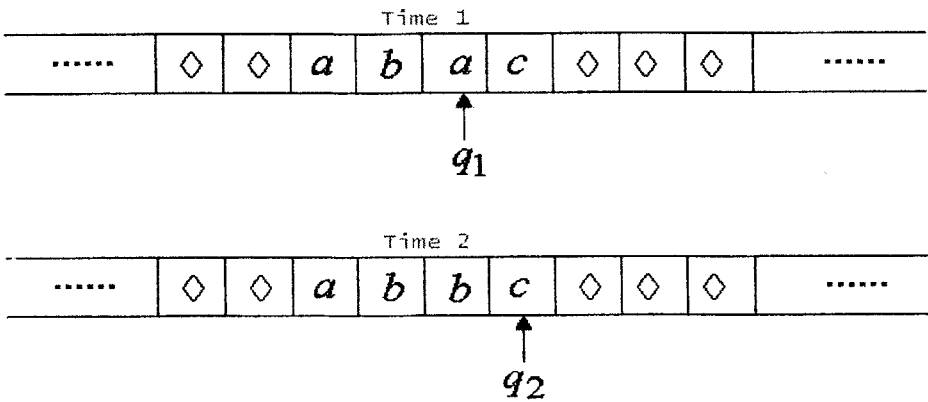


وفيما يلي بعض الاشكال التوضيحية:

.1

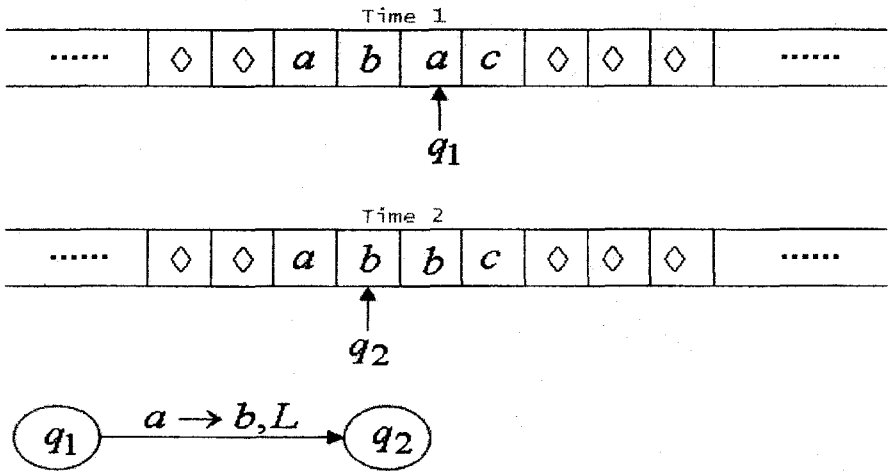


.2

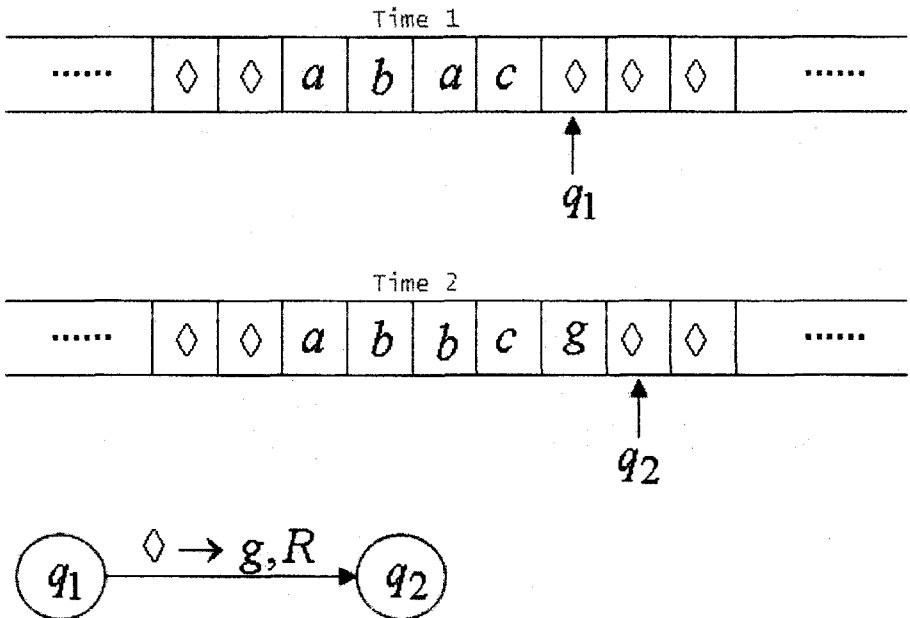




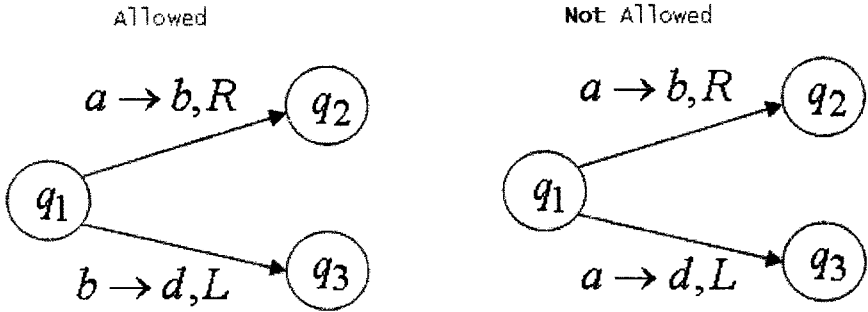
.3



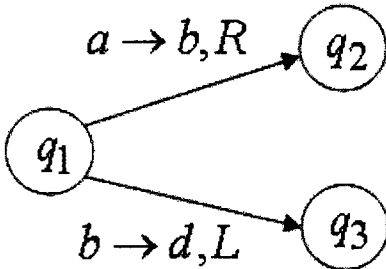
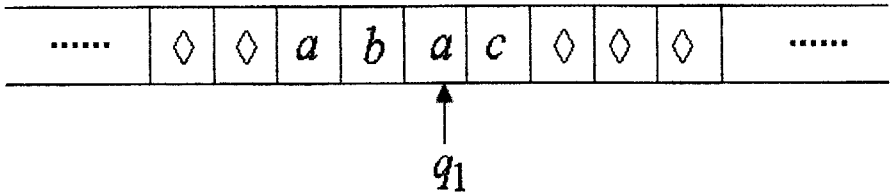
.4



- تعتبر آلة تيورينج من الآلات المحدودة والتي لا تقبل الفرع في المسارات وبهذا فإنها تختلف عن آلة الحالة المنتهية غير المحدودة وتشابه مع آلة الحالة المنتهية المحدودة والشكل التالي يوضح هذه الفكرة.



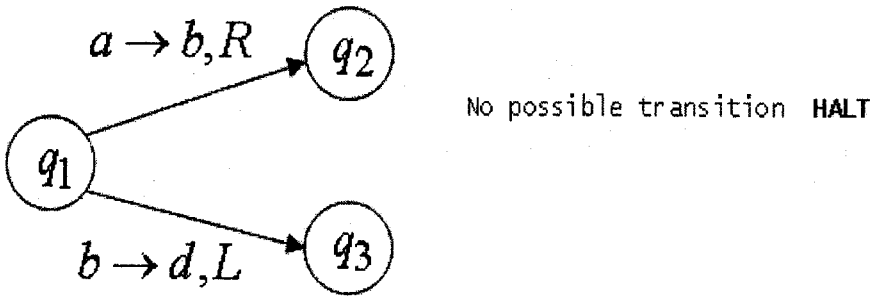
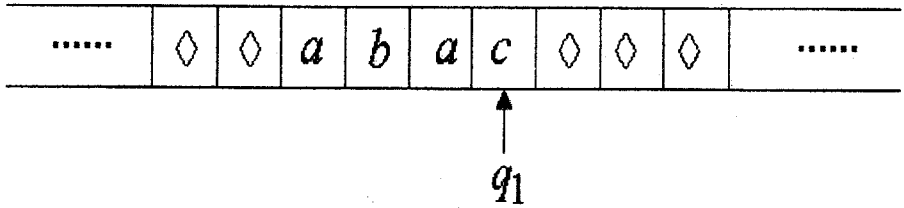
والشكل التالي يبين بعض عمليات الإنتقال المسموحة في آلة تيورينج:



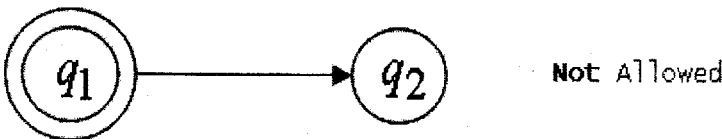
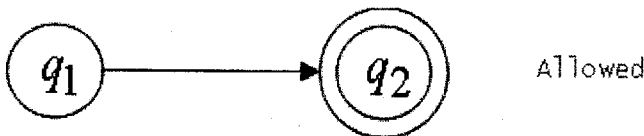
Allowed:

No transition  
for input symbol  $c$

- تنتقل آلة تيورينج الى حالة التوقف في حالة عدم تحقق شروط عملية الانتقال من حالة لاخرى وكما هو مبين في الشكل التالي:



- لا يجوز الانتقال من حالة نهائية الى حالة مقبولة من قبل الآلة ويمكن الانتقال من حالة مقبولة الى حالة نهائية.



- تقبل آلة تيورينج مجموعة المدخلات إذا كانت تقود الى حالة توقف نهائية وترفضها إذا كانت تقود الى حالة ليست نهائية أو حالة لا نهائية.

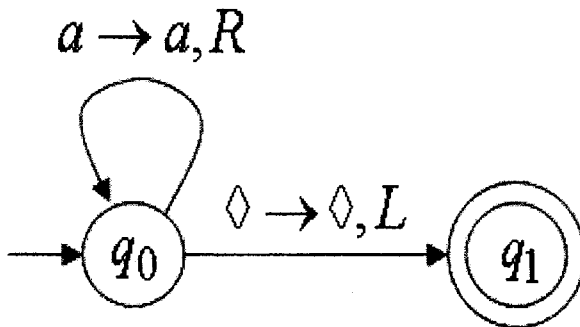
Accept Input  $\longrightarrow$  If machine halts in a final state

Reject Input  $\longrightarrow$  If machine halts in a non-final state  
or  
If machine enters an *infinite loop*

وفيما يلي نستعرض بعض الامثلة والتي تبين كيفية معالجة الرموز والانتقال من حالة لاخرى:

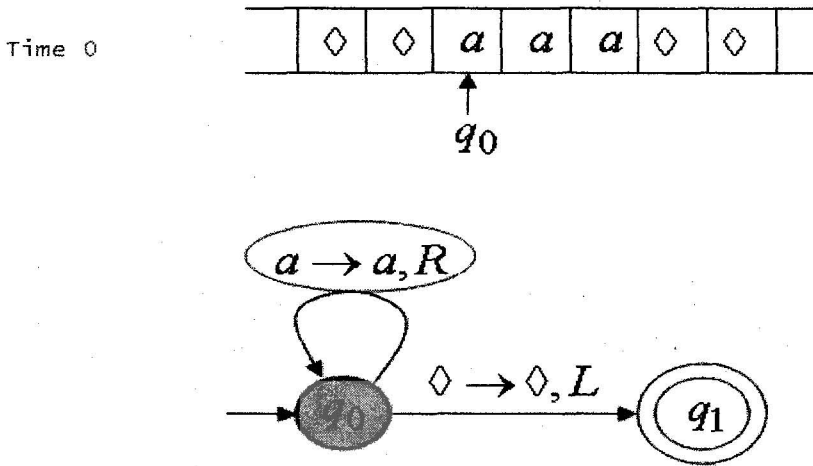
مثال:

ابن الة تيورينج والتي تقبل مجموعة متتابعة من الحرف  $a$  يجب ان يبدأ راس القراءة والكتابة بهذا الحرف ومن ثم تحريك الراس لليمين عند قراءة هذا الحرف والابقاء عليه:

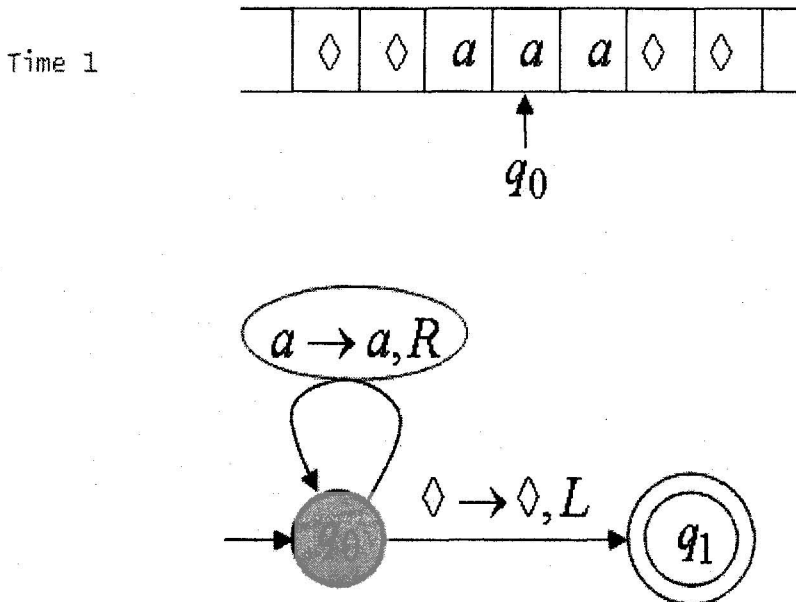


والاشكال التالية تبين تسلسل تنفيذ هذه الالة:

.1

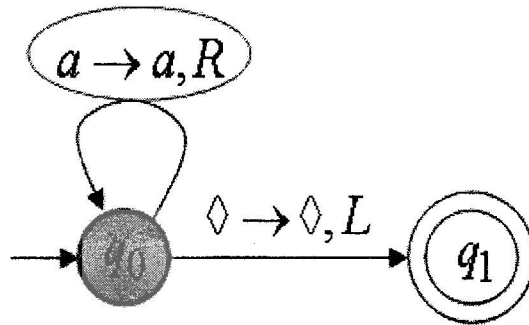
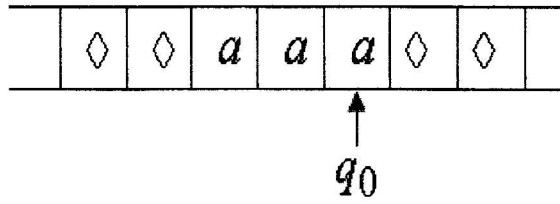


.2



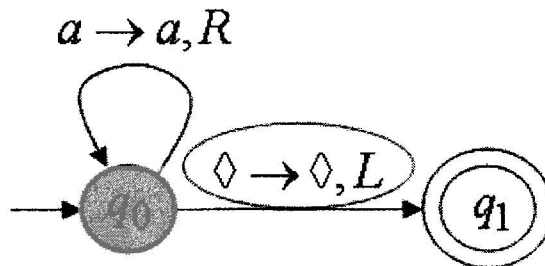
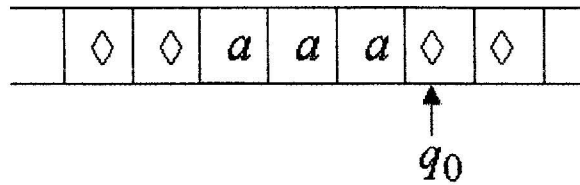
.3

Time 2



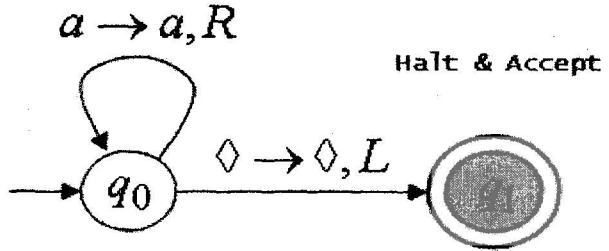
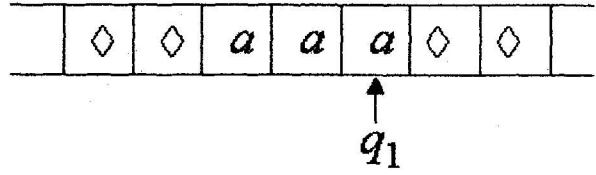
.4

Time 3



.5

Time 4

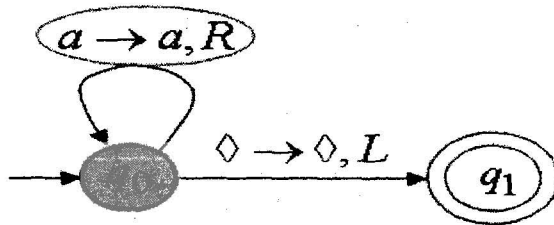
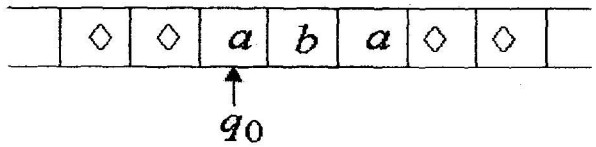


اما عملية رفض السلسلة الرمزية (نفس السلسلة في المثال السابق) فيمكن

بيانها من خلال الأشكال التالية:

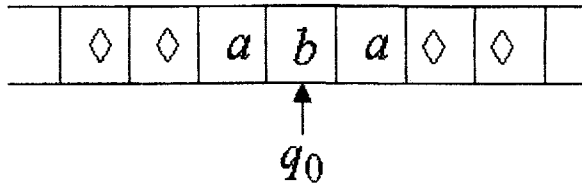
.1

Time 0



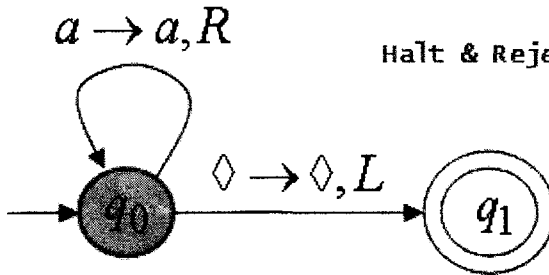
.2

Time 1



No possible Transition

Halt & Reject

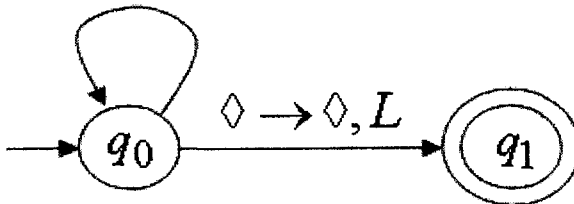


مثال:

فيما يلي آلة مخصصة للتعرف على مجموعة من الأحرف المتتالية لـ  $a$  والتي سوف تنتقل إلى حالة لا نهائية نظرا لعدم توفر هذه الرموز في مجموعة الرموز الموجودة على الشريط:

$b \rightarrow b, L$

$a \rightarrow a, R$

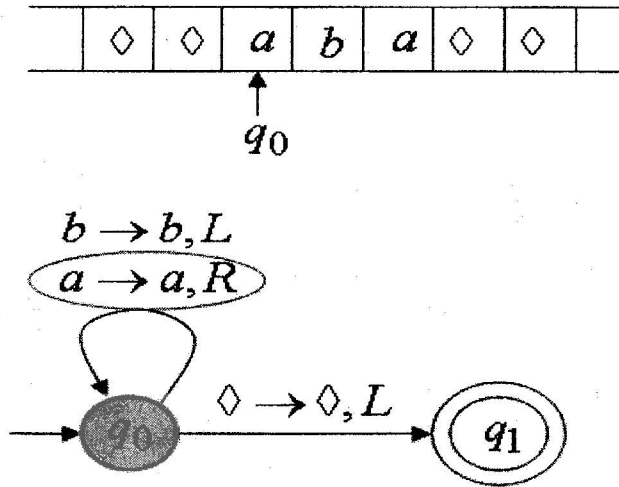




وفيما يلي آلية تنفيذ هذه الآلة:

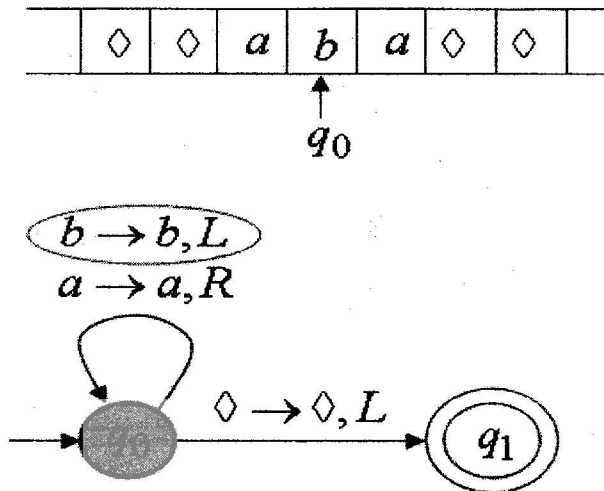
1.

Time 0

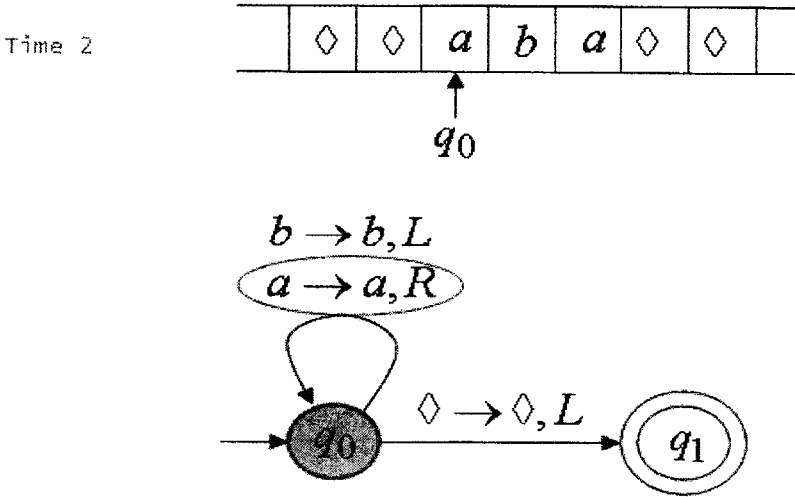


2.

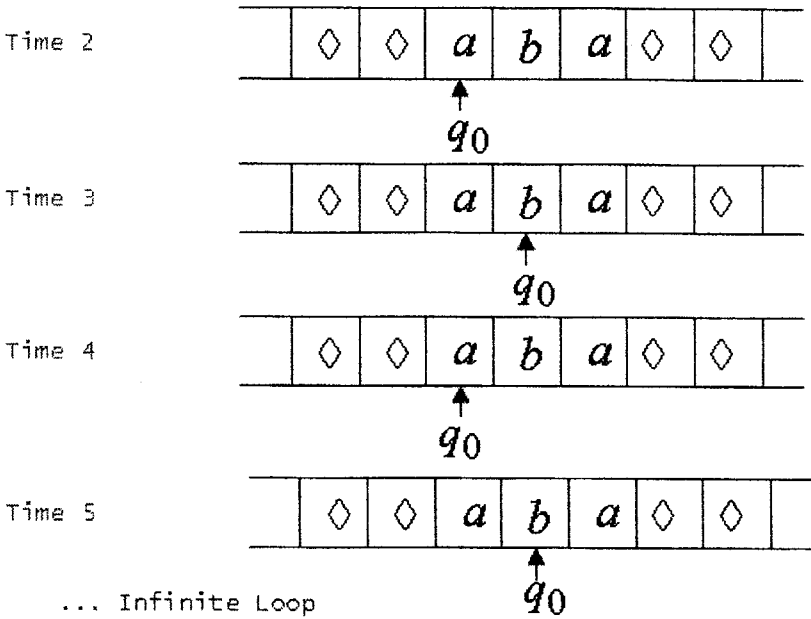
Time 1



.3



.4

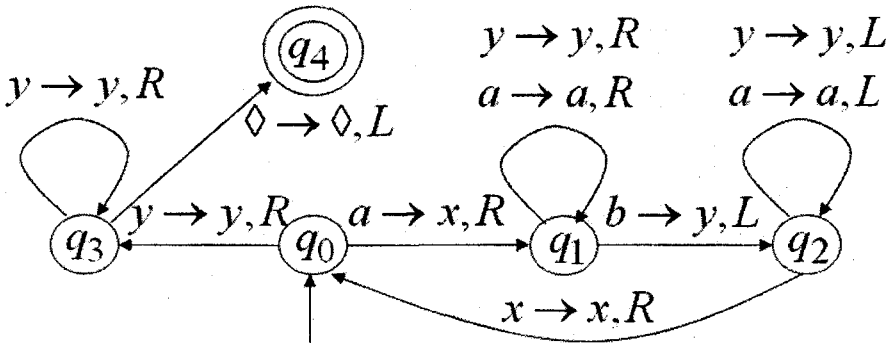


مثال:

صمم آلة تيورينج والتي تفيل التعبير أو اللغة التالية:

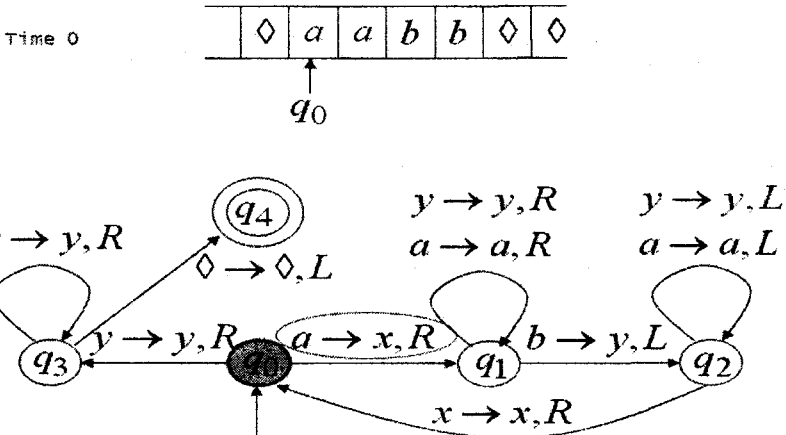
$$\{ a^n b^n \}$$

الحل:

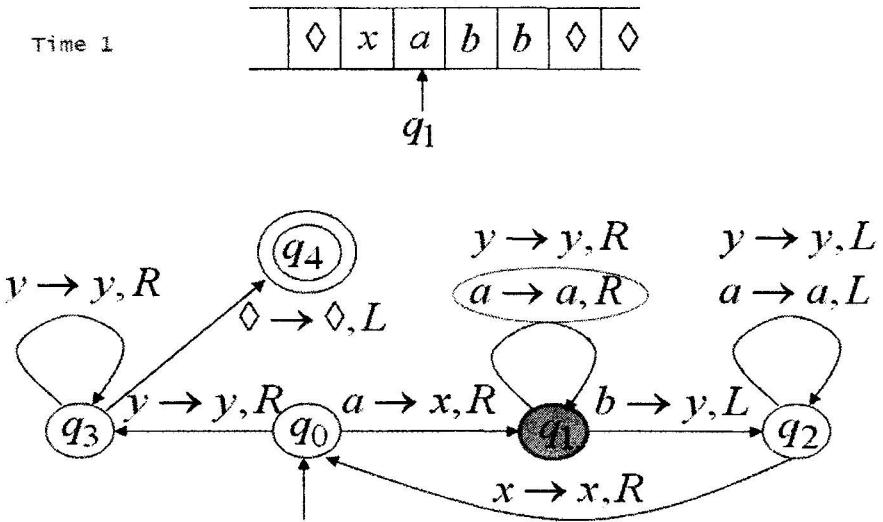


وفيما يلي الآلة تنفيذ هذه الآلة والهيئات التي تمر بها:

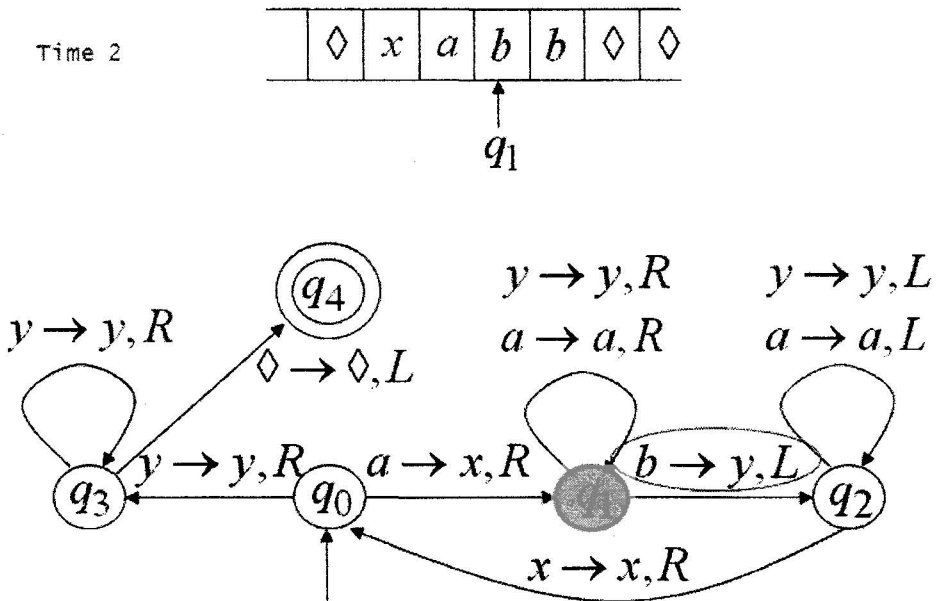
1.



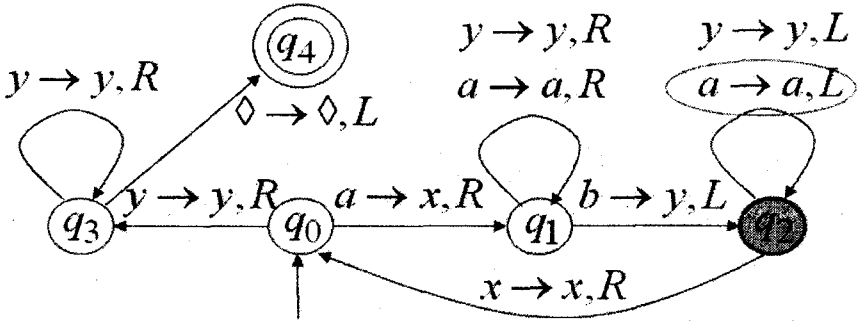
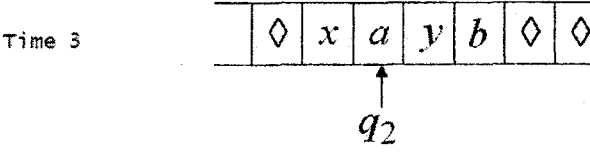
.2



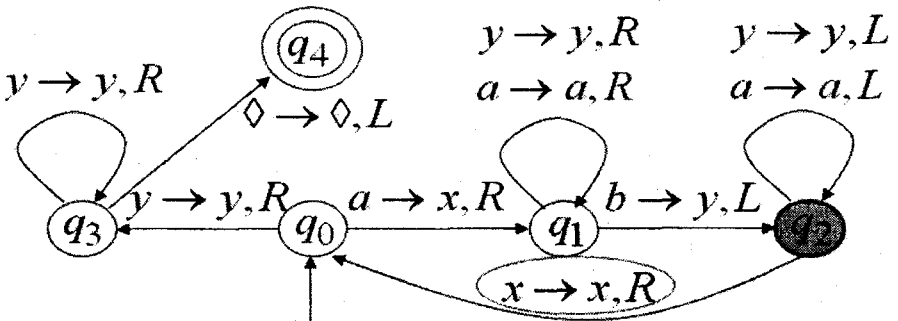
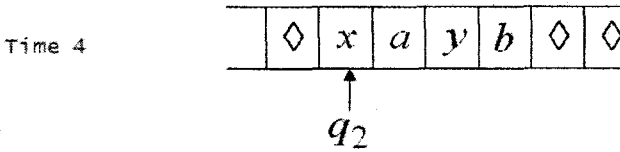
.3



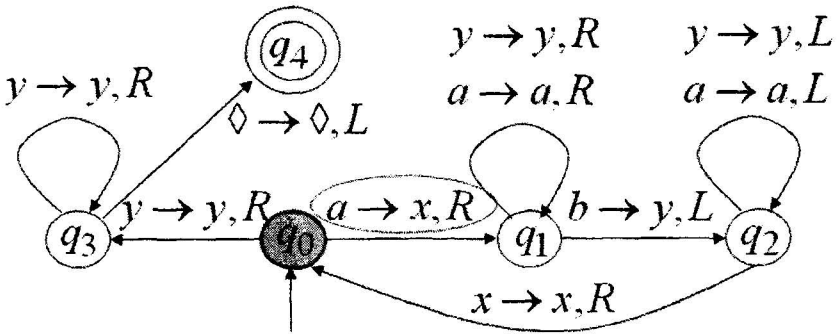
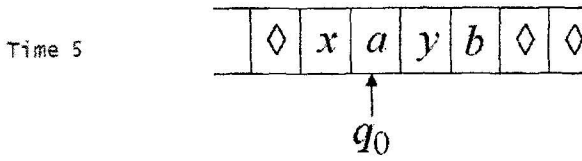
.4



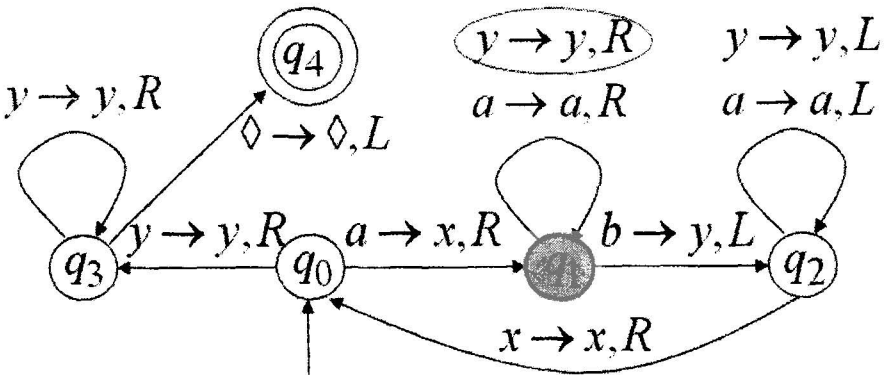
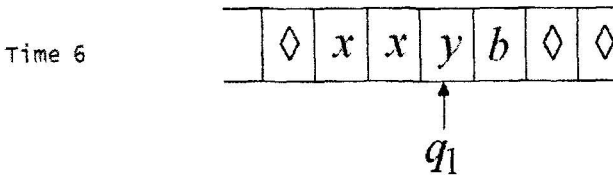
.5

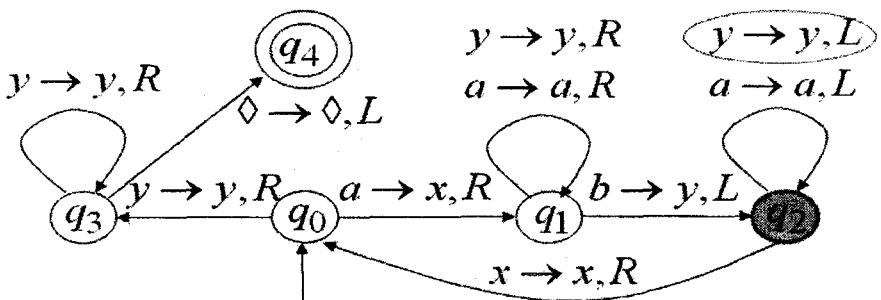
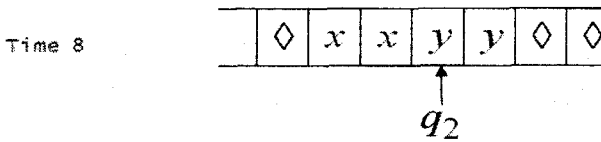
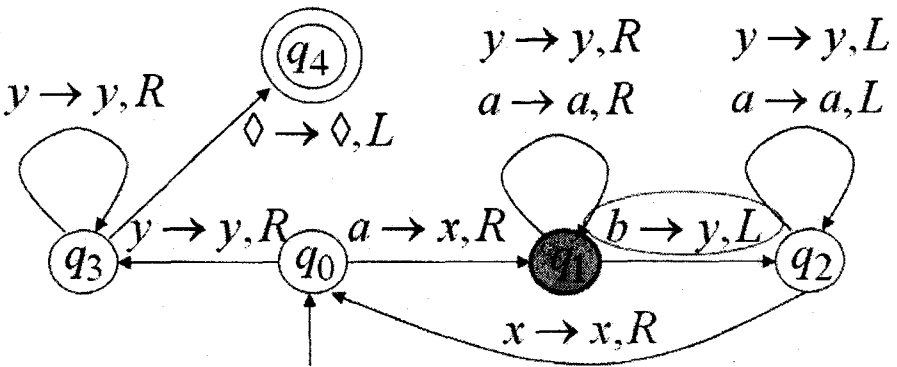
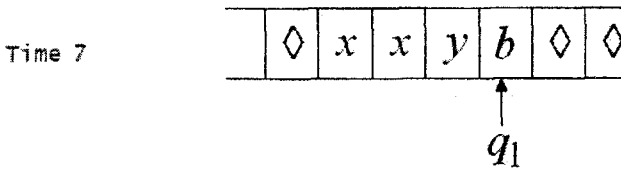


.6



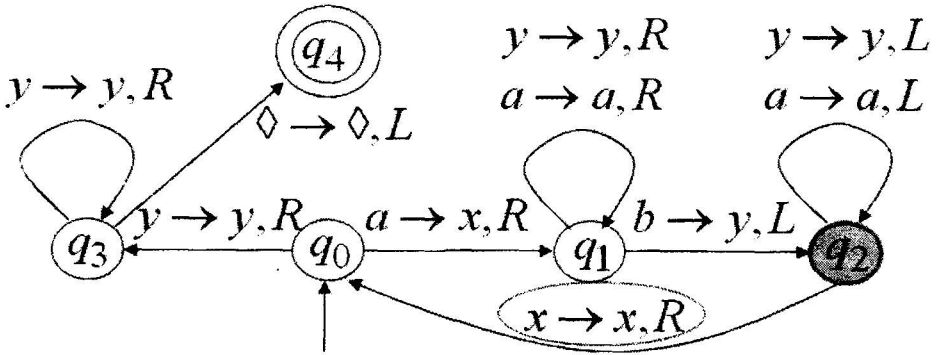
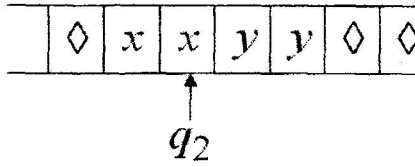
.7





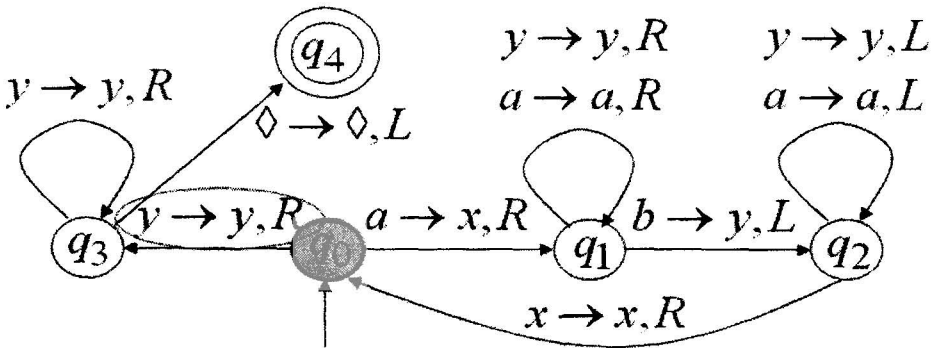
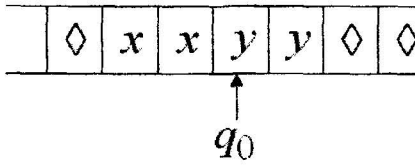
.10

Time 9



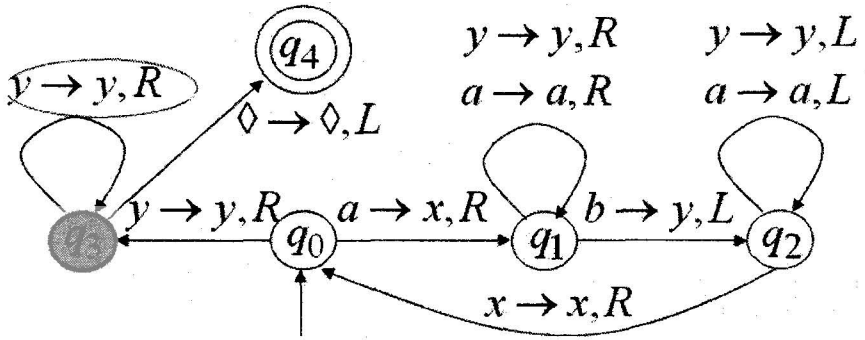
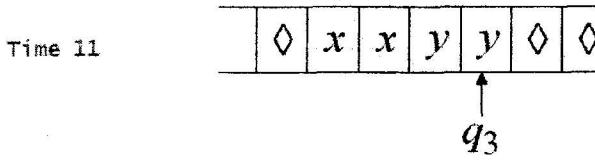
.11

Time 10

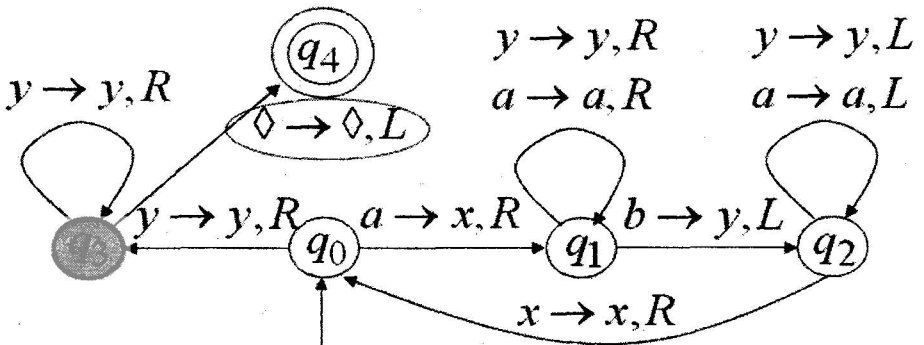
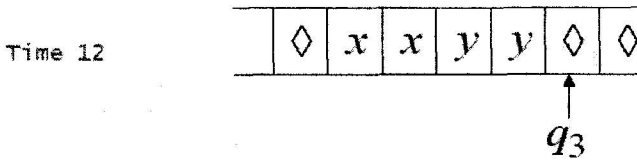




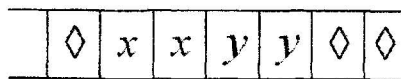
.12



.13

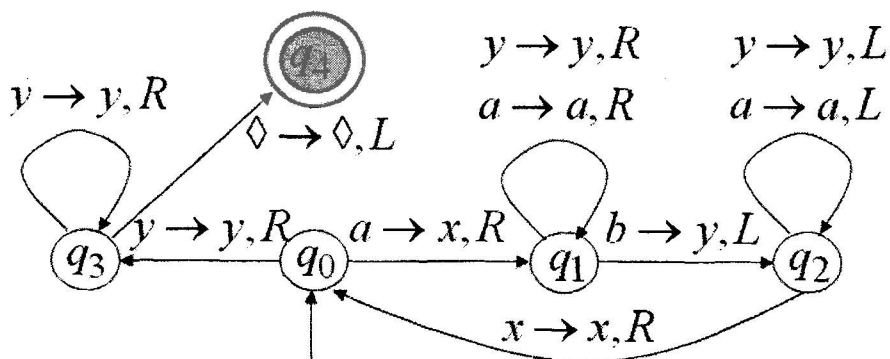


Time 13



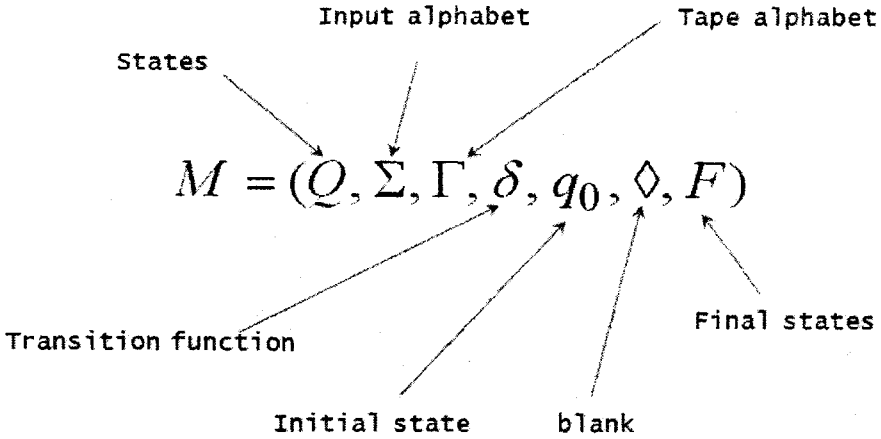
$q_4$

Halt & Accept



2.5 النموذج الرياضي لآلة تيورنج:

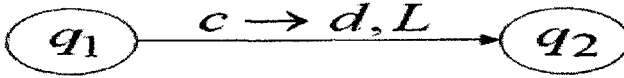
تعرف آلة تيورنج رياضيا كما يلي:



حيث يضم هذا النموذج ما يلي:

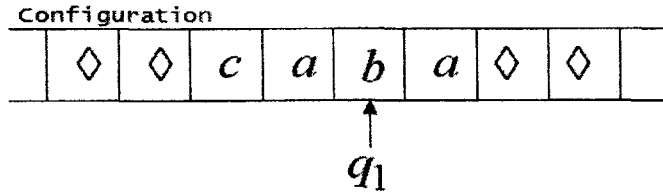
- مجموعة الحالات بما فيها الحالة الابتدائية والحالات النهائية المقبولة والحالات غير المقبولة والحالات المرفوضة.
- مجموعة رموز المدخلات والتي تحتوي على الفراغ ويجب ان لا تحتوي على الحرف R و L لاستخدامها لغاية تحريك رأس القراءة والكتابة لليمين أو اليسار.
- مجموعة رموز الشريط.
- الحالة الابتدائية.
- رمز الفراغ.
- مجموعة الحالات النهائية.

تستخدم دالة الانتقال للتعبير عن الحالة التي يمكن الانتقال اليها من حالة محددة بقراءة رمز للانتقال من الحالة الحالية الى الحالة القادمة مع تنفيذ عملية الكتابة على الشريط وفي نفس الموقع ثم تحريك رأس القراءة والكتابة الى اليمين أو اليسار وكما هو مبين في الشكل التالي:



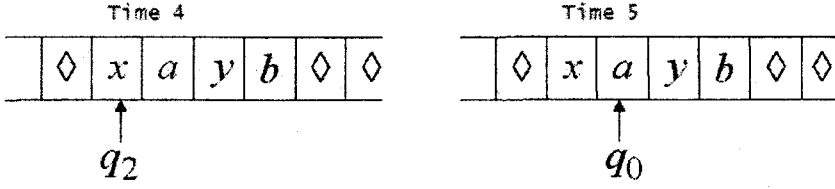
$$\delta(q_1, c) = (q_2, d, L)$$

يمكن وصف الآلة أيضا بالهيئة والتي تبين الحالة الحالية ومجموعة الرموز التي قرأت ومجموعة الرموز المتبقية وكما هو مبين في الشكل التالي:



Instantaneous description:  $ca q_1 ba$

ويمكن استخدام الهيئة لوصف سلوك الآلة كما يلي:



A Move:  $q_2 \ x a y b \succ x \ q_0 \ a y b$

مثال:

For  $\Sigma = \{a,b\}$ , design a Turing Machine that accepts  $L = \{a^n b^n : n \geq 1\}$

الحل:

$Q = \{q_0, q_1, q_2, q_3, q_4\}$ ,

$\Sigma = \{a,b\}$

$\Gamma = \{a,b,x,y,B\}$ ,

$q_4$ : accept state

مدخلات ونواتج تنفيذ الآلة:

aaabbb

xaaybb

فيما يلي دالة الإنتقال وهيئات الآلة:

$$\delta(q_0, a) = (q_1, x, R) ; \quad \delta(q_1, a) = (q_1, a, R)$$

$$\delta(q_1, y) = (q_1, y, R) ; \quad \delta(q_1, b) = (q_2, y, L)$$

$$\delta(q_2, y) = (q_2, y, L) ; \quad \delta(q_2, a) = (q_2, a, L)$$

$$\delta(q_2, x) = (q_0, x, R)$$

$$\delta(q_0, y) = (q_3, y, R) ; \quad \delta(q_3, y) = (q_3, y, R)$$

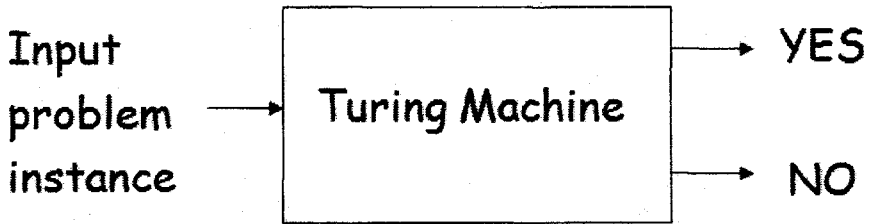
$$\delta(q_3, B) = (q_4, B, R)$$

$$\begin{aligned} q_0 a a b b \vdash x q_1 a b b \vdash x a q_1 b b \vdash x q_2 a y b \vdash q_2 x a y b \\ \vdash x q_0 a y b \vdash x x q_1 y b \vdash x x y q_1 b \vdash x x q_2 y y \\ \vdash x q_2 x y y \vdash x x q_0 y y \vdash x x y q_3 y \vdash x x y y q_3 B \\ \vdash x x y y B q_4 \end{aligned}$$

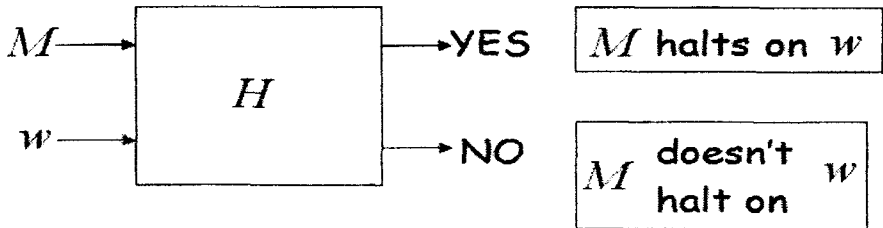
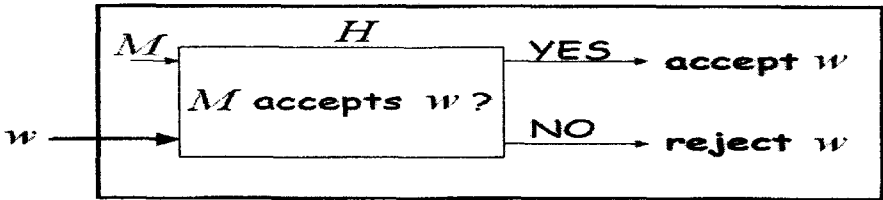
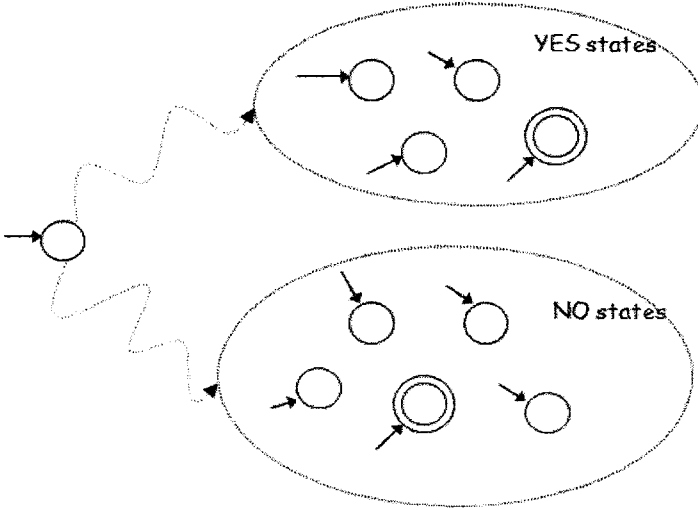
## 3.5 تطبيقات آلة تيورنج:

تستخدم آلة تيورنج في الكثير من التطبيقات ولكننا هنا سنستعرض بعضا من هذه التطبيقات وخاصة حساب الاقتران وتمثيل بعض العمليات في معالج النصوص مثل عملية النسخ والتحرك والازاحة.

كما اشرنا سابقا فان آلة تيورنج تعمل على استعراض لغة أو تعبير منظم وتقرر قبول أو رفض هذه اللغة بناء على التصميم المحدد لهذه الآلة وبناء على الحالة النهائية التي تصل اليها الآلة بعد معالجة الرموز فإذا كانت الحالة منتهية فان مجموعة الرموز مقبولة إما اذا وقعت الآلة في حالة غير منتهية فان مجموعة هذه الرموز تكون غير مقبولة أو مرفوضة وبمعنى آخر يمكن تصور آلة تيورنج بمسارين مسار القبول ومسار الرفض للتعبير أو اللغة وكما هو مبين في الشكل التالي:



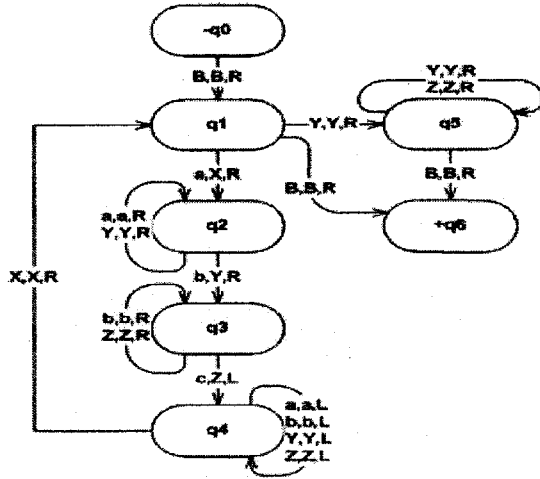
فاذا كان ناتج المسار نعم فان الآلة تتوقف في حالة نهائية إما اذا كان ناتج مسار التنفيذ لا فا الآلة ان تتوقف ولن تصل الى حالة نهائية وكما هو مبين في الشكل التالي:



إذا توقفت آلة تيورينج في حالة نهائية فإن الآلة تكون محسوبة إما إذا سلكت الآلة سلوك رفض الرموز فإنه لن تصل إلى حالة نهائية ولن تتوقف وتكون الآلة في هذه الحالة غير محسوبة.



لناخذ آلة تيورينج التالية:



لنتبع هذه الآلة باستخدام اللغة المؤلفة من مجموعة الرموز التالية

أولنتأكد من إمكانية تمييز أو قبول هذه الرموز من قبل هذه الآلة:  $aabbcc$

فيما يلي نتيجة تتبع هذه الآلة خطوة خطوة:

- BaabbccB
- BaaabbccB
- BXabbccB
- BXabbccB
- BXaYbccB
- BXaYbccB
- BXaYbZcB
- BXaYbZCB
- BXaYbZcB
- BXaYbZcB
- BXaYbZcB
- BXaYbZcB
- BXXYbZcB
- BXXYbZcB

BXXYYZcB  
 BXXYYZcB  
 BXXYYZZB  
 BXXYYZZB  
 BXXYYZZB  
 BXXYYZZB  
 BXXYYZZB  
 BXXYYZZB  
 BXXYYZZB  
 BXXYYZZB  
 BXXYYZZB  
 BXXYYZZB  
 BXXYYZZBB (halt)

بما ان الآلة وصلت الى حالة توقف نهائية فان هذه يعني ان مجموعة الرموز قد قبلت وتم التعرف عليها من قبل هذه الآلة.

والان لنستعرض استخدام الآلة تيورنج في معالجة الاقترانات.

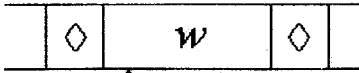
سوف نستخدم في الاقترانات القيم الاحادية بدلا من استخدام القيم العشرية أو الثنائية :

Decimal: 5  
 Binary: 101  
 Unary: 11111

يعتبر الاقتران محسوبا اذا توفرت الآلة تيورنج واستطاعت حساب هذا الاقتران أو بمعنى اخر حققت الشروط المبينة في الشكل التالي:

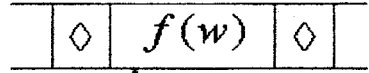
A function  $f$  is computable if there is a Turing Machine  $M$  such that:

Initial configuration



$q_0$  initial state

Final configuration

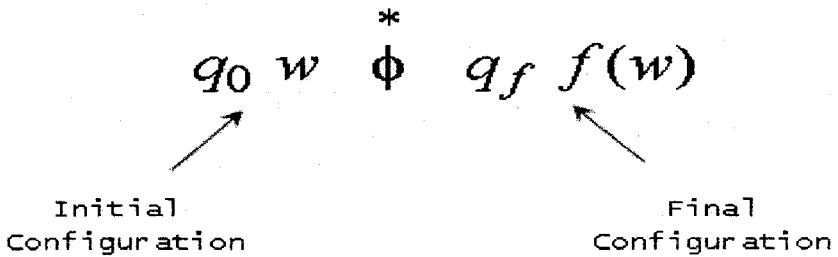


$q_f$  final state

For all  $w \in D$  Domain

أو

A function  $f$  is computable if there is a Turing Machine  $M$  such that:



For all  $w \in D$  Domain

والآن لناخذ الإقتران التالي:

The function  $f(x, y) = x + y$  is computable

$x, y$  are integers

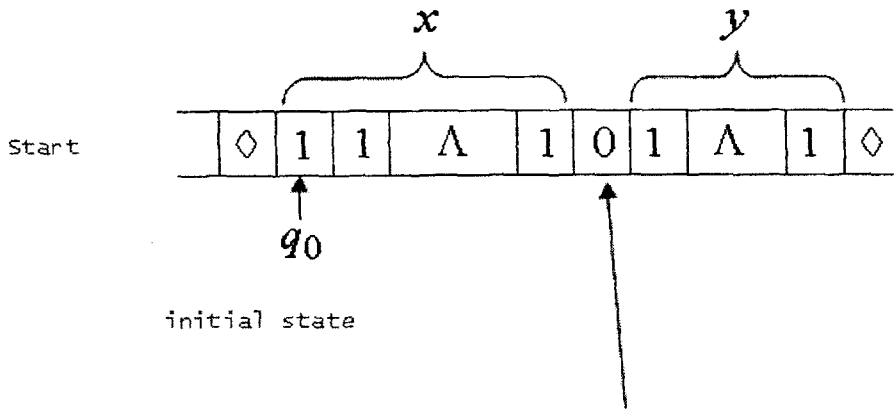
Turing Machine:

Input string:  $x0y$  unary

Output string:  $xy0$  unary

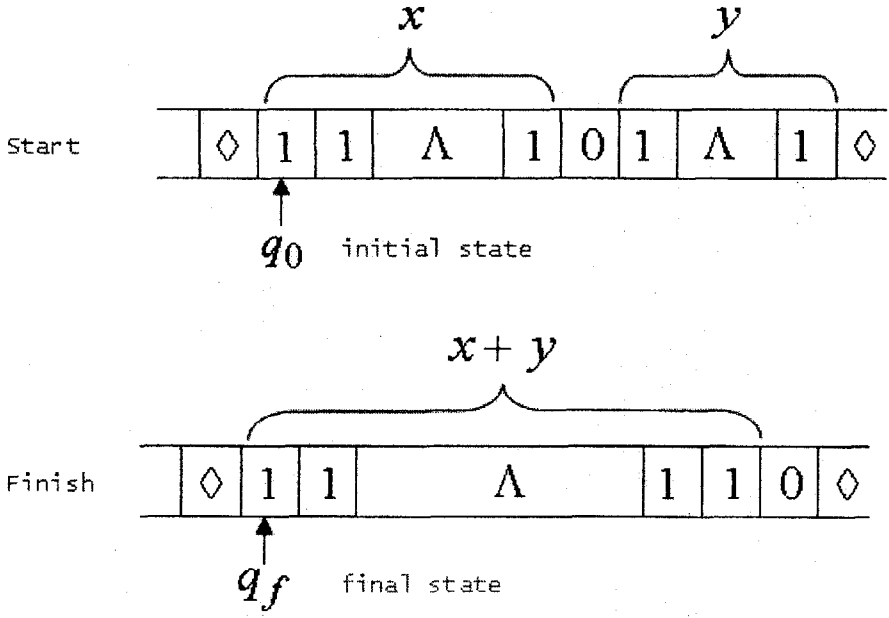
لنبدء بتمثيل الاقتران باستخدام آلة تيورينج:

.1



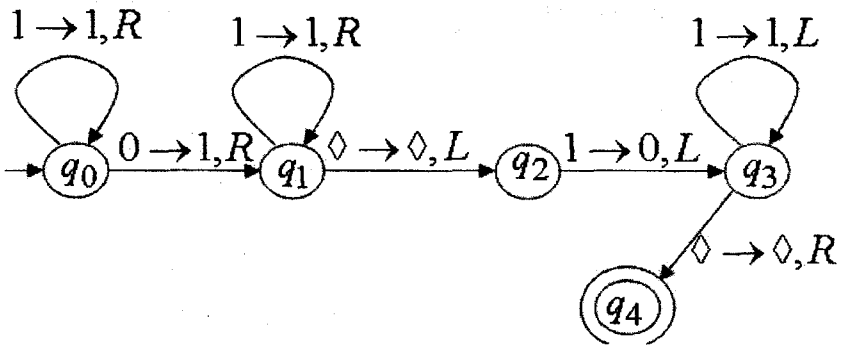
The 0 is the delimiter that separates the two numbers

.2



وفيما يلي آلة تيورينج للتعرف على الإقتران السابق:

$$f(x, y) = x + y$$



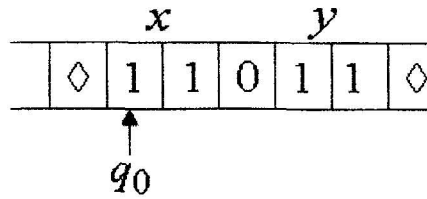
وفيما يلي ناتج تنفيذ هذه الآلة:

Execution Example:

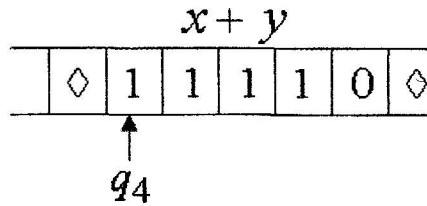
Time 0

$$x = 11 \quad (2)$$

$$y = 11 \quad (2)$$

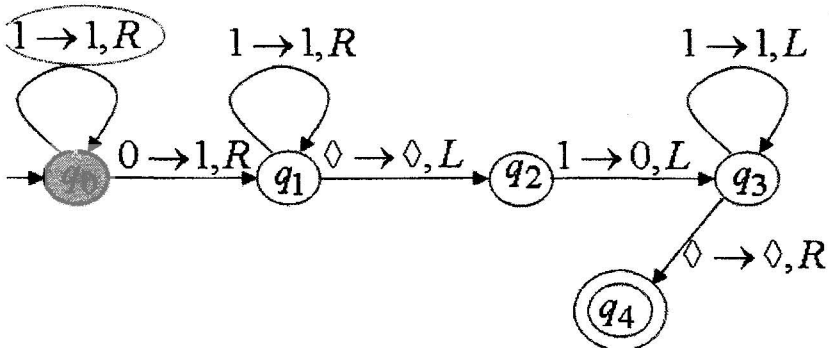
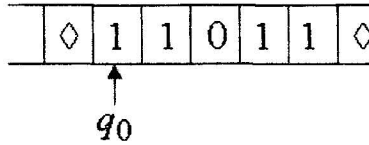


Final Result

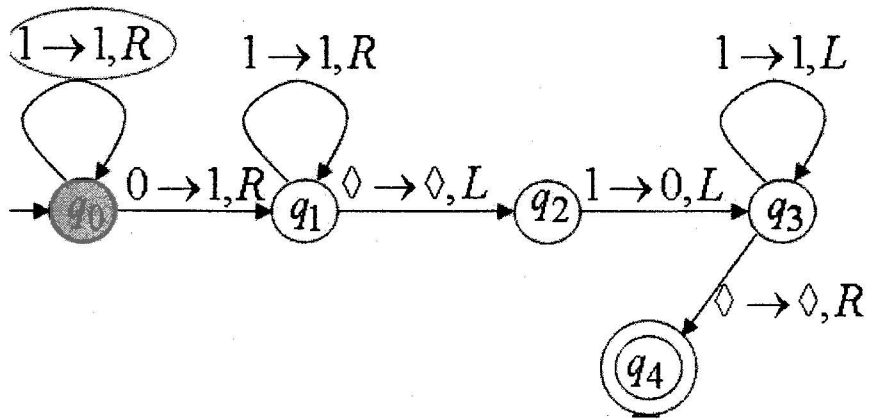
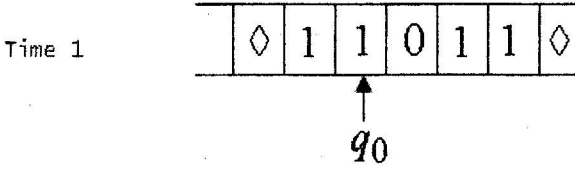


.1

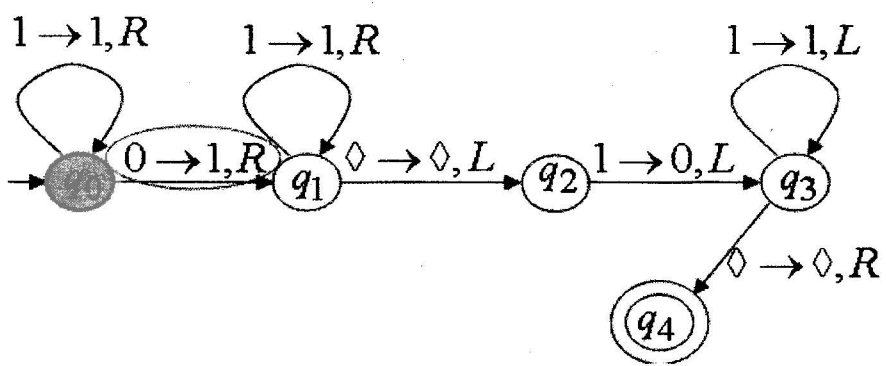
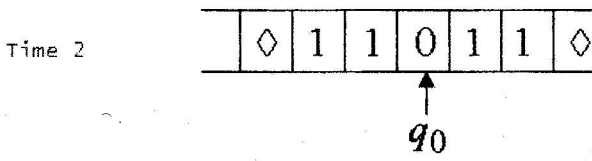
Time 0



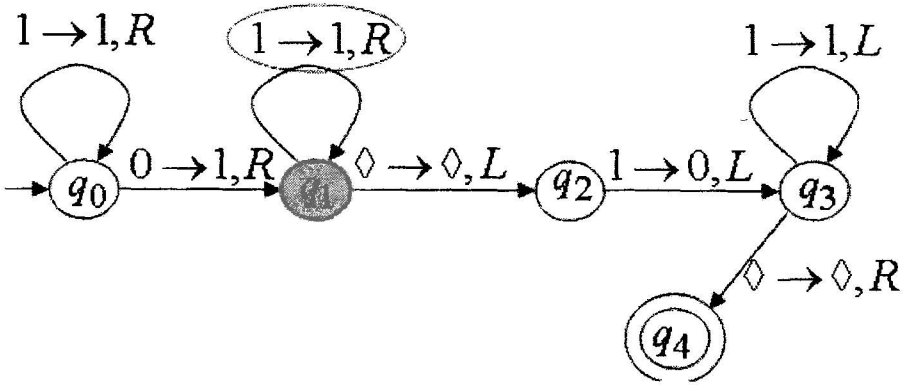
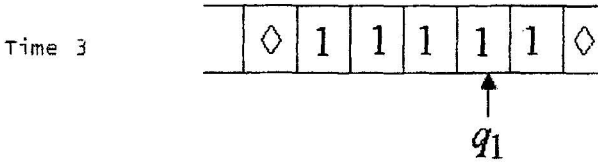
.2



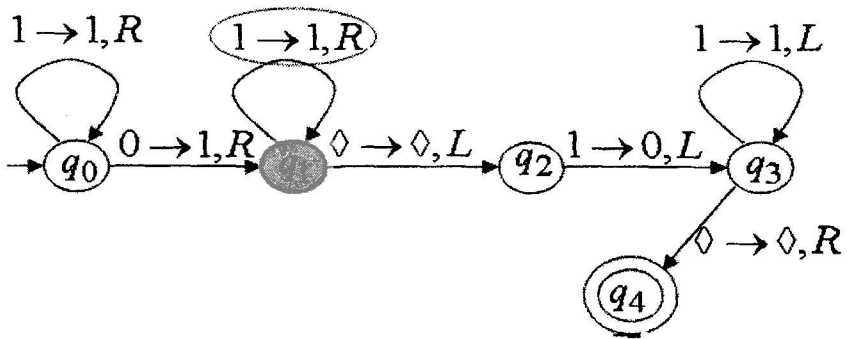
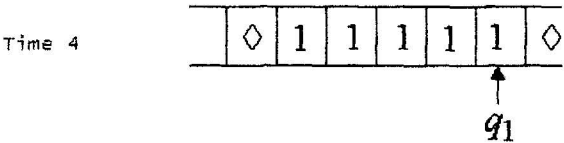
.3



.4



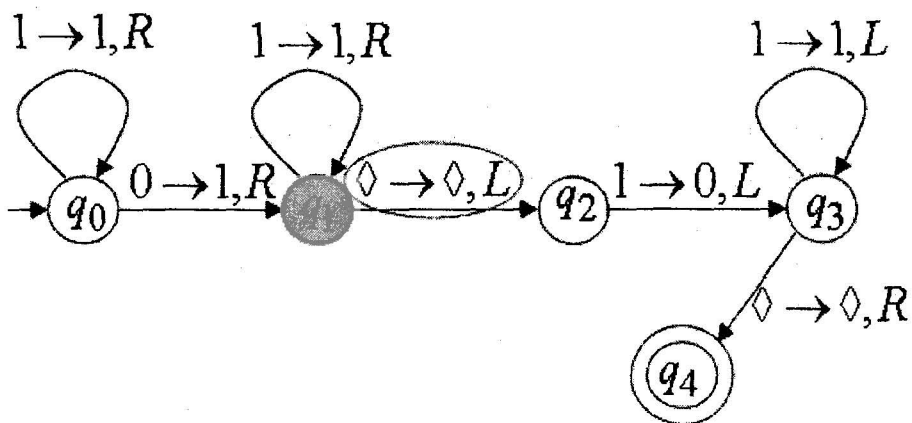
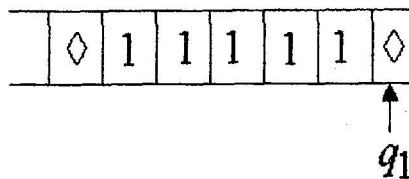
.5





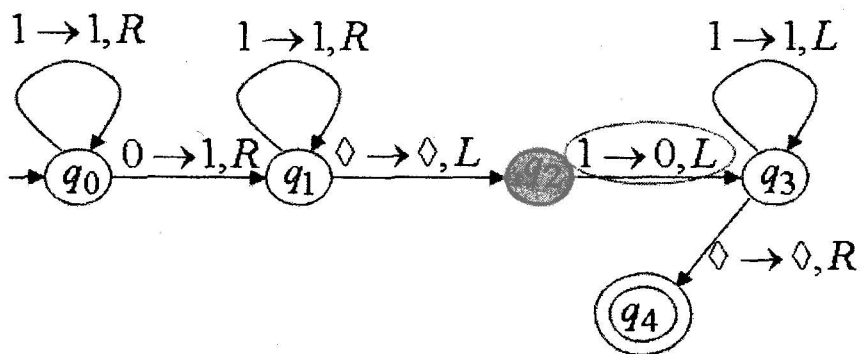
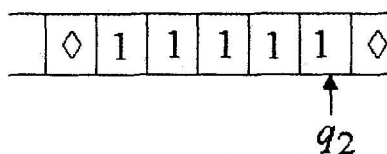
.6

Time 5

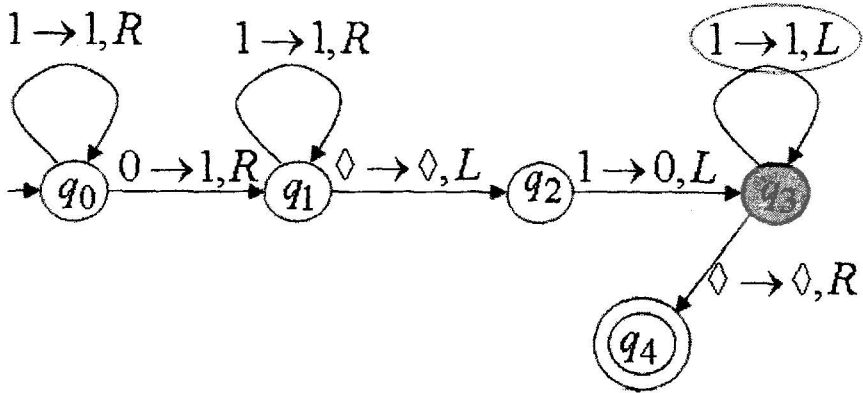
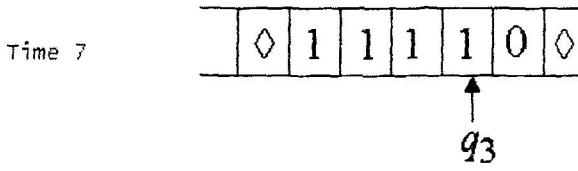


.7

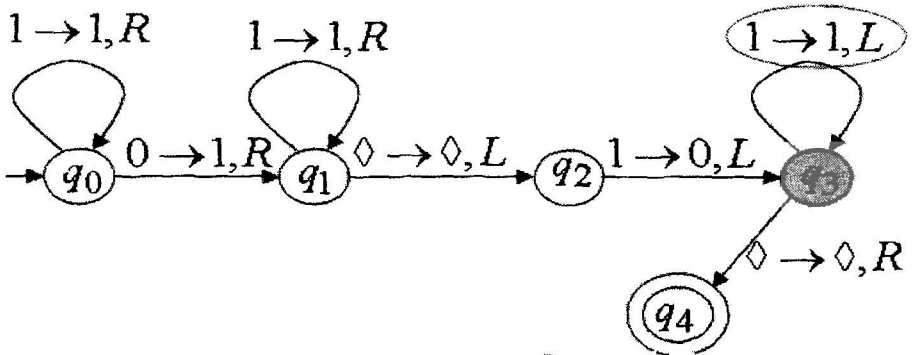
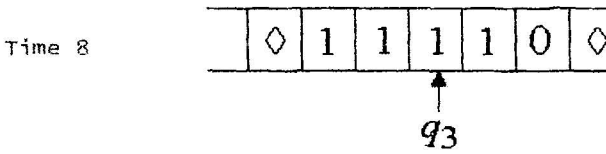
Time 6



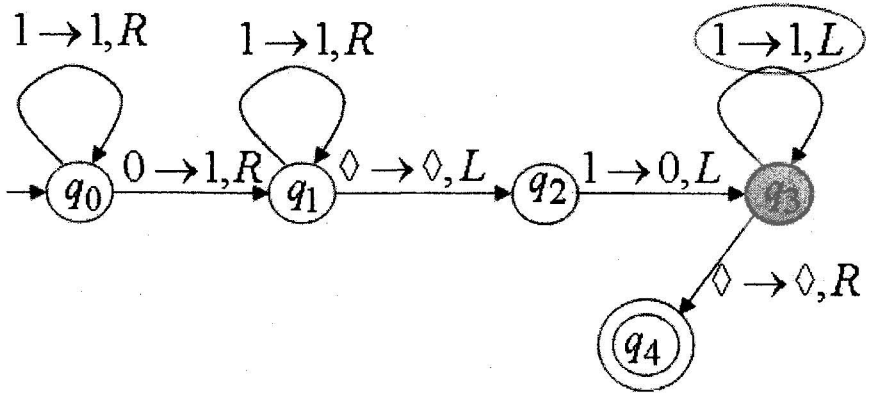
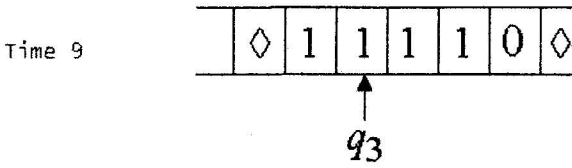
.8



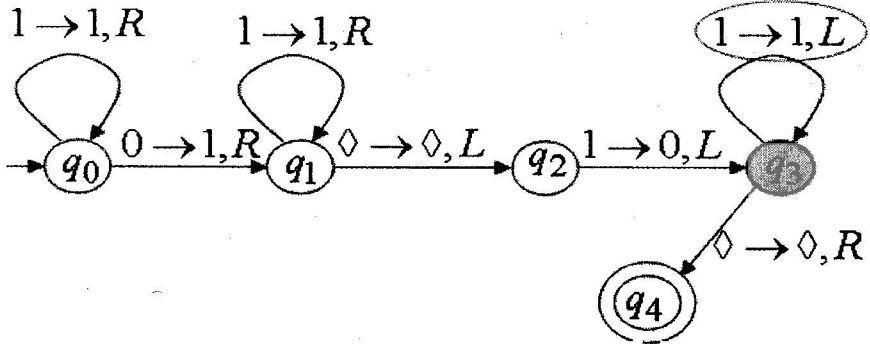
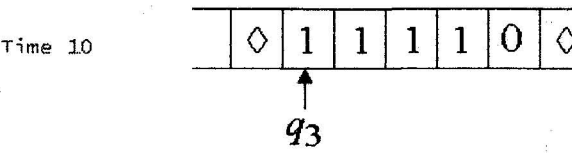
.9



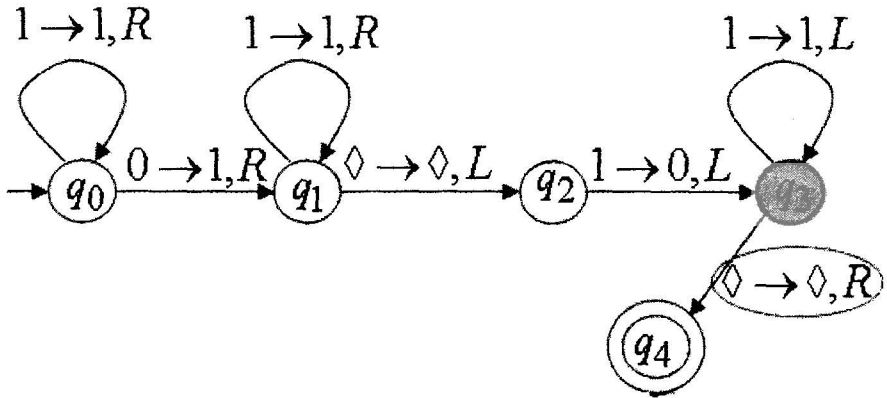
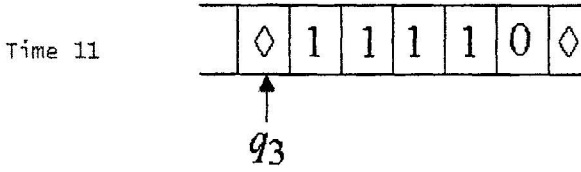
.10



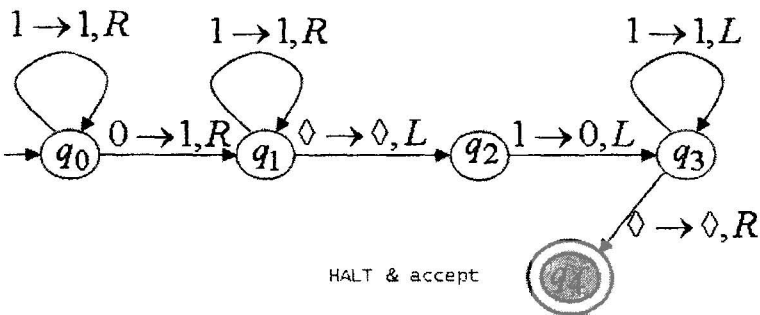
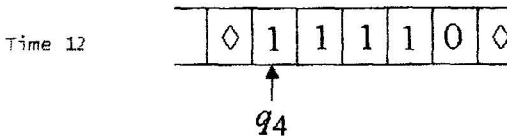
.11



.12



.13



مثال:

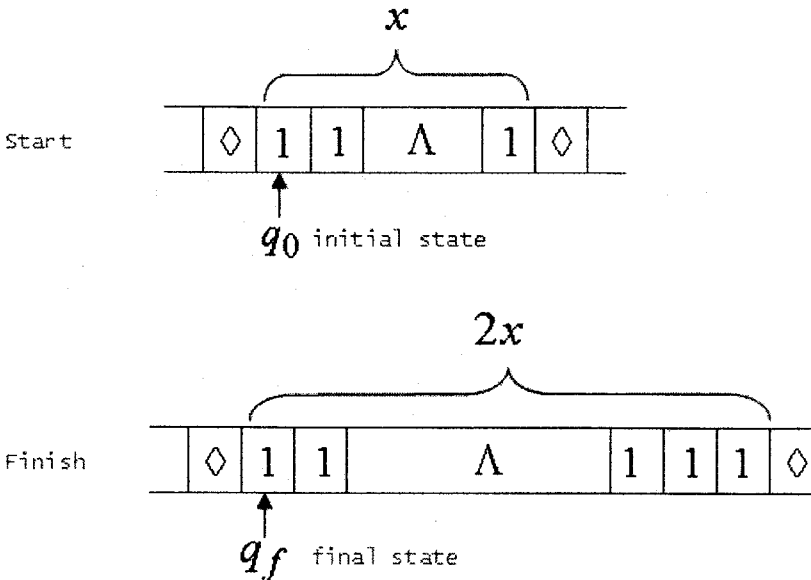
The function  $f(x) = 2x$  is computable

$x$  is integer

Turing Machine:

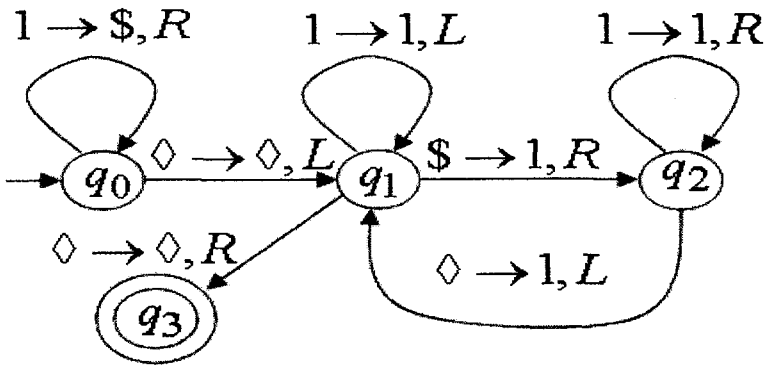
Input string:  $x$  unary

Output string:  $xx$  unary

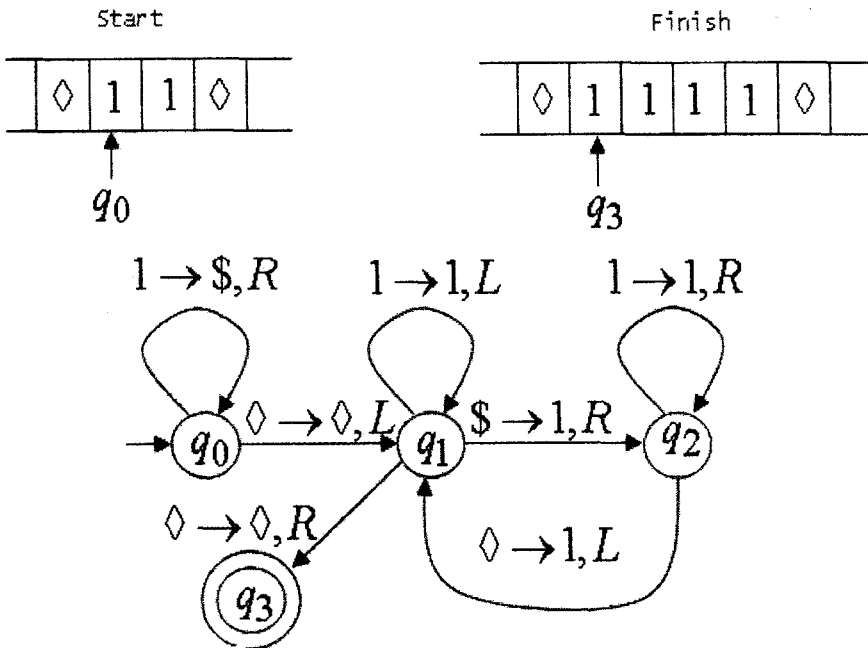


وفيما يلي آلة تيورينج للتعرف على هذا الإقتران:

Turing Machine for  $f(x) = 2x$



وفيما يلي آلية تنفيذ هذه الآلة:



سؤال:

ابن آلة تيورينج للتعرف على الاقتران التالي:

The function

$$f(x, y) = \begin{cases} 1 & \text{if } x > y \\ 0 & \text{if } x \leq y \end{cases}$$

is computable

Turing Machine for  $f(x, y) = \begin{cases} 1 & \text{if } x > y \\ 0 & \text{if } x \leq y \end{cases}$

Input:  $x0y$ Output:  $1$  or  $0$ 

تستخدم آلة تيورينج ايضا في تطبيقات علم الحاسوب وخاصة في تمثيل

العمليات المنفذة من قبل معالجات النصوص ومن هذه العمليات:

- نسخ الرموز أو السلسلة الرمزية.
- نقل و تحريك النصوص.

- استبدال الرموز برموز اخرى.
- إزاحة الرموز لليسار.
- إزاحة الرموز لليمين.
- نفل رأس القراءة والكتابة الى رمز معين أو عملية البحث عن أول حرف أو رمز.
- تكرار اي من العمليات السابقة عدد محدد من المرات.

وفيما يلي نستعرض بعض الامثلة والتي تبين كيفية استخدام آلة تيورينج لتمثيل بعض العمليات المنفذة من قبل برمجيات معالجات النصوص أو ما يسمى ببرامج تحرير النص.

مثال:

صمم آلة تيورينج والتي تعالج مجموعة من الرموز مؤلفة من حرفين بحيث تستبدل الحرف الاول بالثاني والحرف الثاني بالاول.

نستعرض الية بناء هذه الآلة من خلال تسلسل التنفيذ فيها والذي سيكون

كما يلي:

BbabB (character about to be read)

BbabB

BaabB

BabbbB

BabaB

BabaB

BabaB

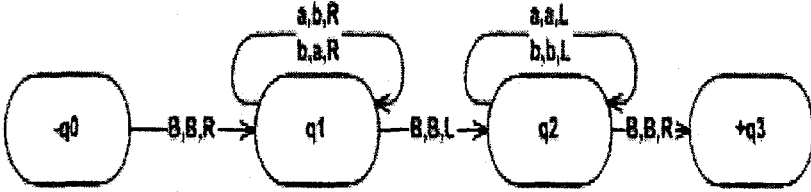
BabaB

BabaB

BabaB (halts pointing at 1<sup>st</sup> output character)



وفيما آلة تيورينج لتنفيذ هذه العملية:



مثال:

### عملية النسخ

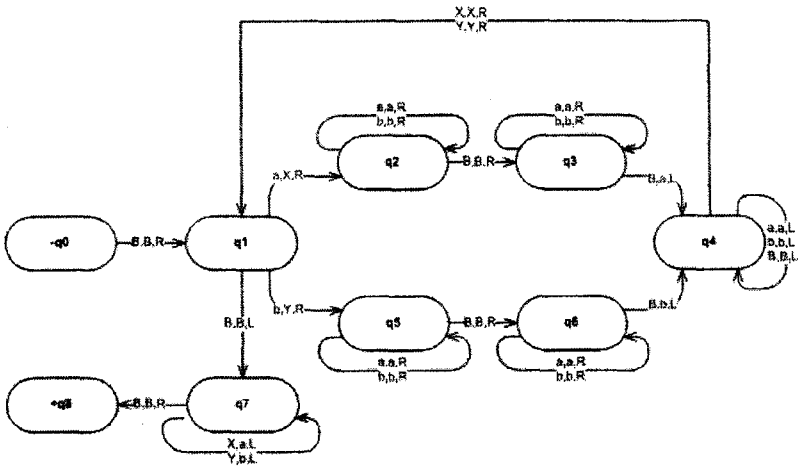
تتلخص عملية نسخ مجموعة من الرموز في الخطوات التالية:

1. نقل رأس القراءة والكتابة الى أول فراغ في اليسار.
2. التحرك لليمين لخطوة واحدة.
3. كتابة فراغ أو أي رمز بدل الحرف والاحتفاظ به.
4. البحث عن ثاني فراغ باتجاه اليمين.
5. كتابة الرمز في الموقع.
6. البحث عن ثاني فراغ باتجاه اليسار.
7. اعادة كتابة الرمز بدل الفراغ.
8. تكرار الخطوات 2 الى 7 ولكل رمز أو حرف.

فلو أخذنا حرفين وكما هو مبين ادناه فان الية التنفيذ ستتم كما يلي:

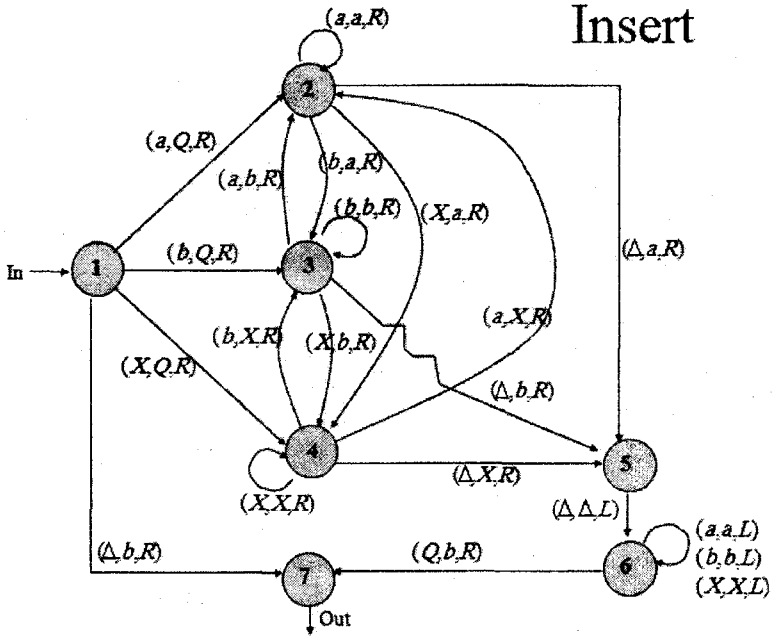
<u>B</u> abB	BXYBa
B <u>a</u> bB	BXYBa <u>B</u>
BX <u>b</u> B	BXYB <u>a</u> b
BXb <u>B</u>	BXY <u>B</u> ab
BXbBB	BXY <u>B</u> ab
BXb <u>B</u> a	BXY <u>B</u> ab
BX <u>b</u> Ba	BXY <u>B</u> ab
BXbBa	BX <u>b</u> Bab
BX <u>b</u> Ba	<u>B</u> abBab (halt)
BXY <u>B</u> a	

اما الة تيورينج والمنفذة لهذه العمليات فهي كما يلي:



مثال:

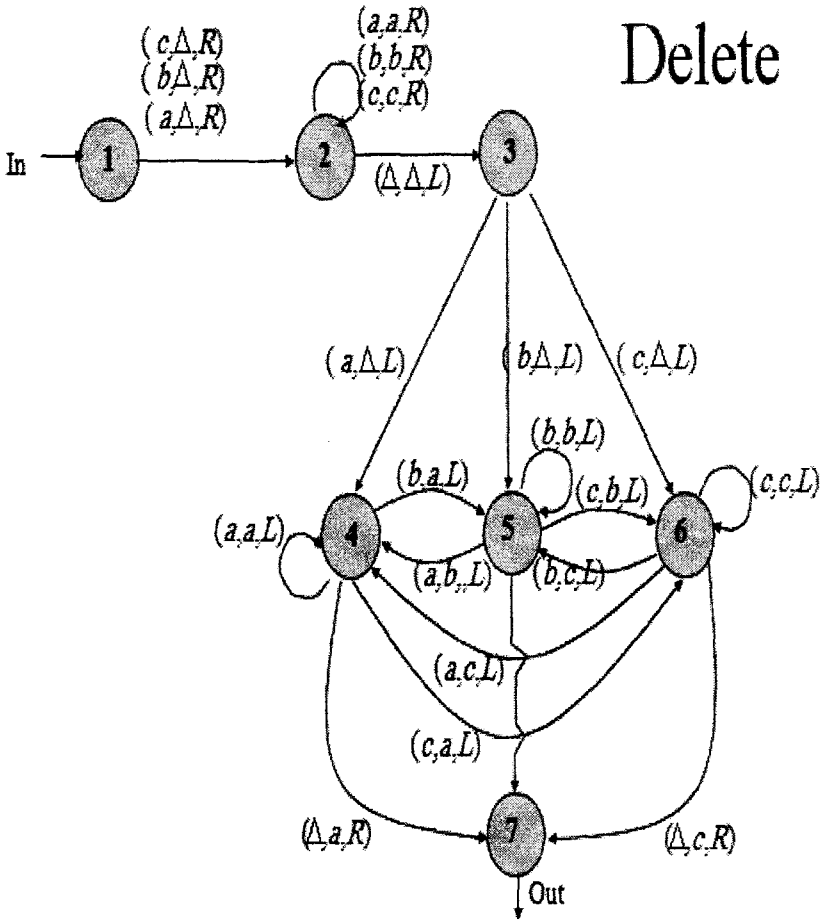
آلة الإدخال



مثال:

آلة الحذف

# Delete



الوحدة السادسة

آلة مور و آلة ميلي

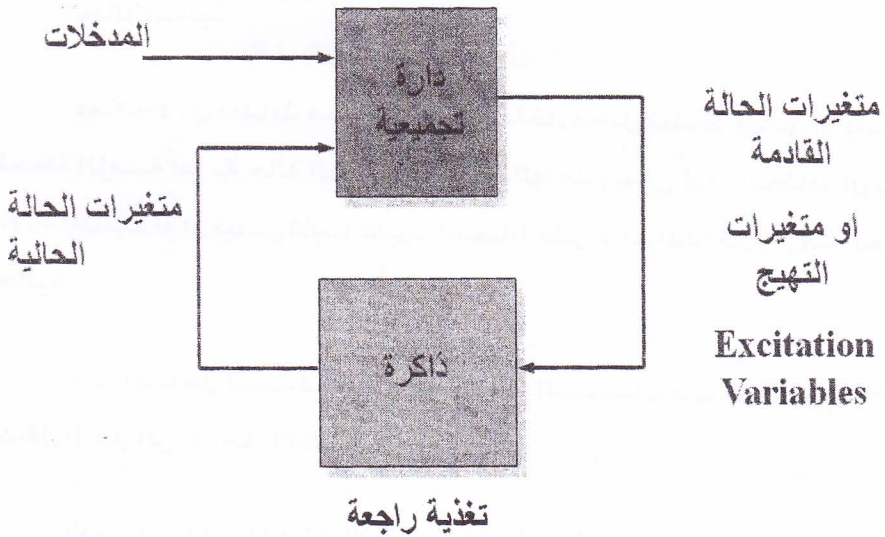
*Moore & Mealy  
Machine*

6

1.6 مقدمة:

تستخدم الة الحالة المنتهية كما اشرنا سابقا في هذا الكتاب في تطبيقات متعددة ومن هذه التطبيقات تصميم الدارات المنطقية التتابعية والمتزامنة.

تمثل الدارة المنطقية التتابعية المتزامنة الة حالة منتهية وتتكون هذه الة في الغالب من دارة منطقية تجميعية مؤلفة من البوابات المنطقية المختلفة ومن ذاكرة للاحتفاظ بحالة الدارة الحالية ويبين الشكل التالي معمار الة الحالة المنتهية والتي يمكن استخدامها في عملية التصميم المنطقي لدارات المنطق التتابعية:



يتم بناء الذاكرة في الة الحالة المنتهية باستخدام أحد انواع النطاطات Flip-Flops ويتوفر من هذه النطاطات اربعة انواع هي:

- النطاط RS
- النطاط JK
- النطاط D
- النطاط T

ولإستخدام أي من هذه النشاطات في تصميم الآلة الحالة المنتهية لا بد من معرفة بعض الامور الاساسية عن كل نشاط ونخص بالذكر:

- مجموعة المداخل والمخارج.
- جدول الصواب والذي يبين كيفية انتقال النشاط من الحالة الحالية الى الحالة التالية بوجود نبضة الساعة وبعض القيم على المداخل.
- المعادلة المميزة والتي تبين الحالة التالية كإقتران يعتمد على قيم المداخل وقيمة الحالة الحالية.
- جدول التهيج والذي يستخدم في عملية تصميم الآلة المنتهية والذي يبين القيم الواجب توفرها على المداخل لنقل النشاط من الحالة الحالية الى الحالة التالية.

يملك أي نشاط مجموعة من المداخل وكل نشاط يمكن ان يقع في اللحظة الزمنية اما في حالة الصفر أو في حالة الواحد ويمكن لهذا النشاط ان يغير حالته التالية أو ان يبقى ثابتا عليها اعتمادا على قيم المداخل وقيمة الحالة الحالية.

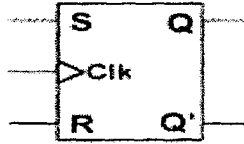
من المداخل المستخدمة في كافة انواع النشاطات مدخل المسح Clear ومدخل النقل الى الواحد Preset

فوجود 1 على المدخل الاول وصفر على المدخل الثاني يؤدي الى نقل النشاط الى الحالة صفر بغض النظر عن القيم الاخرى على المداخل اما وجود الصفر على المدخل الاول وواحد على المدخل الثاني فان هذا يؤدي الى نقل النشاط الى الحالة 1 بغض النظر عن القيم الموجودة على المداخل الاخرى، ولإستخدام النشاط في عملية التصميم وخاصة في تصميم الآلة الحالة المنتهية تستخدم الحالة عندما تكون قيمة المدخل الاول مساوية لقيمة المدخل الثاني ومساوية للصفر وفي

هذه الحالة فان الحالة التالية تتاثر بقيمة الحالة الحالية وقيم المداخل الاخرى بوجود نبضة الساعة.

### النطاق SR

يبين الشكل التالي المخطط الصندوقي لهذا النطاق:



ينتقل هذا النطاق من الحالة الحالية الى الحالة المقبلة أو التالية بوجود قيم المداخل التالية والمبينة في جدول الصواب التالي لهذا النطاق:

S	R	Q(next)
0	0	Q
0	1	0
1	0	1
1	1	?

المعادلة المميزة والتي تربط الحالة التالية بقيم المداخل والحالة الحالية هي:

$$Q (next) = S + R'Q$$

$$SR = 0$$

جدول التهيج والذي يحدد القيم الواجب توفرها على المداخل لنقل

النطاق من الحالة الحالية الى حالة تالية معروفة هو:

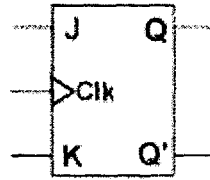


$Q$	$Q(next)$	$S$	$R$
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

### النظام JK

يشبه هذا النظام النظام السابق الا انه يقبل القيم 1 علة الداخل وفي هذه الحالة يعمل النظام على عكس الحالة الحالية في اللحظة الزمنية التالية.

يبين الشكل التالي المخطط الصندوقي لهذا النظام



ينتقل هذا النظام من الحالة الحالية الى الحالة المقبلة أوالتالية بوجود قيم الداخل التالية والمبينة في جدول الصواب التالي لهذا النظام:

$J$	$K$	$Q(next)$
0	0	$Q$
0	1	0
1	0	1
1	1	$Q'$

المعادلة المميزة والتي تربط الحالة التالية بقيم المدخل والحالة الحالية هي:

$$Q(\text{next}) = JQ' + K'Q$$

جدول التهييج والذي يحدد القيم الواجب توفرها على المدخل لنقل

النطاق من الحالة الحالية الى حالة تالية معروفة هو:

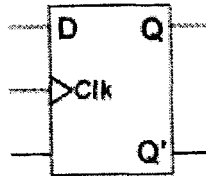
$Q$	$Q(\text{next})$	$J$	$K$
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

### النطاق D

نطاق يشبه الى حد ما النطاق JK لكن بوصل المدخل J مع K عن طريق

بوابة نفي.

يبين الشكل التالي المخطط الصندوقي لهذا النطاق:



ينتقل هذا النطاق من الحالة الحالية الى الحالة المقبلة أو التالية بوجود

قيم المدخل التالية والمبينة في جدول الصواب التالي لهذا النطاق:

$D$	$Q(next)$
0	0
1	1
0	0

المعادلة المميزة والتي تربط الحالة التالية بقيم المدخل والحالة الحالية هي:

$$Q(next) = D$$

جدول التهيج والذي يحدد القيم الواجب توفرها على المدخل لنقل

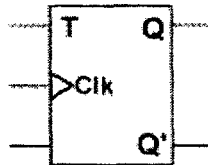
النظام من الحالة الحالية الى حالة تالية معروفة هو:

$Q$	$Q(next)$	$D$
0	0	0
0	1	1
1	0	0
1	1	1

### النظام T

نظام يشبه الى حد ما النظام JK لكن بوصول المدخل J مع K.

يبين الشكل التالي المخطط الصندوقي لهذا النظام:



ينتقل هذا النطاق من الحالة الحالية الى الحالة المقبلة أوالتالية بوجود

قيم المداخل التالية والمبينة في جدول الصواب التالي لهذا النطاق:

$T$	$Q(next)$
0	Q
1	Q'
T	Q(next)

المعادلة المميزة والتي تربط الحالة التالية بقيم المداخل والحالة الحالية هي:

$$Q(next) = TQ' + T'Q$$

جدول التهييج والذي يحدد القيم الواجب توفرها على المداخل لنقل

النطاق من الحالة الحالية الى حالة تالية معروفة هو:

$Q$	$Q(next)$	$T$
0	0	0
0	1	1
1	0	1
1	1	0

وفيما يلي جدول يبين ملخص جداول التهييج للنطاقات السابقة والتي

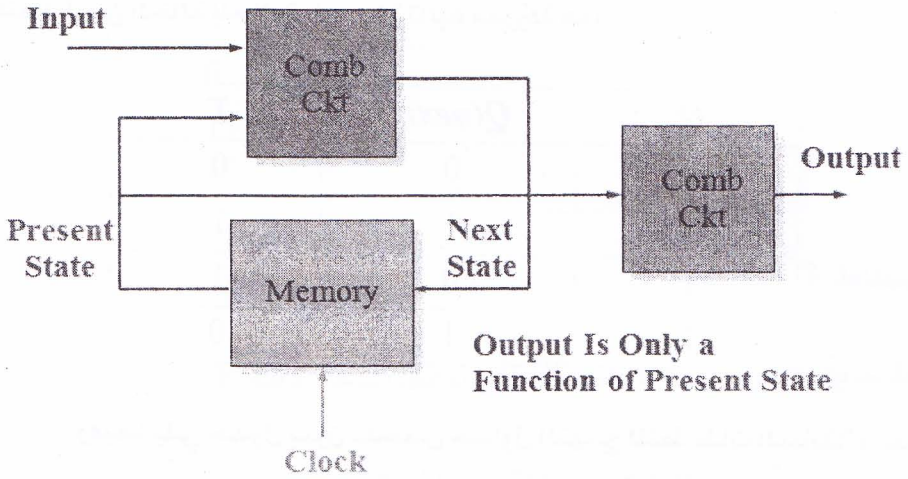
يمكن استخدامها في عملية التصميم:

Desired transition		Triggering signal needed					
$Q(t)$	$Q(t+1)$	S	R	J	K	D	T
0	0	0	x	0	x	0	0
0	1	1	0	1	x	1	1
1	0	0	1	x	1	0	1
1	1	x	0	x	0	1	0

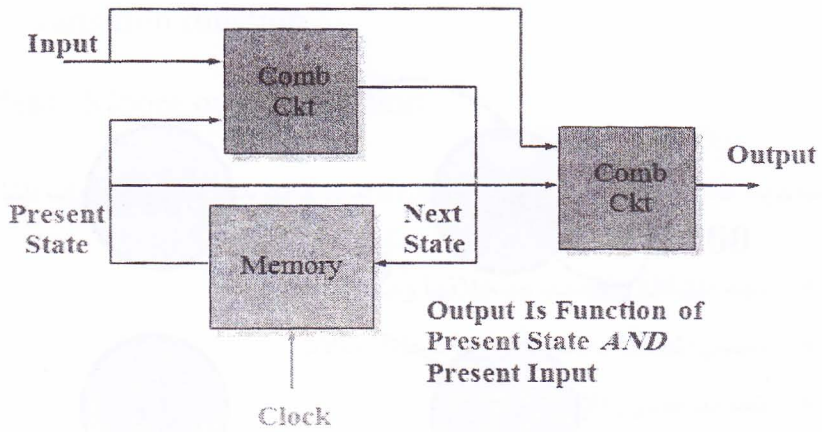
## 2.6 آلة مورو آلة ميلي:

تعتبر هاتان الآلتان نماذج من الآلة الحالة النتهية ويتألف كل منها من دائرة تجميعية وذاكرة وتشبه الآلة مورو آلة ميلي لكن بخلاف بسيط في دالة الانتقال لكل منهما ففى الآلة مورو يتم توليد المخرجات عند استقرار الآلة في حالة معينة اي ان المخرجات تشكل اقترانا يعتمد على الحالة الحالية اما في الآلة ميلي فيتم توليد المخرجات عند انتقال الآلة من حالة الى اخرى اي ان المخرجات هي اقترانات تعتمد على قيم المدخلات الحالية وقيمة الحالة الحالية ويبين الشكلان التاليان الخلاف بين هاتين الآلتين:

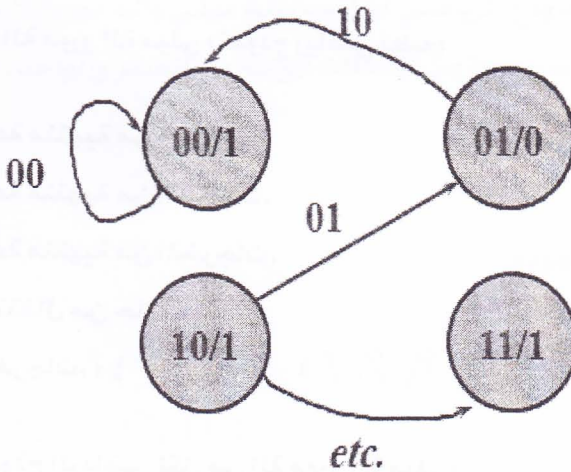
## Synchronous Moore Machine



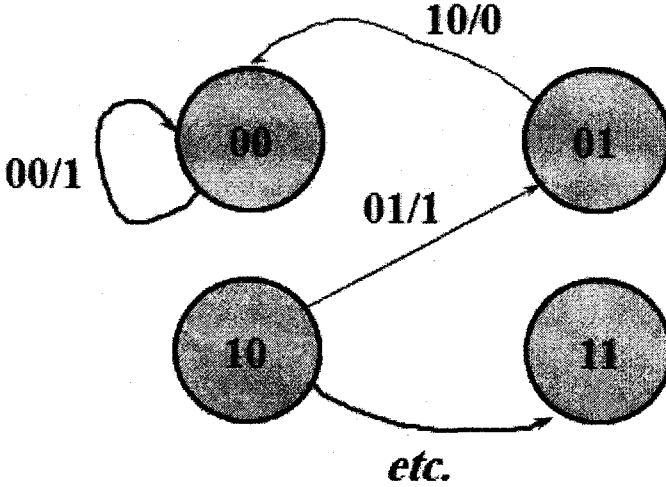
## Synchronous Mealy Machine



تمثل الآلة موروا بمخطط الحالات وذلك بوضع المخرجات في الحالة وكما يلي:



اما في آلة ميلي فيتم تمثيل المخرجات عند الإنتقال من حالة الى اخرى كما يلي:



تمثل كل من آلة موروا آلة ميلي بنموذج رياضي يضم:

- مجموعة منتهية من الحالات.
- مجموعة منتهية من المدخلات.
- مجموعة منتهية من المخرجات.
- دالة الانتقال من حالة لاخرى.
- دالة المخرجات.

وفيما يلي النموذج الرياضي لكل من آلة موروا آلة ميلي:

**Machine is a quintuple of sets**

$$M = (S, I, O, \delta, \beta )$$

S: Finite set of states

I: Finite set of inputs

O: Finite set of outputs

$\delta$ : State transition function

$\beta/\lambda$ : Mealy/Moore output function

ولتحديد عناصر هذا النموذج ولأي من الآلتين لا بد من اتباع الخطوات التالية:

- فهم المشكلة وتحديد مدخلاتها ومخرجاتها.
- تمثيل المشكلة باستخدام مخطط الحالات.
- تحديد جدول الإنتقال.

مثال:

حدد النموذج الرياضي لكل من آلة ميلي وآلة موروا اللازمة لاكتشاف

المتابع 0101 في مجموعة من المدخلات المؤلفة من الصفر والواحد.

الآلة ميلي:

- مكونات النموذج.

$$M = ( S, I, O, \delta, \beta )$$

$S$ : { A, B, C, D }

$I$ : { 0, 1 }

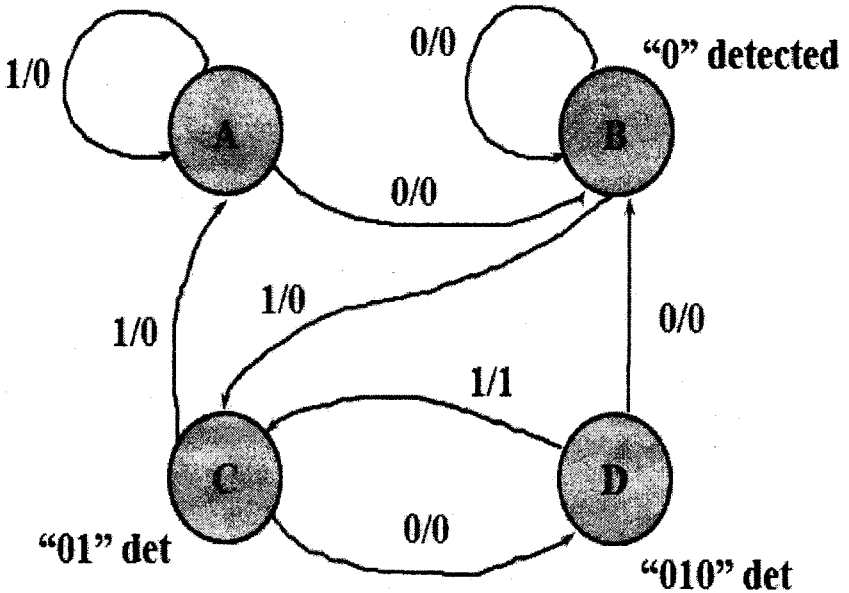
$O$ : { 0, 1 } = { not detected, detected }

$\delta$ : next figure

$\beta$ : next figure



• مخطط الحالات:



• جدول الانتقال (دوال الانتقال):

Present State	Present Input	
	0	1
A	B/0	A/0
B	B/0	C/0
C	D/0	A/0
D	B/0	C/1

Next State/Output

آلة مور:

• مكونات النموذج:

$$M = ( S, I, O, \delta, \lambda )$$

$S$ : { A, B, C, D, E }

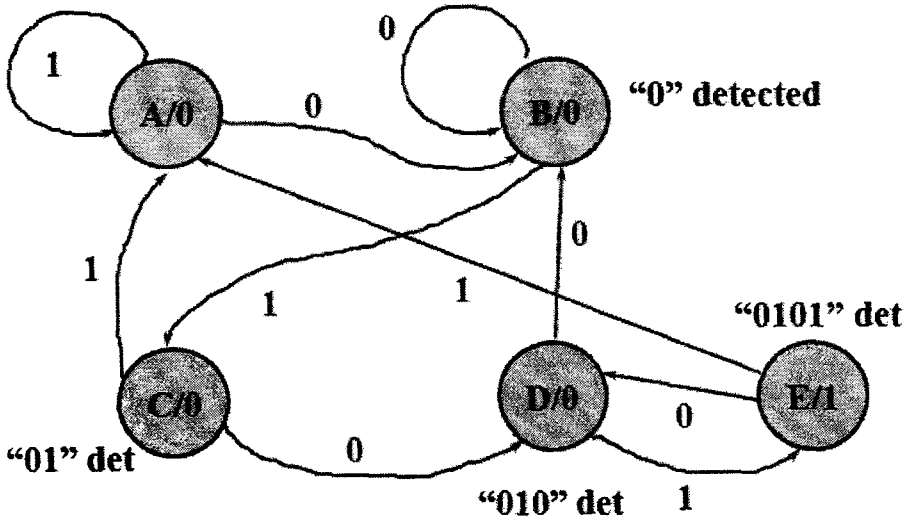
$I$ : { 0, 1 }

$O$ : { 0, 1 } = { not detected, detected }

$\delta$ : next figure

$\lambda$ : next figure

• مخطط الحالات:



• جدول الانتقال:

Present State	Present Input		Output( $\lambda$ )
	0	1	
A	B	A	0
B	B	C	0
C	D	A	0
D	B	E	0
E	D	A	1



Next State

## 3.6 تصميم آلة الحالة المنتهية:

تستخدم في عملية تصميم أي من آلة ميلي أو مور الخطوات الأساسية التالية:

1. فهم المشكلة بتحديد المدخلات والمخرجات وكيفية توليدها.
2. تمثيل المشكلة بمخطط الحالات.
3. استخدام النظام الثنائي في ترقيم الحالات لتحديد عدد النطاقات المطلوبة.
4. تحديد نوع النطاق المراد استخدامه في عملية التصميم.
5. بناء جدول الانتقال وباستخدام جدول التهييج للنطاق.
6. استخراج المعادلات المختصرة لكل مدخل من مداخل النطاق واستخراج المعادلة المختصرة للمخرجات.
7. تمثيل الآلة باستخدام بوابات ودارات المنطق.

مثال:

صمم آلة الحالة المنتهية لاكتشاف 3 وحدات متتابعة في مجموعة من المدخلات المؤلفة من الصفر والواحد.

الحل:

- نحدد الحالات لآلة مور كما يلي:

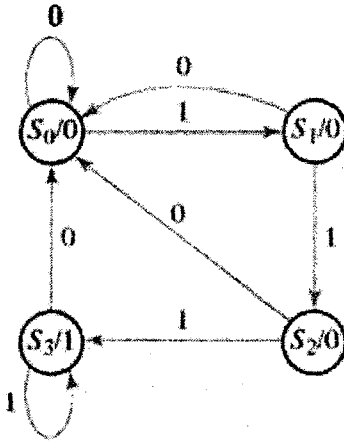
**State  $S_0$  : zero 1s detected**

**State  $S_1$  : one 1 detected**

**State  $S_2$  : two 1s detected**

**State  $S_3$  : three 1s detected**

• نرسم مخطط الحالات:



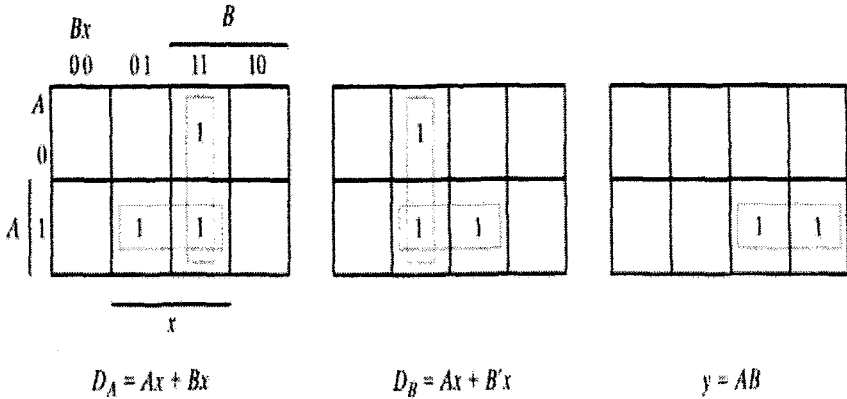
• نرقم الحالات باستخدام النظام الثنائي:

- °  $S_0 = 00$
- °  $S_1 = 01$
- °  $S_2 = 10$
- °  $S_3 = 11$

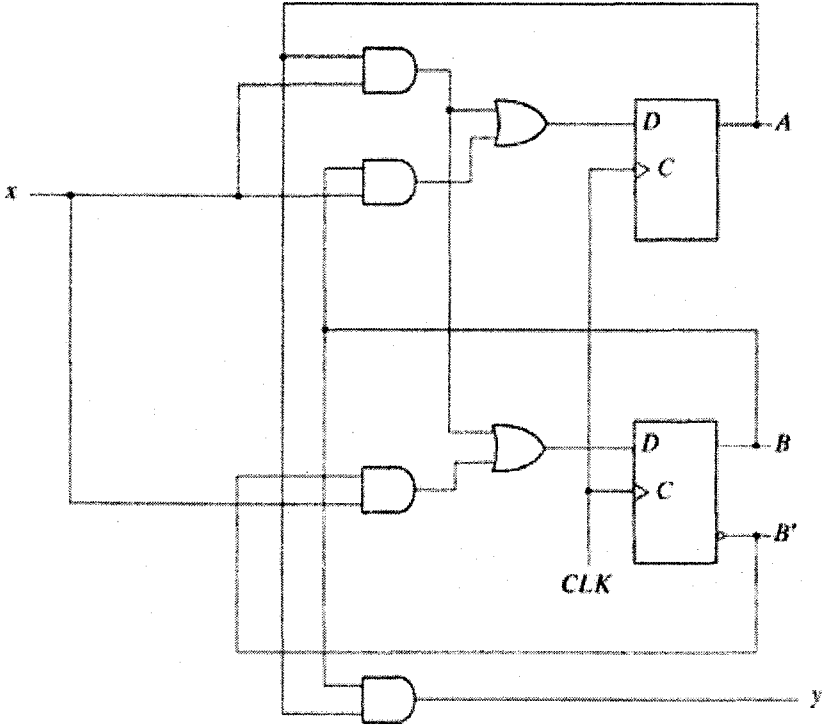
- عدد النشاطات المطلوب هو 2 ولنختار النشاط D.
- نبنى جدول الانتقال لاحظ هنا ان مدخل النشاط يساوي الحالة المقبلة:

Present State			Input	Next State		Output
A	B	A		B		
0	0	0	0	0	0	
0	0	1	0	1	0	
0	1	0	0	0	0	
0	1	1	1	0	0	
1	0	0	0	0	0	
1	0	1	1	1	0	
1	1	0	0	0	1	
1	1	1	1	1	1	

- نستخدم خرائط كارنو للحصول على المعادلات المختصرة لمداخل النشاطات ودالة المخرج:



- باستخدام المعادلات المختصرة نبني آلة مور المنطقية باستخدام دارات المنطق:



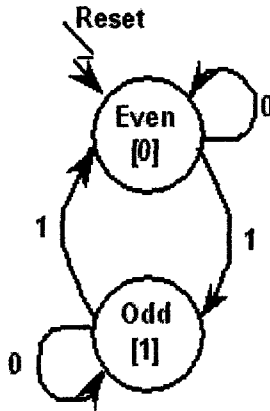
مثال:

فحص التطابق الفردي

فيما يلي آلة مور لفحص التطابق الفردي وإنتاج 1 عندما يكون عدد

الوحدات المقروءة فرديا:

- مخطط الحالات



• نبني جدول الانتقال.

Present State	Input	Next State	Output
Even	0	Even	0
Even	1	Odd	0
Odd	0	Odd	1
Odd	1	Even	1

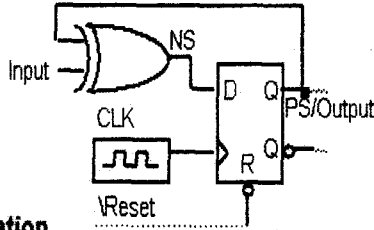
### Symbolic State Transition Table

Present State	Input	Next State	Output
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

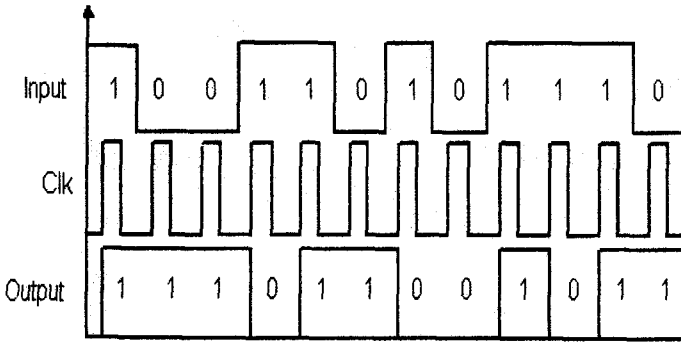


- نحدد المعادلات ونبنى الدارة المنطقية.

$$NS = PS \text{ xor } PI; \text{ OUT} = PS$$

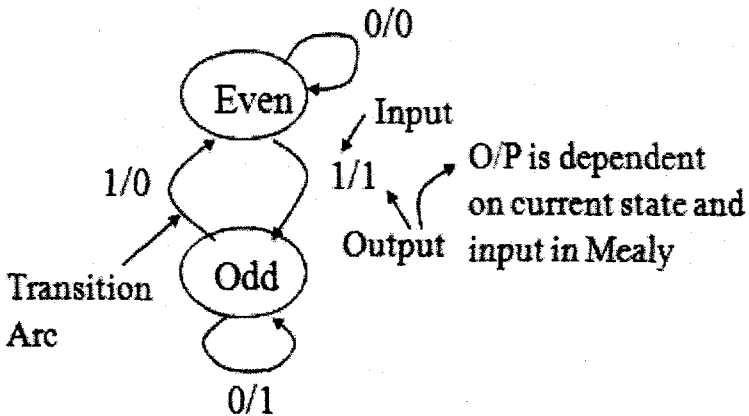


D FF Implementation



Timing Behavior: Input 1 0 0 1 1 0 1 0 1 1 1 0

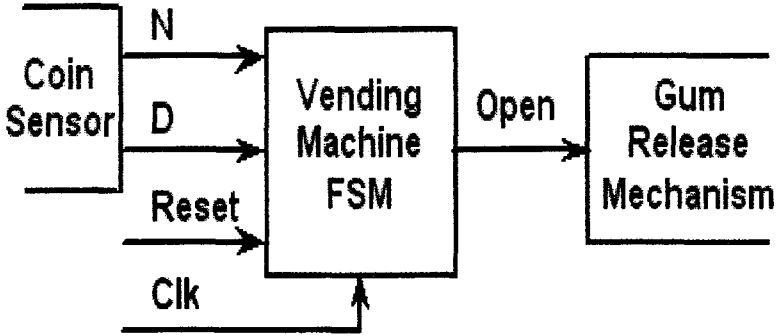
اما مخطط الحالات لآلة ميلي فهو كما يلي:



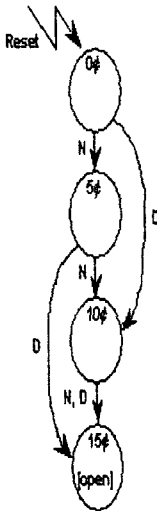
مثال:

تصميم وحدة ذاتية لإعطاء القهوة في آلة القهوة الاوتوماتيكية.

يمكن تصور هذه الآلة كما يلي:



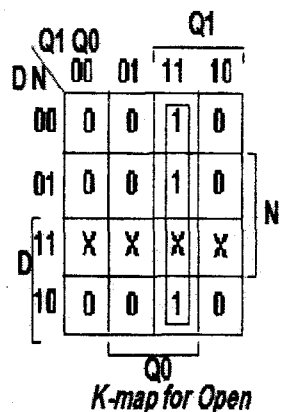
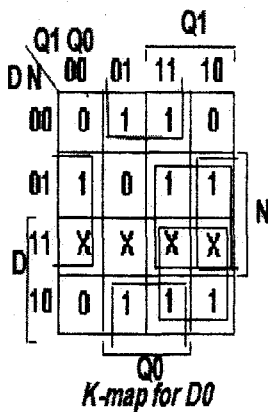
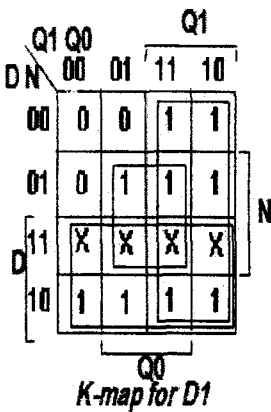
• مخطط الحالات وجدول الانتقال:



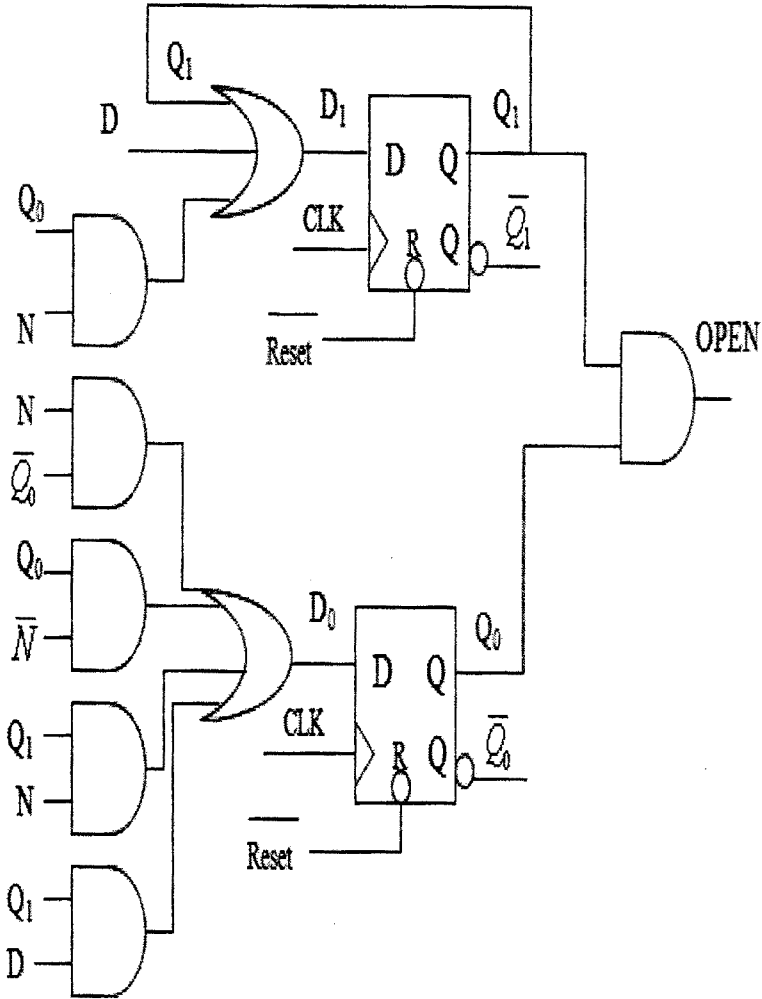
Present State	Inputs		Next State	Output
	D	N		Open
0¢	0	0	0¢	0
	0	1	5¢	0
	1	0	10¢	0
	1	1	X	X
5¢	0	0	5¢	0
	0	1	10¢	0
	1	0	15¢	0
	1	1	X	X
10¢	0	0	10¢	0
	0	1	15¢	0
	1	0	15¢	0
	1	1	X	X
15¢	X	X	15¢	1

Present State		Inputs		Next State		Output
$Q_1$	$Q_0$	D	N	$D_1$	$D_0$	Open
0	0	0	0	0	0	0
		0	1	0	1	0
		1	0	1	0	0
		1	1	X	X	X
0	1	0	0	0	1	0
		0	1	1	0	0
		1	0	1	1	0
		1	1	X	X	X
1	0	0	0	1	0	0
		0	1	1	1	0
		1	0	1	1	0
		1	1	X	X	X
1	1	0	0	1	1	1
		0	1	1	1	1
		1	0	1	1	1
		1	1	X	X	X

• اختصار المعادلات:



• تمثيل الآلة ببوابات المنطق:



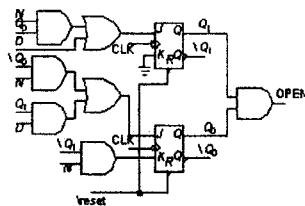
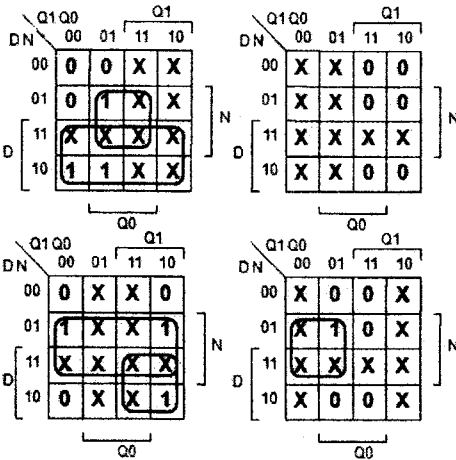
التصميم باستخدام النطاق JK

هذه الحالة نستخدم جدول التهييج لهذا النطاق لبناء جدول الانتقال:

JK Excitation Table

Present State	Inputs	Next State	J <sub>1</sub>	K <sub>1</sub>	J <sub>0</sub>	K <sub>0</sub>
Q <sub>1</sub> Q <sub>0</sub>	D N	Q <sub>1</sub> Q <sub>0</sub>				
0 0	0 0	0 0	0	X	0	X
	0 1	0 1	0	X	1	X
	1 0	1 0	1	X	0	X
	1 1	X X	X	X	X	X
0 1	0 0	0 1	0	X	X	0
	0 1	1 0	1	X	X	1
	1 0	1 1	1	1	X	0
	1 1	X X	X	X	X	X
1 0	0 0	1 0	X	X	0	0
	0 1	1 1	X	0	1	X
	1 0	1 1	X	0	1	X
	1 1	X X	X	X	X	X
1 1	0 0	1 1	X	0	X	0
	0 1	1 1	X	0	X	0
	1 0	1 1	X	0	X	0
	1 1	X X	X	X	X	X

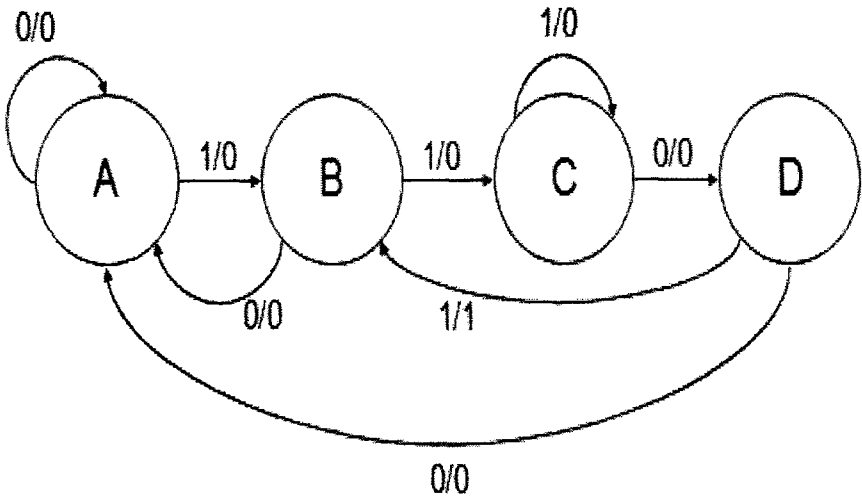
نستخرج المعادلات ز نرسم المخطط المنطقي للالة:



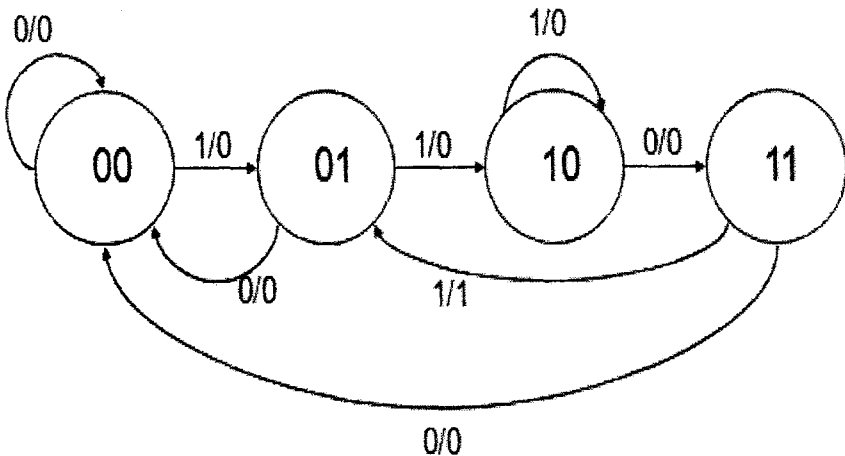
مثال:

صمم الآلة ميلي لتوليد 1 بعد قراءة 1101 في مجموعة من المدخلات المؤلفة من الصفر والواحد (او لإكتشاف النمط 1101)

• مخطط الحالات:



• ترقيم الحالات:



• نبني جدول الانتقال:

Present State		Input	Next State		Output
A	B	X	A	B	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	1	1

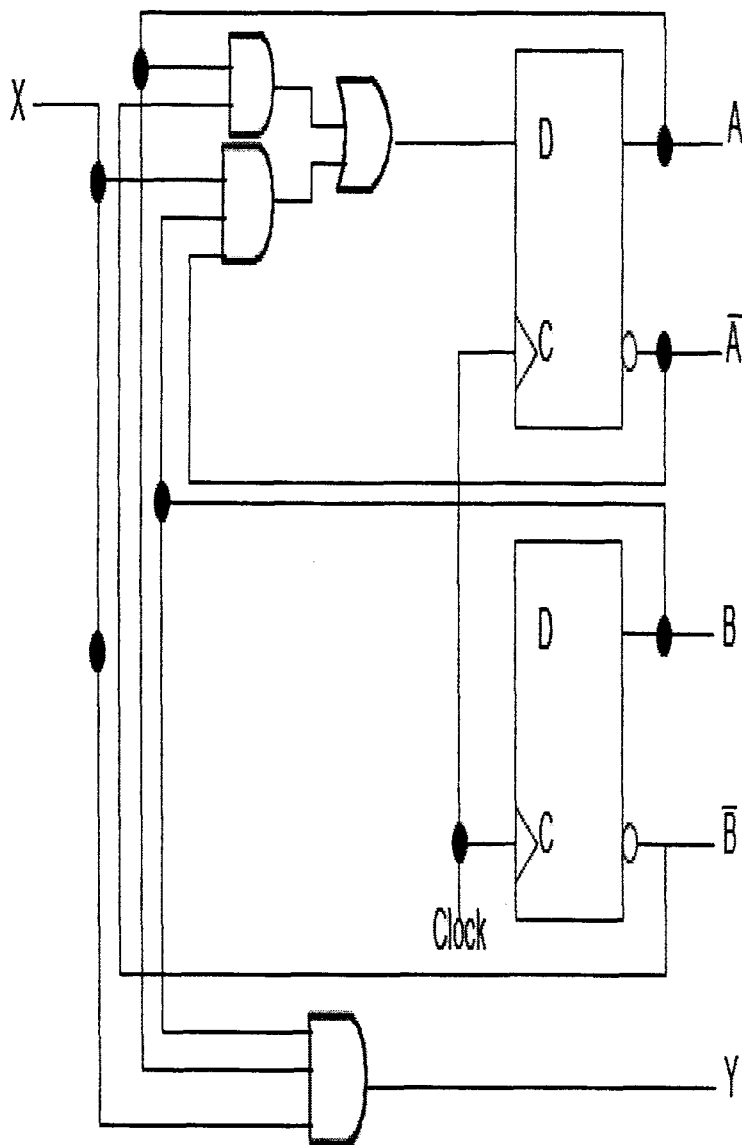
• نستخرج المعادلات:

$$A_{\text{next}} = A'BX + AB'$$

$$B_{\text{next}} = A'B'X + AB'X' + ABX$$

$$Y = ABX$$

• تمثل الآلة ببوابات المنطق:

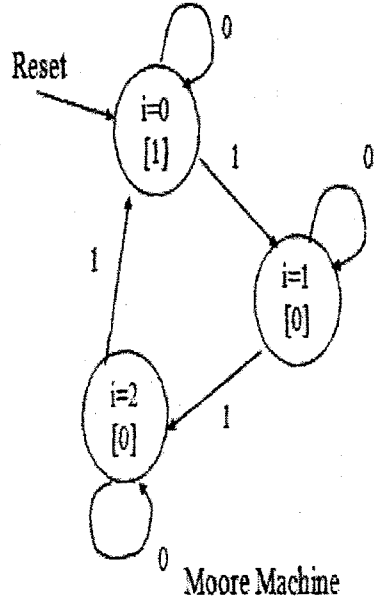
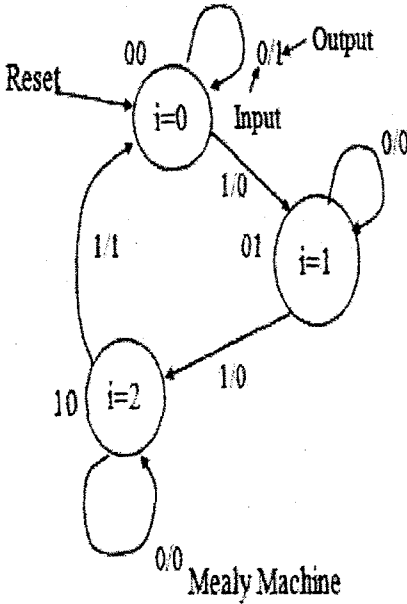




سؤال:

صمم آلة الحالة لتوليد 1 عند يكون عدد الوحدات المقروءة من مضاعفات الرقم 3.

استعن بمخطط الحالات التالي:



## References

Barendregt, Hendrik Pieter. *The Lambda Calculus: Its Syntax and Semantics*. North-Holland (Amsterdam, 1981). ISBN 0-444-85490-8. *Studies in Logic and the Foundations of Mathematics*, vol. 103.

Chaitin, Gregory J. *The Limits of Mathematics: A Course on Information Theory and the Limits of Formal Reasoning*. Springer-Verlag Singapore (Singapore: 1998). ISBN 981-3083-59-X.

Chaitin, Gregory J. *The Unknowable*. Springer-Verlag Singapore (Singapore: 1999). ISBN 981-4021-72-5 (hardcover)

Church, Alonzo. *An Unsolvability Problem of Elementary Number Theory*. *American Journal of Mathematics* 58 (1936), 345-363. Reprinted in pp. 88-107 of [Davis1965]

Davis, Martin (ed.). *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvability Problems and Computable Functions*. Raven Press (New York: 1965). ISBN 0-911216-01-4.

Davis, Martin. *Computability and Unsolvability*. Dover (New York: 1958, 1973, 1982). ISBN 0-486-61471-9 pbk.

Garey, Michael R., Johnson, David S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman (New York: 1979). ISBN 0-7167-1045-5 pbk.

John E., Motwani, Rajeev., Ullman, John D. *Introduction to Automata Theory, Languages, and Computation*. ed.2. Addison-Wesley (Boston, MA: 2001). ISBN 0-201-44124-1.

---

Garey, Michael R., Johnson, David S. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (New York: 1979). ISBN 0-7167-1045-5 pbk. See [Garey1979]

Lewis, Harry R., Papadimitriou, Christos H. Elements of the Theory of Computation. Prentice-Hall (Englewood Cliffs, NJ: 1981). ISBN 0-13-273417-6.

Lifschitz, Vladimir (ed.) Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy. Academic Press (San Diego: 1991). ISBN 0-12-450010-2.

Hopcroft, John E., Motwani, Rajeev., Ullman, John D. Introduction to Automata Theory, Languages, and Computation. ed.2. Addison-Wesley (Boston, MA: 2001). ISBN 0-201-44124-1. See [Hopcroft2001]

Lewis, Harry R., Papadimitriou, Christos H. Elements of the Theory of Computation. Prentice-Hall (Englewood Cliffs, NJ: 1981). ISBN 0-13-273417-6. See [Lewis1981]

Révész, Gy. rgy E. Lambda-Calculus, Combinators and Functional Programming. Cambridge University Press (Cambridge, 1988). ISBN 0-521-34589-8.

Sipser, Michael. Introduction to the Theory of Computation. PWS Publishing (Boston, MA: 1997). ISBN 0-534-94728-X.

Hopcroft, John E., Motwani, Rajeev., Ullman, John D. Introduction to Automata Theory, Languages, and Computation. ed.2. Addison-Wesley (Boston, MA: 2001). ISBN 0-201-44124-1. See [Hopcroft2001]