الفصل الثالث

المجموعات المنتظمة والتعابير المنتظمة

- المجموعات المنتظمة
- التعابير المنتظمة
- الحصول على تعبير منتظم من الأتمتة المحدودة
- الطريقة التجريبية لرسم الأتمتة غير المحددة

  وإيجاد المجموعة المنتظمة
- مميز الـ FSA
- خوارزمية Aho & corasick
- تمارين

الفصل الثالث: المجموعات المنتظمة والتعابير المنتظمة

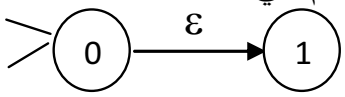
3.1 المجموعات المنتظمة Regular Sets

المجموعة المنتظمة هي مجموعة كلمات (سلاسل) بحيث توجد أتمتة محدودة (finite automata) تقبل تلك المجموعة (أي هناك مجموعة من الكلمات تكون مقبولة في تلك الأتمتة وبمعنى آخر إذا كانت R مجموعة منتظمة فان R = L(M) حيث أن M أتمتة محدودة و L تمثل اللغة المقابلة للقواعد الموصوفة بالتعبير المنتظم، وكذلك إذا كانت M أتمتة محدودة فان L(M) هي دائما مجموعة منتظمة.

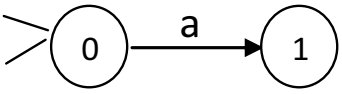
3.2 التعابير المنتظمة Regular Expression (RE)

يعتبر التعبير المنتظم Regular Expression(RE) طريقة جيدة لوصف القواعد وهو صيغة لتحديد المجموعة المنتظمة (Regular Set) وان لكل تعبير منتظم أتمتة تقبل اللغة المحددة من قبل التعبير المنتظم. وبالمقابل فان لكل أتمتة محدودة M يوجد هناك تعبير منتظم يحدد اللغة المقابلة له L(M) ، إن المكونات الرئيسية للتعابير المنتظمة هي :

a) إن **ε** يعتبر تعبير منتظم وان المجموعة المنتظمة له هي $L=\{\varepsilon\}$ والطريقة الرسمية لرسم هذا التعبير المنتظم هي :

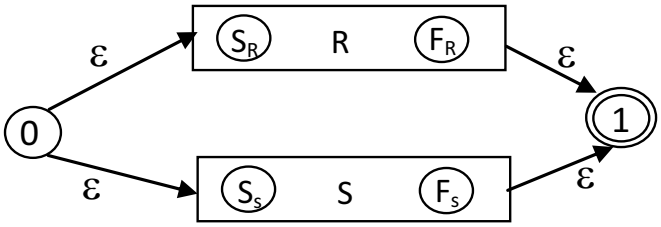


b) أي **عنصر غير قابل للاشتقاق** (terminal) a فهو تعبير منتظم والمجموعة المنتظمة له هي $L=\{ a \}$ والطريقة الرسمية لرسم هذا التعبير المنتظم هي:



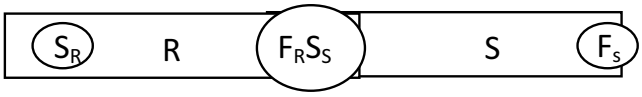
c) إذا كان كل من R و S تعبير منتظم وله المجموعة المنتظمة له وهي على التوالي L_R , L_S فان :

i. **R |S** تعبير منتظم وان المجموعة المنتظمة له هي $L_S \cup L_R$ والطريقة القانونية لرسم هذا التعبير المنتظم هي:

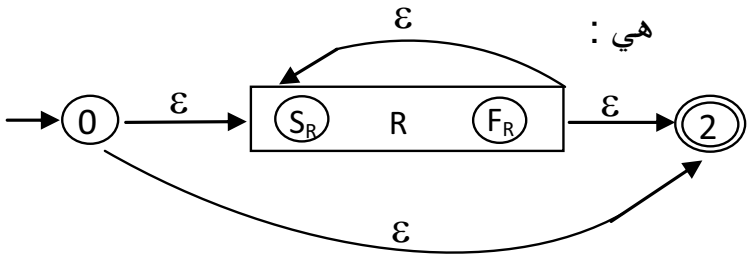


حيث أن S_R مختصر لـ(Start of R) وتعني حالة البداية للتعبير المنتظم R، أما F_R فهي مختصر لـ (Final of R) وتعني حالة النهاية للتعبير المنتظم R وهكذا لبقية الرموز المشابهة في الرسم أعلاه.

ii. **S . R** تعبير منتظم وان المجموعة المنتظمة له هي L_R . L_S والطريقة الرسمية لرسم هذا التعبير المنتظم هي:



iii. ***R** تعبير منتظم وان المجموعة المنتظمة له هي $L_R^0 \cup L_R^1 \cup ... L_R^n$ والطريقة الرسمية لرسم هذا التعبير المنتظم هي :



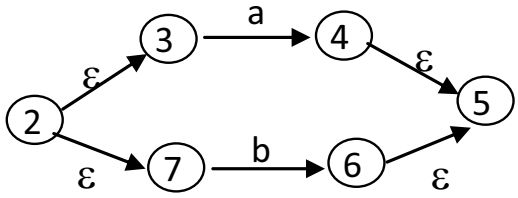
ملاحظة: إن القواعد (Grammar) المقابلة للتعبير المنتظم تسمى القواعد المنتظمة (Regular Grammar RG) .

مثال3.1: هل أن a|b تعبير منتظم ؟ ما هي المجموعة المنتظمة له ؟ وما هي الأتمتة غير المحددة له؟

a تعبير منتظم والمجموعة المنتظمة له L={a}

b تعبير منتظم والمجموعة المنتظمة له L={b}

a|b تعبير منتظم والمجموعة المنتظمة له L={a,b} حسب القواعد السابقة . أما الشكل (3.1) فيمثل الأتمتة المقابلة له بالطريقة القانونية.



شكل (3.1) الأتمتة الغير المحددة للتعبير المنتظم a|b

مثال3.2 : هل أن abb*(a|b) تعبير منتظم ؟ ما هي المجموعة المنتظمة له ؟ وما هي الأتمتة غير المحددة له؟

a تعبير منتظم والمجموعة المنتظمة له هي L={a}

b تعبير منتظم والمجموعة المنتظمة له هي L={b}

a|b تعبير منتظم والمجموعة المنتظمة له هي L={a,b}

*(a|b) تعبير منتظم والمجموعة المنتظمة له هي

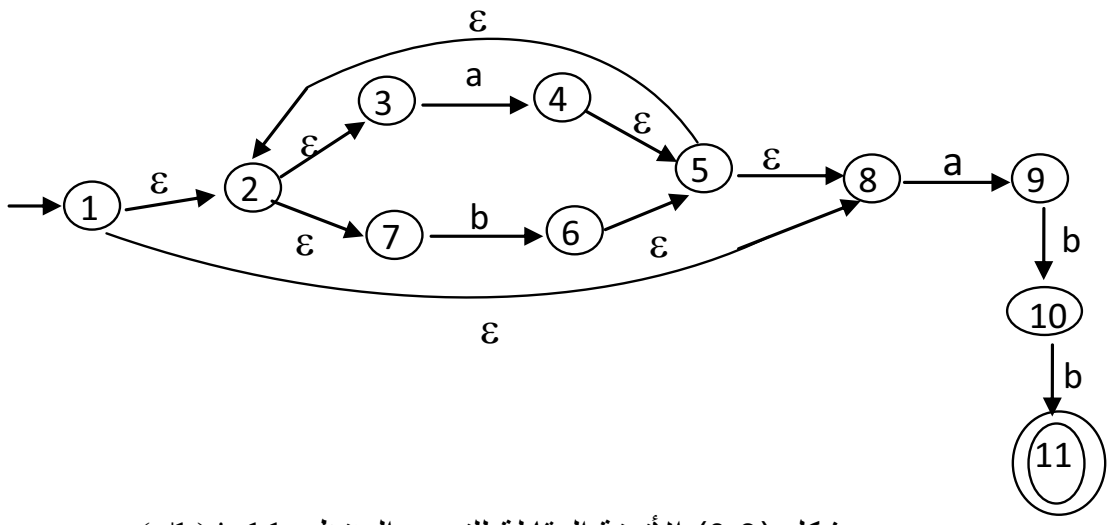
L={ε, a , b ,aabb , aaabbb, abab, aaa, bbaab,……}

abb تعبير منتظم والمجموعة المنتظمة له هي L={abb}

وأخيرا abb*(a|b) تعبير منتظم والمجموعة المنتظمة له هي

L={abb ,aabb, babb ,aabbabb , aaabbbabb, abababb, aaaabb, bbaababb,……}

أما الشكل (3.2) فيمثل الأتمتة المقابلة له بالطريقة القانونية .



شكل (3.2) الأتمتة المقابلة للتعبير المنتظم (a|b)*abb

مثال3.3: هل أن a(a|b)b تعبير منتظم وما هي المجموعة المنتظمة له ؟ وما هي الأتمتة غير المحددة له بالطريقة القانونية ؟

a تعبير منتظم والمجموعة المنتظمة له هي L={a}

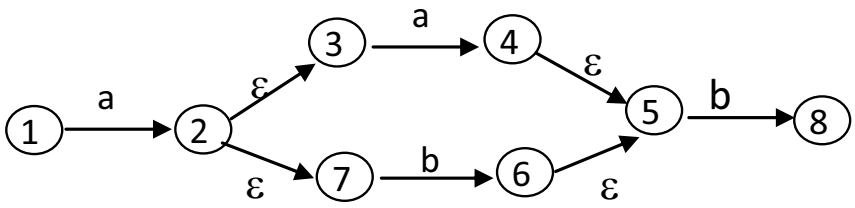
b تعبير منتظم والمجموعة المنتظمة له هي L={b}

a|b تعبير منتظم والمجموعة المنتظمة له هي L={a,b}

a(a|b) تعبير منتظم والمجموعة المنتظمة له هي L={aa,ab}

a(a|b)b تعبير منتظم والمجموعة المنتظمة له هي L={aab, abb}

أما الشكل (3.3) فيمثل الأتمتة المقابلة له بالطريقة القانونية .



شكل (3.3) الأتمتة المقابلة للتعبير المنتظم a(a|b)b

من خصائص التعابير المنتظمة ما يلي :

- S|R تكافئ R|S
- (S|R)|T تكافئ S|(R|T)
- (S|R)|T تكافئ R|(S|T)
- (R.S).T تكافئ R.(S.T)
- R.S | R.T تكافئ R.(S|T)
- R تكافئ $R.\in$ وتكافئ $\in .R$

مثال3.4: إذا كان S = a | b وكان R = (a | b)* فان R | S ≡ S | R

المجموعة المنتظمة للتعبير المنتظم S هي { a , b} أما المجموعة المنتظمة للتعبير المنتظم R هي {a,b,aa,ab,ba,bb,…} لذا فان المجموعة المنتظمة للتعبير المنتظم S | R هي :

{ a,b, aa,ab,ba,bb,…}

والمجموعة المنتظمة للتعبير المنتظم R | S هي:

{ a,b, aa,ab,ba,bb,…}

ولان اللغات المقابلة لكلا التعبيرين المنتظمين متشابهة ومتساوية فان التعبير المنتظم S | R يكافئ التعبير المنتظم S | R.

3.3 الحصول على تعبير منتظم من الأتمتة المحدودة

إذا كانت لدينا أتمتة محدودة FSA، للحصول على تعبير منتظم الذي من خلاله يتم تحديد المجموعة المنتظمة وتكون مقبولة من الأتمتة المحدودة المعطاة فان هناك خطوات ضرورية:

1. Associate suitable variables (e.g. A, B, C, etc.) with the states of finite automata.
2. From a set of equations using the following rules:
    a. If there exists a transition from a state associated with variables A to a state associated with variable B on an input symbol a, then add the equation

        A=aB to the set of equation

    b. If the state associated with variable A is a final state, add A= ε to the set of equation

    c. If we have the two equations A=aB and A=bC, then they can be combined as A=aB | bC

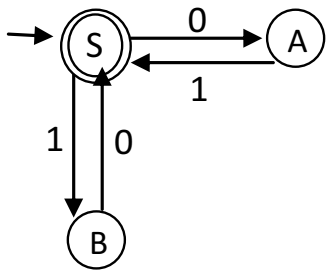
    d. Solve these equations to get the value of the variable associated with the starting state of the automata. In order to solve these equations, it is

necessary to bring the equation in the following form :

S = aS | b

Where S is a variable, and a and b are expressions that do not contain S. The solution to this equation is S=a*b.

مثال3.5: لو كانت لدينا الأتمتة المحدودة في الشكل (3.4) .



شكل (3.4) أتمتة محددة (DFA)

سنستخدم أسماء الحالات(states) كأسماء للمتغيرات (variables) كما هو مطلوب في الخطوات الضرورية المشار إليها أعلاه. إن مجموعة المعادلات التي نحصل عليها من تطبيق القواعد أعلاه هي:

S = 0A | 1B | ε (I)

A = 1S (II)

B = 0S (III)

حيث أن الانتقالة بين الحالة S والحالة A هي عن طريق المدخل 0 والانتقالية بين الحالة S والحالة B هي عن طريق المدخل 1 ولان الحالة S نهائية فإنها تؤدي إلى **ε.**

لحل المعادلات أعلاه نقوم بتعويض المعادلتين (II) و (III) في المعادلة (I) لنحصل على:

$S = 01S \mid 10S \mid \varepsilon$

$S = (01 \mid 10)\ S \mid \varepsilon$

لذا فان المتغير S سيصبح :

$S = (01 \mid 10)^* \mid \varepsilon \quad = (01 \mid 10)^*$

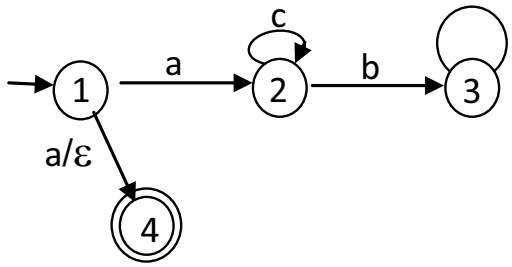
وبالتالي فان التعبير المنتظم الذي يحدد المجموعة المنتظمة المقبولة في الأتمتة المحدودة أعلاه هو :

$(01 \mid 10)^*$

3.4 الطريقة التجريبية لرسم الأتمتة غير المحددة وإيجاد المجموعة المنتظمة

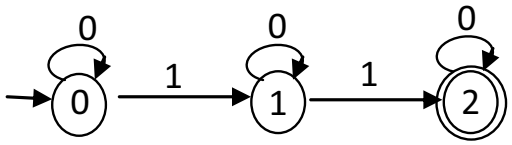
بينا من قبل عملية رسم الأتمتة الغير محددة للتعابير المنتظمة بالطريقة الرسمية كما تم تبيان عملية إيجاد المجموعة المنتظمة للقواعد الممثلة بتلك التعابير المنتظمة ، أما الآن فسيتم إعطاء أمثلة على كيفية رسم أتمتة غير محددة للتعابير المنتظمة بطريقة غير رسمية (Empirical Method) كما سيتم شرح من خلال الأمثلة أيضا عملية إيجاد المجموعة المنتظمة للقواعد الممثلة بتلك التعابير المنتظمة.

مثال3.6: الشكل (3.5) يمثل أتمتة غير محددة للتعبير المنتظم $RE=(a|ac^*b)^*$ بالطريقة غير القانونية (الطريقة التجريبية) .



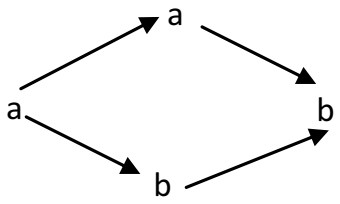
شكل (3.5) أتمتة غير محددة للتعبير المنتظم $(a|ac^*b)^*$

مثـال3.7 : الشـكل (3.6) يمثـل أتمتـة غيـر محـددة للتعبيـر المنـتظم RE=0*10*10* بالطريقة غير القانونية (الطريقة التجريبية) .



شكل (3.6) أتمتة غير محددة للتعبير المنتظم 0*10*10*

مثال3.8 : ما هي المجموعة المنتظمة للتعبير المنتظم a(a|b)b بالطريقة غير الرسمية ؟



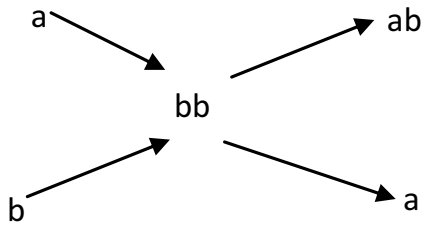
شكل (3.7) الأتمتة المقابلة للتعبير المنتظم a(a|b)b

من خلال الشكل (3.7) نلاحظ انه عندما نبدأ بالرمز a بعدها يوجد طريقين احدهما إلى الرمز a ثم إلى الرمز b والطريق الآخر إلى الرمز b ثم إلى الرمز b وهذان الطريقان يشكلان المجموعة المنتظمة لذلك التعبير المنتظم:

L={aab, abb}

مثـال3.9 : هـل أن (a|b)bb(ab|a) تعبيـر منـتظم ؟ ومـا هـي المجموعـة المنتظمـة لـه بالطريقة التجريبية؟

لـو طبقنـا قواعـد التعبيـر المنـتظم لوجـدنا أن (a|b)bb(ab|a) هـو تعبيـر منـتظم ولإيجاد المجموعة المنتظمة له بالطريقة التجريبية نلاحظ الرسم في الشكل (3.8):



شكل (3.8) الأتمتة المقابلة للتعبير المنتظم (a|b)bb(ab|a)

في مثالنا هذا من البداية هناك طريقين احدهما يبدأ بالرمز a والطريق الثاني يبدأ بالرمز b وكلاهما يذهبان إلى bb وهنا يتم التفرع إلى طريقين الأول إلى ab والثاني إلى a فهناك عدة طرق: abbab ، abba ، bbbab و bbba لتتشكل منهم المجموعة المنتظمة لذلك التعبير المنتظم وهي كما يلي :

L={abbab ,abba ,bbbab ,bbba }

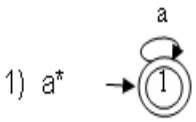
إن هناك تعابير منتظمة وما يقابلها من لغات جاهزة مثل :

- a* ➔ L={∈ ,a ,aa ,aaa ,aaaa ,……..}
- a+ ➔ L={ a ,aa ,aaa ,aaaa ,……..}
- (ab)* ➔ L={∈ ,ab ,abab ,ababab ,……}
- a* |b* ➔ L={∈ ,a ,aa ,aaa ,aaaa ,……..} ∪ {∈ , b ,bb ,bbb ,bbbb ,…}={ ∈ ,a ,aa ,aaa, aaaa , b, bb , bbb ,bbbb,… }

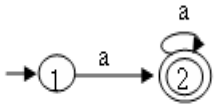
- a*b* ➔ L={ ∈,a ,aa ,b ,bb ,ab ,aabb ,aaab,abbbb,…….}
- (a|b)* ➔ L={∈,a ,aa ,b ,bb ,ab ,aabb ,aaab,baba ,bba,…….}

وهناك بعض القوالب للتعابير المنتظمة والأتمتة المقابلة لها :

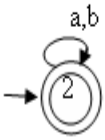
1) a*



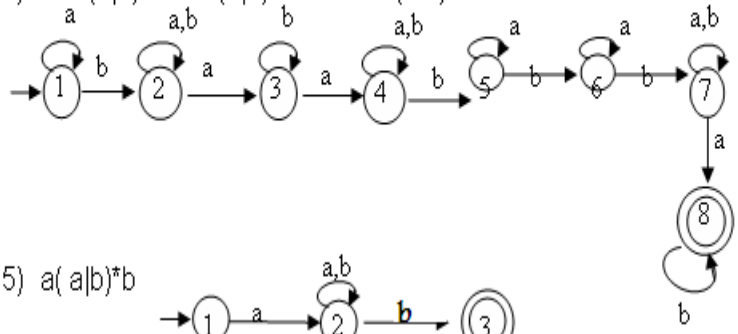
2) a+



3) (a|b)*

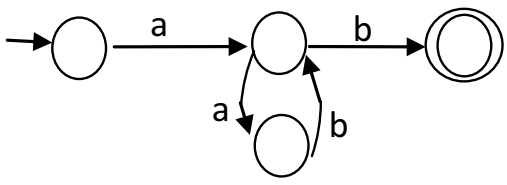


4) a*b (a|b)* ab*a (a|b)* ba*ba*b (a/b)* ab*

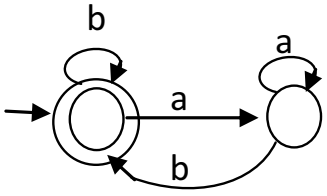


5) a(a|b)*b

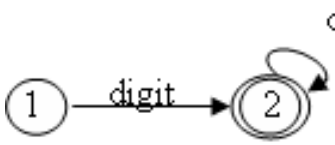
6) a(ab)*b



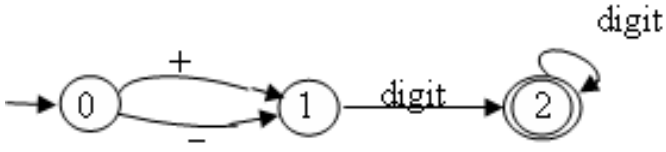
7) (b | a a* b)*



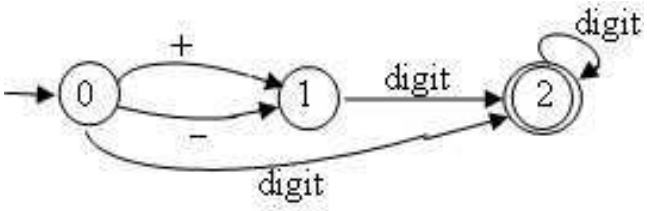
8) الأعداد الصحيحة الموجبة (Unsigned integer)



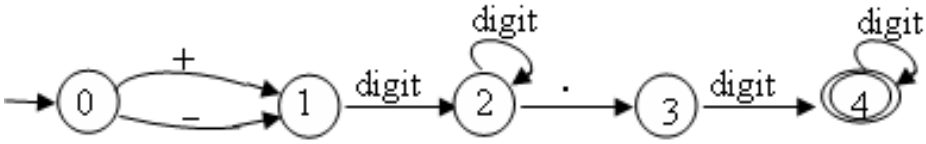
9) الأعداد الصحيحة (signed integer)



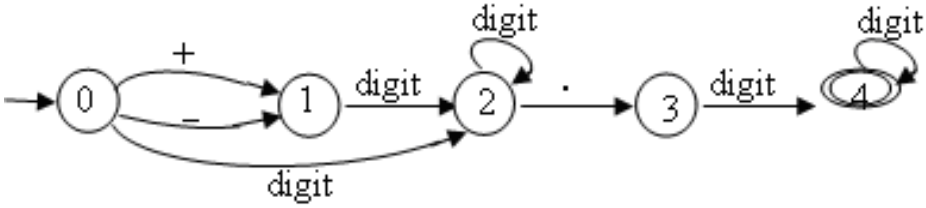
10) الأعداد الصحيحة (Signed and unsigned integer)



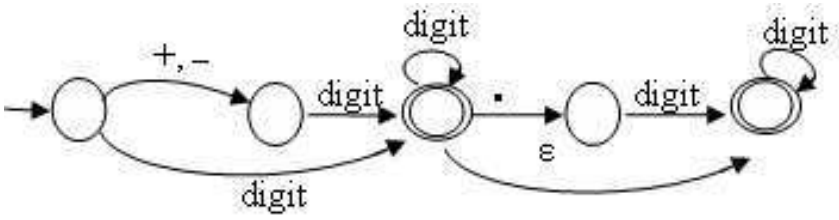
11) الأعداد الحقيقية (signed real)



12) الأعداد الحقيقية التي تحمل إشارة والتي لا تحمل إشارة



13) الأعداد الصحيحة والحقيقية (real and integer)



مثال :لو كان لدينا التعبير المنتظم التالي : RE=(a|b)* abb

على فرض أن S={a, b} فان $S^* = S^0 \cup S^1 \cup \ldots$

S*={ε , a , b , ab , ba , aaa , bbb , ….}

L = {ε , a , b , ab , ba , aaa , bbb , ….}.{abb} = {abb , aabb , ababb , baabb, aaaabb , bbbabb, … }

وهي المجموعة المنتظمة للتعبير المنتظم أعلاه .

كما يمكن تصميم تعابير منتظمة لإغراض مختلفة واليك الأمثلة التالية :

- صمم تعبير منتظم لكل الكلمات في الأبجدية {a ,b ,c} ؟

RE=(a|b|c)*

- صمم تعبير منتظم لكل الكلمات في الأبجدية {a ,b ,c} التي تبدأ بحرف a؟

RE=a(a|b|c)*

- صمم تعبير منتظم لكل الكلمات في الأبجدية {a ,b ,c} التي تبدأ بـ ba وتنتهي بـ ab ؟

RE=ba(a|b|c)* ab

- صمم تعبير منتظم لكل الكلمات في الأبجدية{a,b,c}التي تنتهي بحرفين متشابهين ؟

RE= (a|b|c)* (aa|bb|cc)

- صمم تعبير منتظم لكل الكلمات في الأبجدية {a ,b ,c} التي تبدأ أو تنتهي بحرفين متشابهين ؟

RE= (aa|bb|cc) (a|b|c)* | (a|b|c)* (aa|bb|cc)

- صمم تعبير منتظم لكل الكلمات في الأبجدية {a ,b ,c} التي تحتوي على الاقل على حرفين a؟

RE= (a|b|c)* a (a|b|c)* a (a|b|c)*

- صمم تعبير منتظم لكل الكلمات في الأبجدية {a ,b ,c} التي تحتوي بالضبط على حرفين a؟

RE = (b|c)* a (b|c)* a (b|c)*

- صمم تعبير منتظم لكل الكلمات في الأبجدية {a,b} من الـ a's والـ b's والتي بطول 2.

RE = (a|b)(a|b)

حيث ان المجموعة المنتظمة للتعبير المنتظم أعلاه هي {aa,ab,ba,bb} وهي اللغة التي تحتوي على كل الكلمات المعرّفة على الأبجدية {a,b} من الـ a's والـ b's التي بطول 2.

- صمم تعبير منتظم للمعرّفات (identifier) .

id → letter (letter| digit)*

letter → a| b| c|….| z| A| B| C|…...|Z

digit → 0|1|2|3|……|9

أما القواعد المقابلة لقواعد الإنتاج أعلاه فهي:

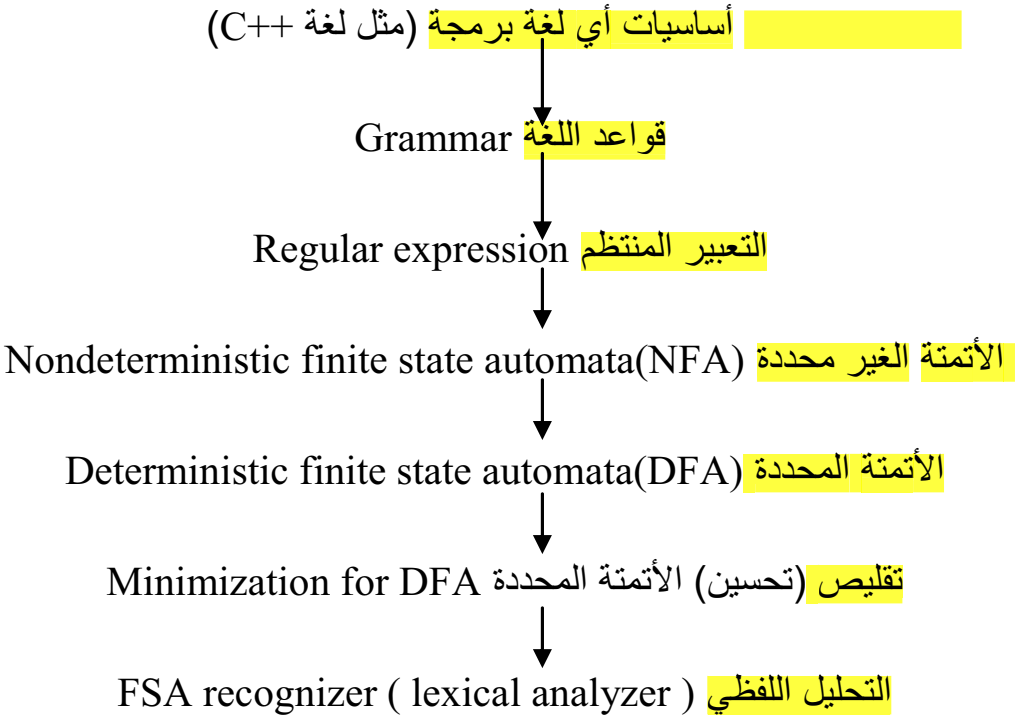
id →letter A

A→ letter A| digit A| ∈

letter→ a| b| c|….| z| A| B| C|……|Z

digit→ 0|1|2|3|……|9

من الملاحظ أن هناك تسلسل معين للدخول إلى مرحلة التحليل اللفظي (Lexical Analysis):



حيث أن لكل لغة قواعد خاصة بها وهناك طرق لوصف تلك القواعد منها التعبير المنتظم والأخير يحول إلى أتمتة غير محددة (NFA) وبالخوارزمية التي تم شرحها سابقا يتم تحويل الأخيرة إلى أتمتة محددة (DFA) وتقلص وتحسن الأتمتة المحددة إذا كان فيها حالات زائدة ليتم استخدام خوارزمية مميز الـ FSA التي من خلالها يتم معرّفة كون جملة ما مقبولة في اللغة المقابلة للقواعد التي تم وصفها مسبقا.

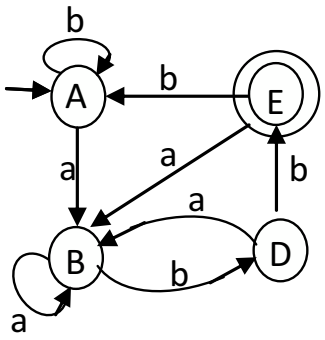
3.5 مميز الـ FSA (FSA acceptor)

وهو عبارة عن خوارزمية من خلالها يتم تشخيص والتعرف على الجملة فيما إذا كانت مقبولة في القواعد المقابلة للغة المعرّفة على تلك القواعد الموصوفة بالتعبير المنتظم، واليك الخوارزمية:

```
Begin
        State = start state of the DFA
        Symbol = first input symbol
        While input symbol not already exhausted do
                If matrix [state, symbol] < > error indicator then
                        Begin
                                State = matrix [state, symbol]
                                Symbol = next input symbol
                        End
                Else    Input is not accepted
                If state is a final state of the DFA then
                        Input is accepted
                Else    Input is not accepted
end
```

مثال3.10 : افترض التعبير المنتظم (alb)*abb ، هل أن الجملة aabaabb$ مقبولة مستخدما مميز FSA؟

إن الأتمتة المحددة المقلصة المقابلة للتعبير المنتظم أعلاه مبينة في الشكل (3.9).



شكل (3.9) أتمتة محسنة (مقلصة) للتعبير(alb)*abb

والمصفوفة التي تصف الأتمتة المحددة في الشكل (3.9) هي:

الحالات	a	b
A	B	A
B	B	D
D	B	E
E	B	A

ولتطبيق خوارزمية مميز الـ FSA على الأتمتة المحددة المقلصة الممثلة بالجدول السابق من هذا المثال نحصل على ما يلي :

الحالات	المدخلات
A	a
B	a
B	b
D	a
B	a
B	b
D	b
E	$

وكما نلاحظ من تتبع الخوارزمية نجد انه عند الانتهاء من المدخلات ووصلنا إلى علامة $ نجد انه قد وصلنا إلى الحالة E التي تعتبر حالة نهائية لذا تعتبر الجملة aabaabb$ مقبولة في المجموعة المنتظمة للقواعد الموصوفة بالتعبير المنتظم أعلاه.

وللتأكد بطريقة ثانية من أن الجملة aabaabb$ مقبولة نجد المجموعة المنتظمة للتعبير المنتظم أعلاه (a|b)*abb وهي :

L={ ε, a ,b ,aa, ab, ba, bb, aaa, aab, aba, abb, baa, bab, bba, bbb, …}.{abb}

L={abb, aabb ,babb ,aaabb, bbabb, ababb ,baabb ….}

لوجدنا أن الجملة أعلاه من ضمن الجمل في اللغة أعلاه، لذا فإنها مقبولة.

3.6 خوارزمية Aho & corasick

هناك باحثان في تصميم المترجمات أعابوا الخوارزمية السابقة لأنها بطيئة و لا تستطيع تميز الكلمات المتداخلة ولكن تستطيع تمييز الكلمات المحجوزة فقط كما أنها بطيئة لذا وجدا الخوارزمية التالية:

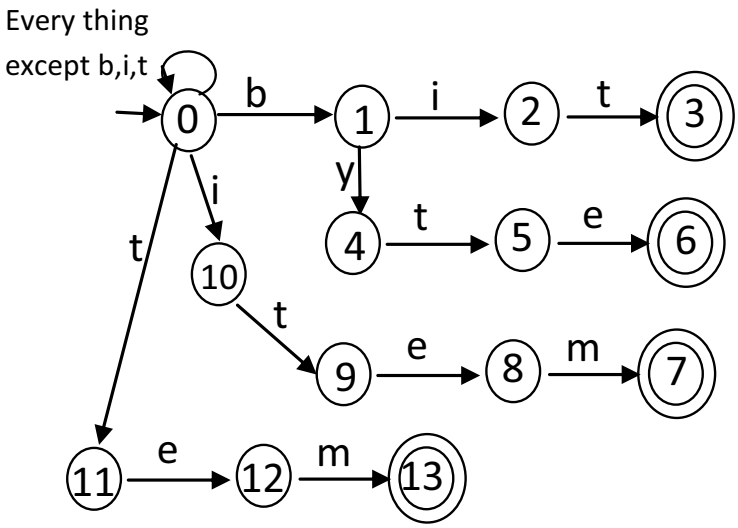
```
Begin
        State = 0;
        Ch = first char of input;
        While text not exhausted do
                If matrix [state, ch] < > error indicator then
                        Begin
                                State = matrix [state, ch];
                                Ch = next char;
```

```
            End;
        Else    Begin
                    If state is a final then signal;
                    If state = 0 then
                        Ch = next char;
                    Else
                        State = f(state);
                    End;
End;
```

مثال3.11:لو كانت لدينا اللغة التالية L={'bit' ,'byte' ,'item' , 'tem'}

ولدينا الجملة التالية :W= bitemporal$.

في البداية يتم بناء الشجرة كما في الشكل (3.10) التي كل مسار فيها يمثل كلمة من كلمات اللغة المحجوزة فالمسار (0,1,2,3) يمثل كلمة (bit) والمسار (0,1,4,5,6) يمثل كلمة (byte) والمسار (0,10,9,8,7) يمثل كلمة (item) وهكذا بالنسبة للبقية .



شكل (3.10) شجرة الكلمات المحجوزة

ثم يتم إنشاء ما يسمى بدالة الرجوع الخلفي (Fall back function) والتي تأخذ الصيغة التالية : F(Q)=R

حيث يتم حسابها بان نجد أطول طريق يوصلنا إلى العقدة Q ليس من البداية ثم نجد نفس ذلك الطريق ولكن هذه المرة من البداية يوصلنا إلى تلك العقدة كما في مثالنا:

Q	0	1	2	3	4	5	6	7	8	9	10	11	12	13
F(Q)	0	0	10	9	0	11	12	13	12	11	0	0	0	0

ففي مثالنا فان (Q=2) وان (F(Q)=10) فان أطول طريق يوصل إلى الحالة (2) ليس من البداية هو (i) ونفس هذا الطريق (i) ولكن من البداية فانه يوصل إلى الحالة (10)، وكذلك اذ كانت (Q=7) فان (F(Q)=13) فان أطول طريق يوصل إلى الحالة (7) ليس من البداية هو (tem) ونفس هذا الطريق (tem) ولكن من البداية فانه يوصل إلى الحالة (13) ، أما للحالة (1) فلا يوجد أطول طريق ليس من البداية لذا فان (F(1)=0) وهكذا بالنسبة للبقية . أما الجدول التالي فيعكس التعامل بين الحالات والمدخلات استنادا إلى الرسم أعلاه .

	b	i	t	e	y	m
0	1	10	11	0	0	0
1	#	2	#	#	4	#
2	#	#	3	#	#	#
3	#	#	#	#	#	#
4	#	#	5	#	#	#
5	#	#	#	6	#	#
6	#	#	#	#	#	#
7	#	#	#	#	#	#
8	#	#	#	#	#	7
9	#	#	#	8	#	#
10	#	#	9	#	#	#
11	#	#	#	12	#	#
12	#	#	#	#	#	13
13	#	#	#	#	#	#

ثم يتم تطبيق الخوارزمية على الجملة المراد معرّفة هل أنها مقبولة وهل تتكون من أكثر من كلمة محجوزة.

State	Ch	
0	b	
1	i	
2	t	
3	e	bit
9	e	
8	m	
7	p	item
13	p	tem
0	p	
0	o	
0	r	
0	a	
0	l	
0	$	