

عملي

4



8



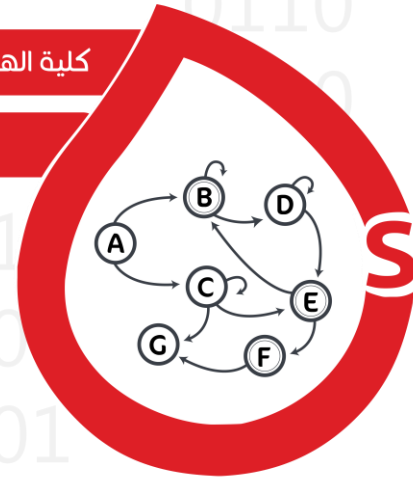
960 sp

كلية الهندسة المعلوماتية

السنة الثالثة

Reguler Expretion

عملي مشترك



RB Informatics; 23/05/2025

اللغات الصورية

السلام عليكم يا أصدقاء

أهلاً وسهلاً أصدقائنا عدنا اليكم من جديد...

بعد ما تعلمنا رسم الأوتومات اليوم سوف نتعلم العمليات عليها وشكل جديد...

مثال

ليكن لدينا النص التالي:

⁽¹⁾ ⁽²⁾ ⁽³⁾ ⁽⁴⁾ ⁽⁵⁾ ⁽⁶⁾
^[a - z 0 - 9]{3,5}@ g(oogle)? mail\ .com\$

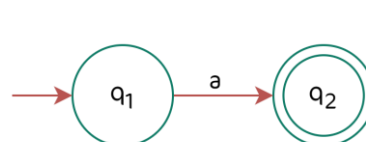
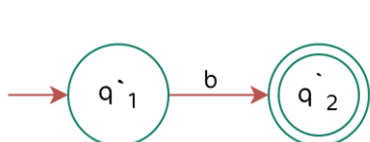
لكي يكون شكل الـ email صحيح يجب:

1. يجب أن يتضمن أحرف $a \rightarrow z$ وأرقام $0 \rightarrow 9$
2. لا يتجاوز $3 \rightarrow 5$ محارف
3. @ بعدها
4. يمكنك كتابة g فقط أو كلمة google
5. كلمة mail بعدها
6. بعدها .com

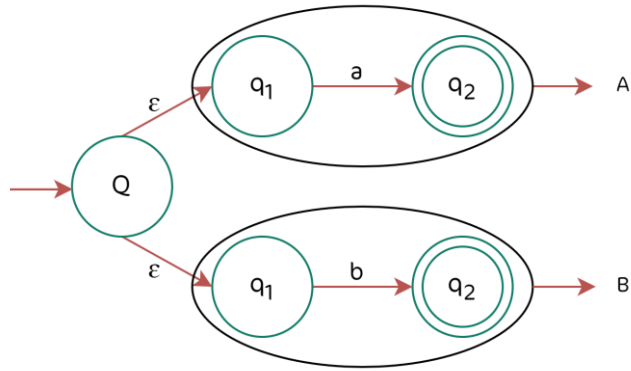
ملاحظة: إذا سبق الشرط بإشارة استفهام فهذا يعني أنه اختياري.

العمليات على الأوتومات

■ اجتماع أوتوماتين:

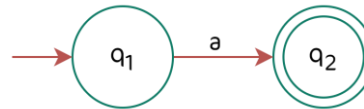
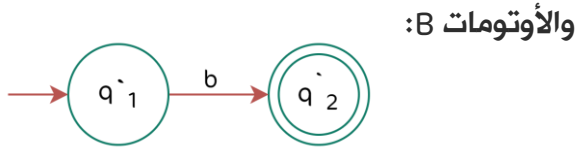


أصنع عقدة إضافية وانتقل منها ب ϵ إلى بداية الأوتومات A وبداية الأوتومات B



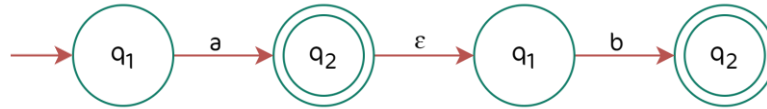
$$Q = \{q_1, q_2, q'_1, q'_2\}$$

جاء أوتوماتين:



وأريد $R = A.B$

ننتقل من نهاية الأوتومات A إلى بداية الأوتومات B ب ϵ



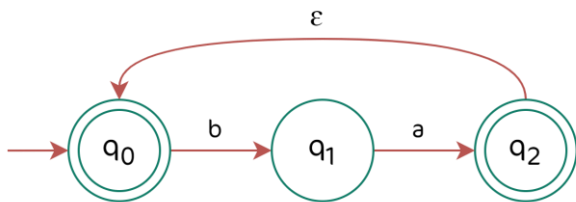
وإذا كان $R = B.A$

نضع العكس، من نهاية B ننتقل ب ϵ إلى بداية A



Keep Going!

التكرارات A^*



يجب ان يعيد الأوتومات تكرار نفسه عندما يصل إلى الحالة النهائية acceptable state وتكون الحالة الابتدائية مقبولة يعني نهائية

مثل A* لكن الحالة الابتدائية غير مقبولة غير نهائية

المعايير المنتظمة

مثلاً

$$5 + 4 = 9$$

وهذا ينطبق على اللغة

$$L_1 + L_2 = L$$

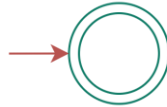
ملاحظة:

ε يوجد شيء مقبول لكن غير موجود يعني فارغ بينما ϕ لا يوجد شيء مقبول أبداً

مثال

ليكن لدينا اللغة $L = \{0, 1\}$ ارسم أوتومات مقبول به.

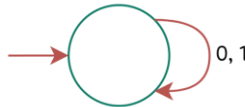
الحل:



مثال

ليكن لدينا اللغة $L = \{0, 1\}$ ارسم أوتومات يعبر عن ϕ

الحل:



أمثلة:

$$1. R = 0 + 1$$

اللغة صفر أو واحد
0, 1

$$2. R = 0^*$$

اللغة تتضمن تكرار للصفر
0, 00, 000, 0000,

$$3. R = 01^* + 10^*$$

اللغة تتضمن صفر وتكرار للواحد أو واحد وتكرار للصفر

0, 01, 011, 0111, ..., 1, 10, 100, 1000, ...

$$4. R = (0 + 1) 0^*$$

اللغة تتضمن صفر أو واحد ثم تكرار للصفر

0, 00, 10, 000, 100, ...

$$5. R = (0 + 1)^*$$

صفر أو واحد تكرارات

0101110

قواعد:

$$1. R_1 + R_2 = R_2 + R_1$$

$$2. R_1 \cdot \phi = \phi \cdot R_1 = \phi$$

$$3. \phi^* = \varepsilon$$

$$4. R_1 \cdot \varepsilon = \varepsilon \cdot R_1 = R_1$$

$$5. R_1 (R_2 + R_3) = R_1 \cdot R_2 + R_1 \cdot R_3$$

$$6. (R_1 + R_2) \cdot R_3 = R_1 \cdot R_3 + R_2 \cdot R_3$$

$$7. R_1 + R_1 = R_1$$

$$8. R_1 + \phi = R_1$$

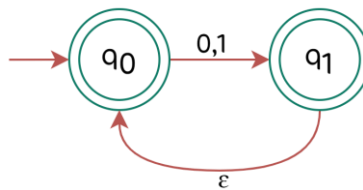
$$9. \varepsilon^* = \varepsilon$$

$$10. (R_1^*)^* = R_1^*$$

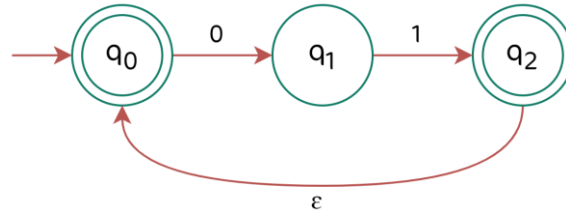


أمثلة رسم:

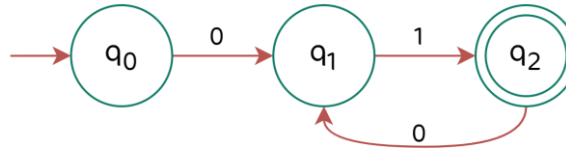
$$1. R = (0 + 1)^*$$



(+) تعني أو، أي يوجد سهم واحد

2. $(0.1)^*$ 

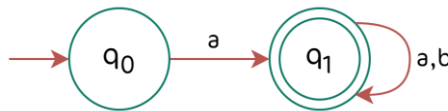
(.) تعني و، أي يوجد سهمين

3. $(0.1)^+$ 

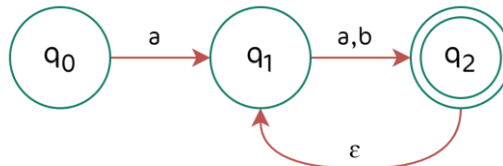
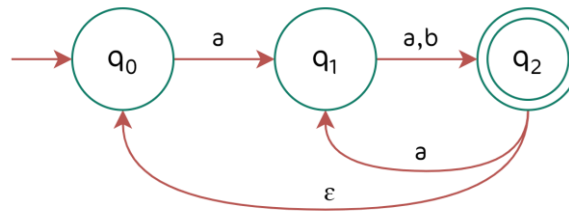
الحالة الابتدائية غير مقبولة

4. $R = a(a+b)^*$

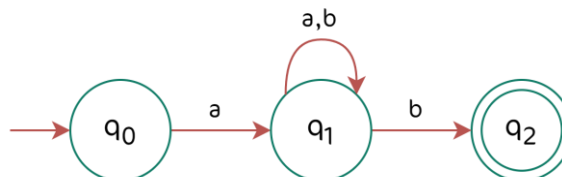
a أو b عند الحالة المقبولة و (*) يوجد تكرار

5. $R = a(a+b)^+$

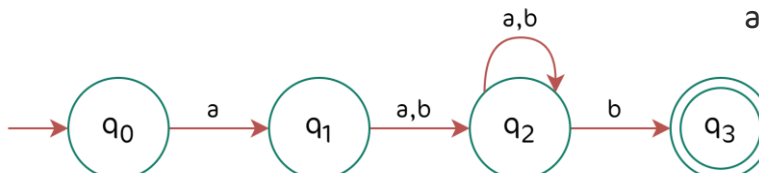
يوجد تكرار يعود الانتقال ب-ε ونضيف عقدة

6. $R = (a(a+b))^+$ تكرار لكل نعود لأول عقدة ب-ε **أو** a7. $R = a(a+b)^*b$

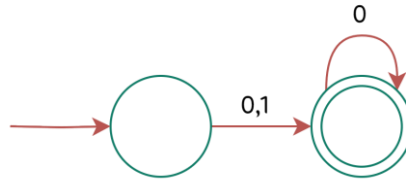
تكرار a أو b عند العقدة الثانية

8. $R = a(a+b)^+b$

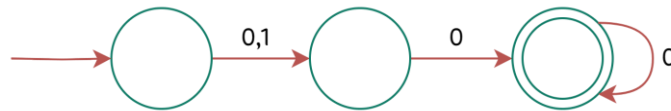
نضيف عقدة ويكون عليها تكرار a,b



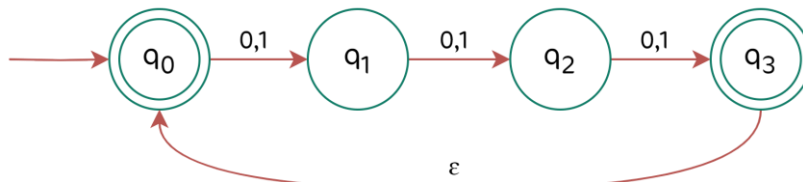
9. $R = (0 + 1) 0^*$



10. $R = (0 + 1) 0^+$

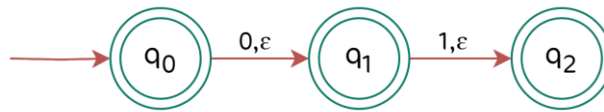


11. $R = (\varepsilon \varepsilon \varepsilon)^*$



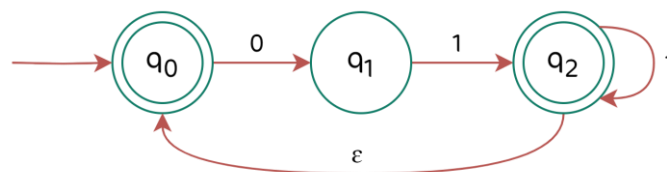
يعني تكرار ثلاث رموز
أي string قابل للقسمه
على 3

12. $R = (0 \cup \varepsilon) (1 \cup \varepsilon)$



لا يوجد تكرار

13. $R = (0, 1)^*$



الحالة الابتدائية مقبولة

Converting RE into NFA

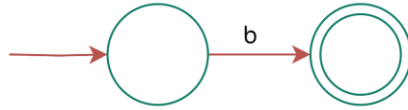
Example:

Convert $RE = (ab + a)^*$ into NFA

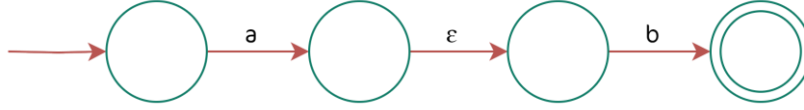
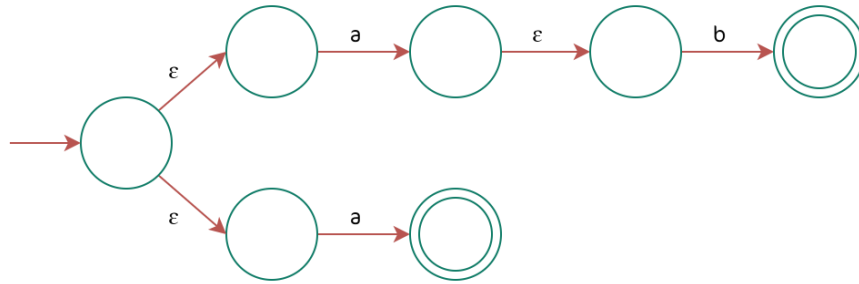
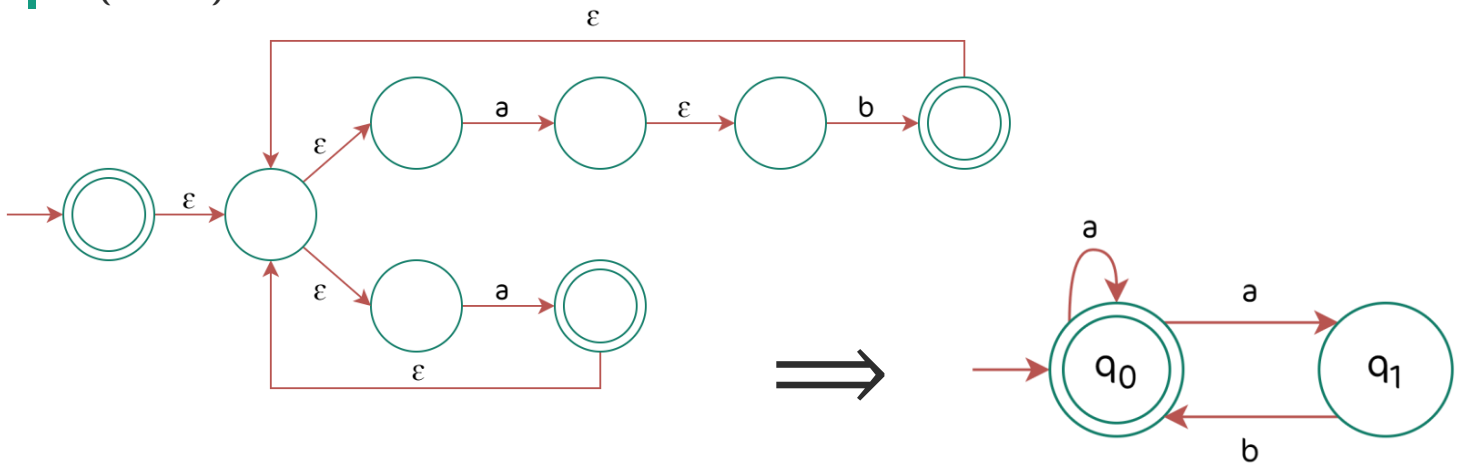
1. a



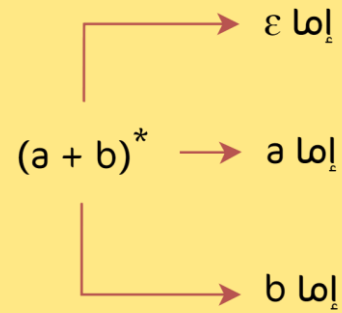
2. b



3. ab

4. $ab + a$ 5. $(ab + a)^+$ Example: $(a + b)^* aba$ $L = \{a, b\}$ يعبر عن سلسلة تبدأ بأي شيء وتنتهي حصراً بـ aba

ملاحظة:

أي شيء ينتمي للأبجدية $\{a, b\}$ **The End**