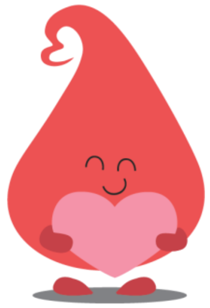
التحويل بين أنواع الأوتومات المنتهي

د. محمد الأحمد

5/4/2023
محتوى مجاني غير مخصص للبيع التجاري
RBCs Informatics;
اللغات الصورية

السلام عليكم ورحمة الله وبركاته



سنتكلم في هذه المحاضرة عن :

- تطبيقات Finite Automata .
- أمثلة إضافية عن المحاضرة السابقة
- التحويل من NFA إلى DFA .
- مقدمة عن التعابير المنتظمة Regular expressions .

تطبيقات الـ Finite Automata :

من الاستخدامات الهامة للأوتومات في الـ Software هو عمل الـ Compiler :

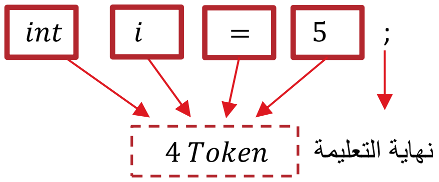
<u>Compiler</u>: هو تطبيق يقوم بتحويل النص البرمجي من لغة عالية المستوى (Source code) إلى لغة أقل مستوى (Destination code) مثل: assembly- byte code-machine code .

Source code (SC) ⟶ $\boxed{\text{Compiler}}$ ⟶ Destination code (DC)

<u>وتمر عملية تحويل النص البرمجي بعدة مراحل :</u>

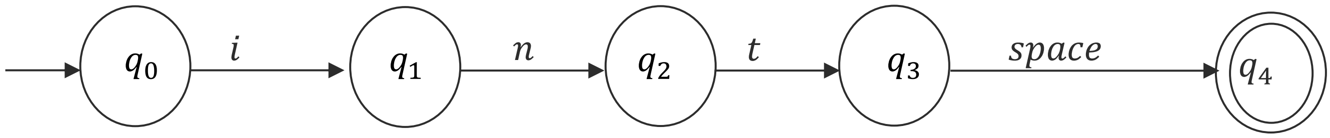
1. المرحلة الأولى (lexical analyzer) : في هذه المرحلة يتم تقسيم الـ Source code إلى مجموعة من الـtokens

مثال: ليكن لدينا التعليمة التالية : $int\ i = 5;$ ، يتم تقسيم النص الى $Tokens$ على الشكل التالي :



بحيث يتم التعرف على المتغيرات والكلمات المفتاحية في أي نص برمجي من خلال استخدام أوتومات من النوع FA .

مثال:رسم أوتومات للتعرف على الكلمة المفتاحية int :



في الحالة q_0 عندما يقرأ i ينتقل للحالة q_1 .

في الحالة q_1 عندما يقرأ n ينتقل للحالة q_2 .

في الحالة q_2 عندما يقرأ t ينتقل للحالة q_3 .

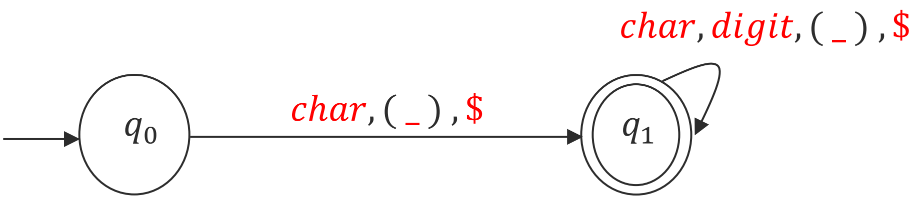
بقدر الكدِّ
تكتسب
المعالي

ثمَّ يقرأ $space$ فتكون انتهت سلسلة الإدخالات فيذهب لحالة نهائية ويتعرف الأوتومات عليها على أنها كلمة مفتاحية .

مثال: رسم أوتومات للتعرف على المتحولات :

- **نعلم أن أسماء المتحولات الصحيحة في أي لغة برمجية تبدأ بـ $\left(A \to Z \text{ أو } a \to z\right)$ أو تبدأ بـ $underscore$ (_) أو ($) وبعد هذه الرموز إذا جاء أي رقم أو محرف تكون مقبولة أو $underscore$ أو $.**

ومن خلال الأوتومات التالي يتم تحديد إذا كانت تسمية المتحولات مقبولة أم لا.



- **مثلاً عند إدخال أي محرف ينتقل لحالة نهائية ونبقى في الحالة النهائية إذا كانت بقية الإدخالات عبارة عن $digit$ أو $char$ او (_)أو($) .**

وعند انتهاء سلسلة الإدخالات عند الحالة النهائية يتم قبول تسمية المتحول .

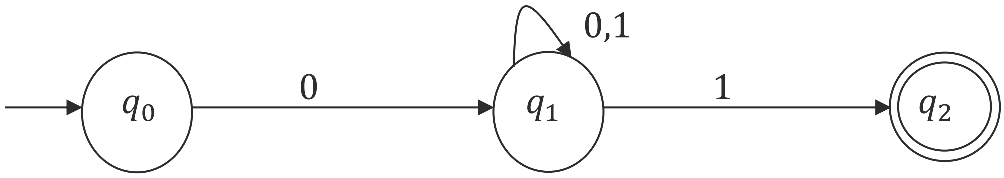
- المرحلة الثانية (parser) : في هذه المرحلة نحتاج لمعرفة فيما إذا كانت الجملة صحيحة قواعدياً أم لا $Syntax\ check$ ويتم استخدام أوتومات من النوع $push-down\ Automata$

تمارين عن المحاضرات السابقة :

التمرين الأول : صمم أوتومات FA يقبل جميع الكلمات التي تبدأ بصفر وتنتهي بواحد .

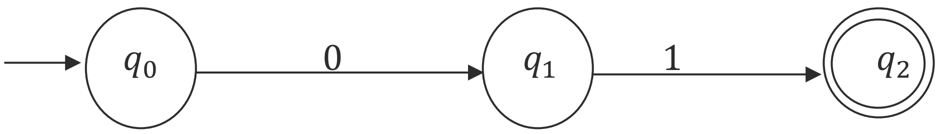
$$L=\{w: w\in 0\,(0|1)^*\,1 : \Sigma=\{0,1\}\,\}$$

<u>الحل</u> : بدايةً يجب أن نفهم أن بشرط قبول الأوتومات هو أن تبدأ السلسلة بصفر وتنتهي بواحد وفي المنتصف أيّاً كانت تشكيلة الرموز من الأبجدية فهي مقبولة فيكون NFA بالشكل التالي



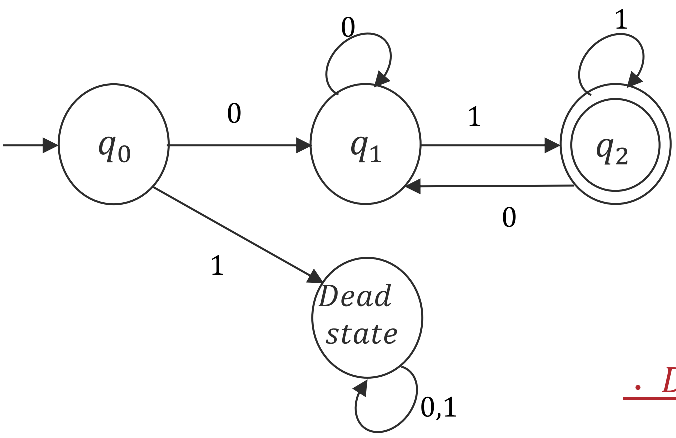
تكون السلاسل : 01، 001، 011، 00101 جميعها مقبولة .

<u>لرسم أوتومات DFA نأخذ أبسط سلسلة تحقق شرط القبول ويمكن الإنطلاق فيها وهي السلسلة 01 :</u>



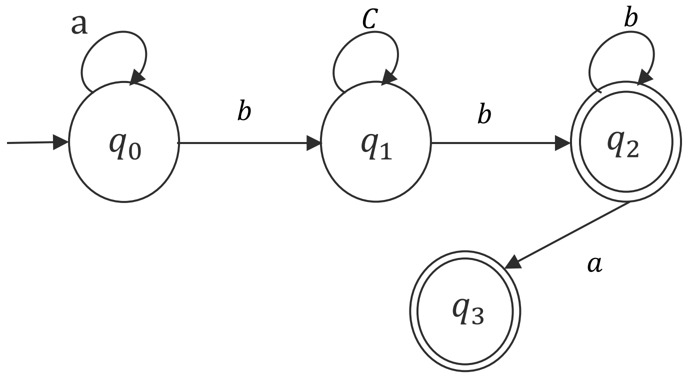
نرسم الانتقالات لبقية الرموز :

1. الحالة q_0 : إذا كان الدخل 1 إذ من الممكن أن تبدأ السلسلة بـ 1 وهي حالة غير مقبولة لذلك تذهب لحالة ميتة $dead\ state$.
2. الحالة q_1 : اذا كان الدخل 0 نبقى في الحالة نفسها q_1 (لا تذهب لحالة ميتة لأنه اذا كانت السلسلة 001 فهي مقبولة).
3. الحالة q_2: اذا كان الدخل 1 نبقى في q_2 او اذا كان الدخل 0 من الممكن ان تنتهي السلسة بصفر وهي غير مقبولة لذلك تذهب لحالة ميتة ايضاً .



<u>يجب معالجة الحالات الميتة للمحافظة على تعريف الـ DFA .</u>

تمرين ثاني : ما الكلمات التي يعرفها الأوتومات التالي :

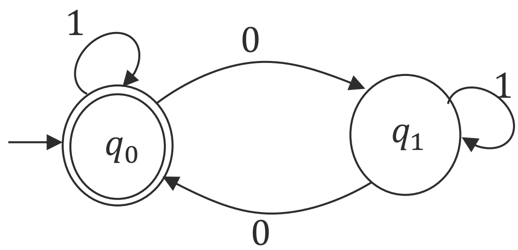


$$l(u) = \{bb, bbb, bba, abcb, bcb, bcba \ldots \ldots\}$$

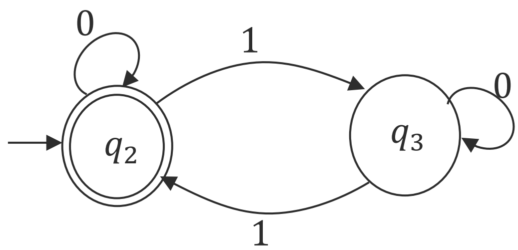
جميع الكلمات من الشكل $a^* \; b \; c^* \; b \; b^* \; a^*$ تكون مقبولة. أي يمكن أن تبدأ بأي عدد من الرمز a (قد يكون 0) و بعدها رمز b ثم أي عدد من رموز c (قد يكون 0) ثم رمز b واحد على الأقل، ولا مشكلة في حال جاء بعدها رمز a.

تمرين ثالث: صمم أتومات يقبل عدد زوجي من الأصفار أو عدد فردي من الواحدات.

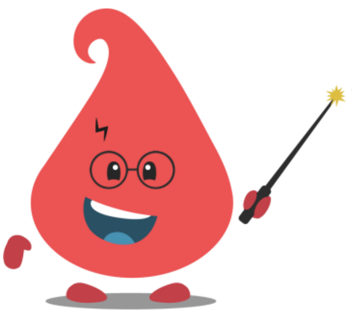
- تذكرة: في المحاضرة السابقة صممنا أوتومات يقبل عدد زوجي من الأصفار:

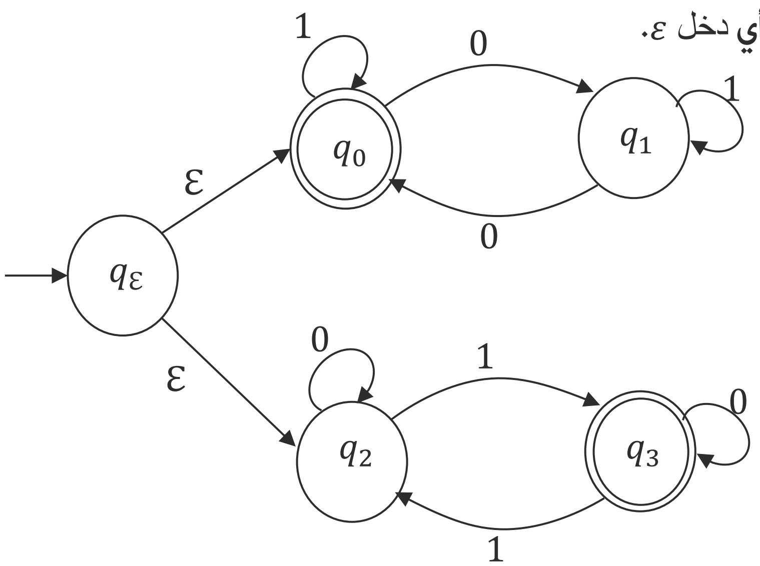


- و أوتومات يقبل بعدد فردي من الواحدات:



نقوم بدمج الأوتوماتين السابقين بأتومات جديد ونأخذ حالة ابتدائية عامة بحيث يكون الانتقال من هذه الحالة الابتدائية لباقي الحالات بدون قراءة أي دخل ε.

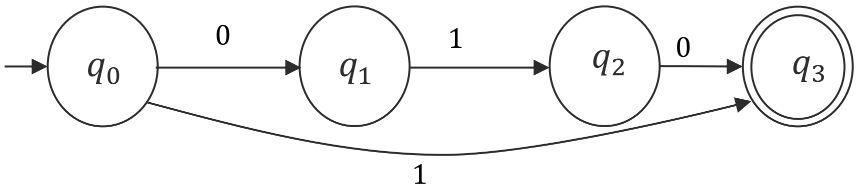




التمرين الرابع : صمم أوتومات DFA يقبل اللغة $L = \{010\ ,1\}$ حيث $\Sigma = \{0,1\}$

الحل: يجب أن نعلم أن الهدف من هذا الأوتومات هو قبول السلسلتان 010 و 1 **فقط**

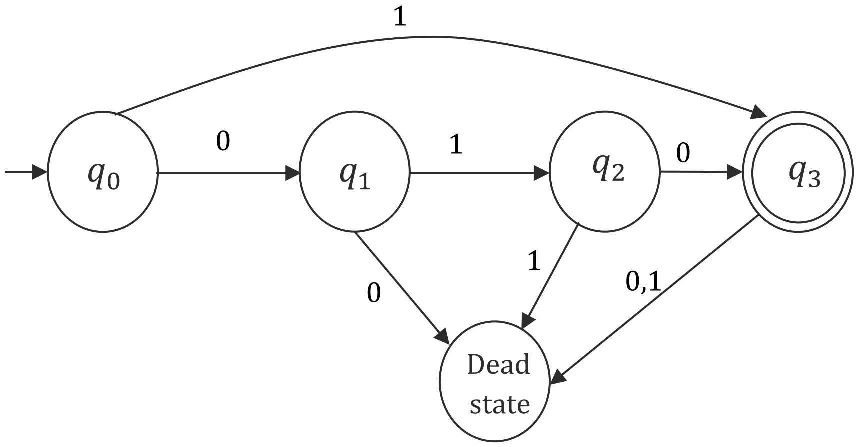
نبدأ برسم السلسلة الأولى: 010من الحالة الابتدائية q_0 عند الدخل 0نذهب للحالة q_1 ومن هذه الحالة عند الدخل 1 نذهب للحالة q_2 ومنها عند الدخل 0 ننتقل للحالة q_3 وهي حالة نهائية وتكون والسلسلة الأولى تحققت .

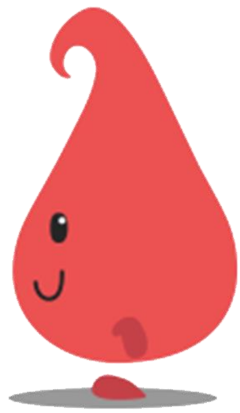


عند q_0 إذا كان الدخل 1 ننتقل للحالة النهائية q_3 ونكون بهذه الحالة قد حققنا هدف الأوتومات ولكن هذا الأوتومات ليس DFA لذلك يجب أن نكمل ونرسم بقية الانتقالات.

نلاحظ أن بقية الانتقالات عند الحالات q_1 و q_2 و q_3 تؤدي لحالة ميتة لأنه ينتج عنها سلاسل غير السلسلتان 010 و 1

يصبح الأوتومات DFA بالشكل التالي :





التحويل من NFA إلى DFA

تذكرة بتوابع الإنتقال :

$$NFA\ \delta: Q \times \Sigma \rightarrow p(Q)$$

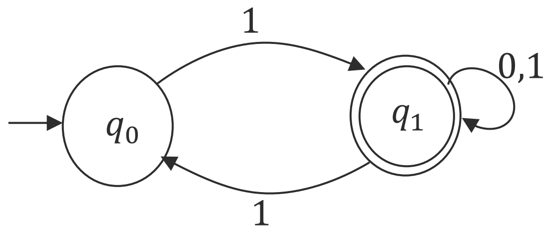
- عند كل رمز من رموز الأبجدية يمكن عدم وجود انتقال أو يوجد انتقال ولكن ليس بالضرورة أن يكون وحيد .

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

- عند كل رمز هناك انتقال وانتقال وحيد فقط ويوجد انتقالات على جميع رموز الأبجدية .

مثال لتوضيح الفرق بين NFA و DFA وحل كل مشكلة :

ليكن لدينا الأوتومات التالي :



- عند الحالة q_1 إذا كان الدخل 1 نذهب لحالتين إما q_1 أو q_0 (انتقال متعدد) وللتحويل لـ DFA يجب أن يكون الانتقال (انتقال وحيد) ولحل هذه المشكلة نتعبر الحالتين $\{q_0, q_1\}$ حالة جديدة وحيدة .
- عند الحالة q_0 ليس هناك انتقال عند الرمز 0 وللتحويل لـ DFA يجب أن يوجد انتقالات على جميع الرموز ولحل المشكلة نعتبر جميع الانتقالات الغير مذكورة في NFA إلى الحالة الميتة $\emptyset$ في DFA .
- في حال كان عدد الحالات في NFA هو n فإن عدد الحالات في DFA هو 2^n .

مثال للتوضيح : $Q_{NFA} = \{q_0, q_1\}$

$$Q_{DFA} = \{\emptyset, \{q_0\}, \{q_1\}, \{q_0, q_1\}\}$$

$\{q_0, q_1\}$: تعبر عن الحالات التي دمجت.

$\emptyset$: تعبر عن الانتقالات الغير مذكورة عند رموز معينة .

وتكون الطريقة العالمة للتعبير عن الحالات :

FA	NFA	DFA
$States$	$q_0, q_1, \dots \dots . . q_n$	$\emptyset, \{q_0\}, \{q_1\}, \dots \dots \{q_0, \dots . q_n\}$

- الانتقالات للحالات الجديدة عند دخل معين مثل الحالة $\{q_0, q_1\}$ تساوي اجتماع انتقالات الحالتين $\{q_0\}$و$\{q_1\}$عند هذا الدخل .

الطريقة العامة للتعبير عن الإنتقالات ، حيث a هي الدخل :

FA	NFA	DFA
$Transitions$	δ	$\delta(\{q_{i1}, \dots \dots \dots, q_{ik}\}, a) = \delta(q_{i1}, a) \cup \dots . \delta(q_{ik}, a)$

- الحالة الابتدائية في DFA هي نفس حالة NFA .
- الحالات النهائية في DFA هي نفس الحالات النهائية في NFA وكل مجموعة جديدة تحوي على حالة نهائية من NFA هي أيضاً حالة نهائية .

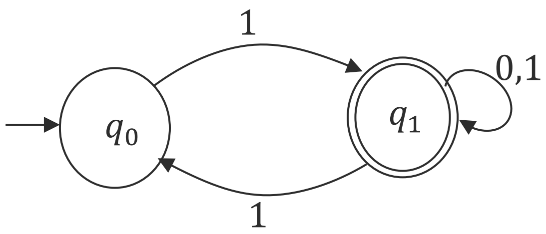
خطوات التحويل من NFA الى DFA :

1_ نوجد مجموعة الأجزاء الجزئية $P(Q)$.

2_ نوجد الإنتقالات عند جميع الحالات ونمثلها بجدول الانتقالات ونحدد الحالة الابتدائية والحالات النهائية .

3_ نرسم مخطط الانتقالات اعتماداً على الجدول .

تمرين : حول الأوتومات NFA التالي إلى الأوتومات DFA.



1. لدينا في NFA الحالات q_0 & q_1 فيكون عدد حالات DFA $2^2 = 4$.

$P(Q) = \{\emptyset, \{q_0\}, \{q_1\}, \{q_0, q_1\}\}$ مجموعة الأجزاء

state	0	1
$\emptyset$	$\emptyset$	$\emptyset$
q_0	$\emptyset$	q_1
q_1	q_1	$\{q_0, q_1\}$
$\{q_0, q_1\}$	q_1	$\{q_0, q_1\}$

2. الخطوة الثانية:

➢ الحالة$\emptyset$: بما أن هذه الحالة هي حالة ميتة

فمهما كان الدخل الوارد لهذه الحالة نبقى في الحالة نفسها

أي تكون التنقلات بالطريقة العامة كالتالي :

$$\delta(\emptyset, 0) = \emptyset, \delta(\emptyset, 1) = \emptyset$$

➢ الحالة q_0 :

1. الدخل 0 يكون الانتقال غير موجود فتكون حالة ميتة $\emptyset$
2. الدخل 1 ننتقل للحالة q_1 .

➢ الحالة q_1:

1. الدخل 0 نبقى في الحالة نفسها q_1 .

2. الدخل 1 لدينا انتقال لـ q_0 وانتقال لـ q_1 أي ننتقل لحالة جديدة وهي $\{q_0, q_1\}$.

➢ الحالة $\{q_0, q_1\}$:

1. الدخل 0 يكون الانتقال مساو اجتماع الانتقالات عند الحالتين $\{q_0\}$ و $\{q_1\}$ عند الدخل 0 :

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

2. الدخل 1 يكون الانتقال مساو اجتماع الانتقالات عند الحالتين $\{q_0\}, \{q_1\}$، عند الدخل 1 :

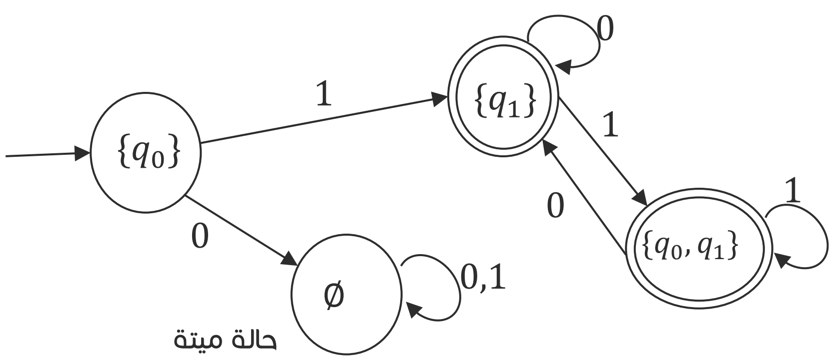
$$\delta(\{q_0, q_1\}, 1) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_1\} \cup \{q_0, q_1\} = \{q_0, q_1\}$$

➢ الحالة الإبتدائية نفسها في NFA وهي q_0

➢ والحالات النهائية هي الحالات النهائية نفسها الموجودة في NFA أي $\{q_1\}$

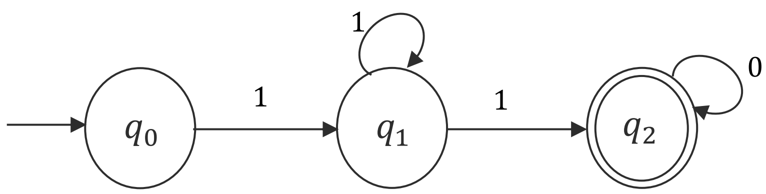
وكل مجموعة جديدة تحوي الحالة النهائية السابقة $\{q_1\}$ تكون حالة نهائية في مثالنا $\{q_0, q_1\}$.

3. **الخطوة 3:** رسم الأوتومات DFA بتتبع جدول التنقلات .



- الأوتومات DFA يكافئ الأوتومات NFA أي يقبلان نفس اللغة .
- عند التحويل من NFA إلى DFA يزداد عدد الحالات وتعقيد الأوتومات .

تمرين : حول الأوتومات التالي من NFA إلى DFA :



عدد حالات DFA : $2^3 = 8$

1. **خطوة 1 :** $P(Q) = \{\emptyset, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}\}$

2. **خطوة 2 :** ندرس التنقلات عند كل حالة ,كما تم توضيحها في المثال السابق .

مثلاً عند الحالة $\{q_0, q_1\}$ عند الدخل 0:

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

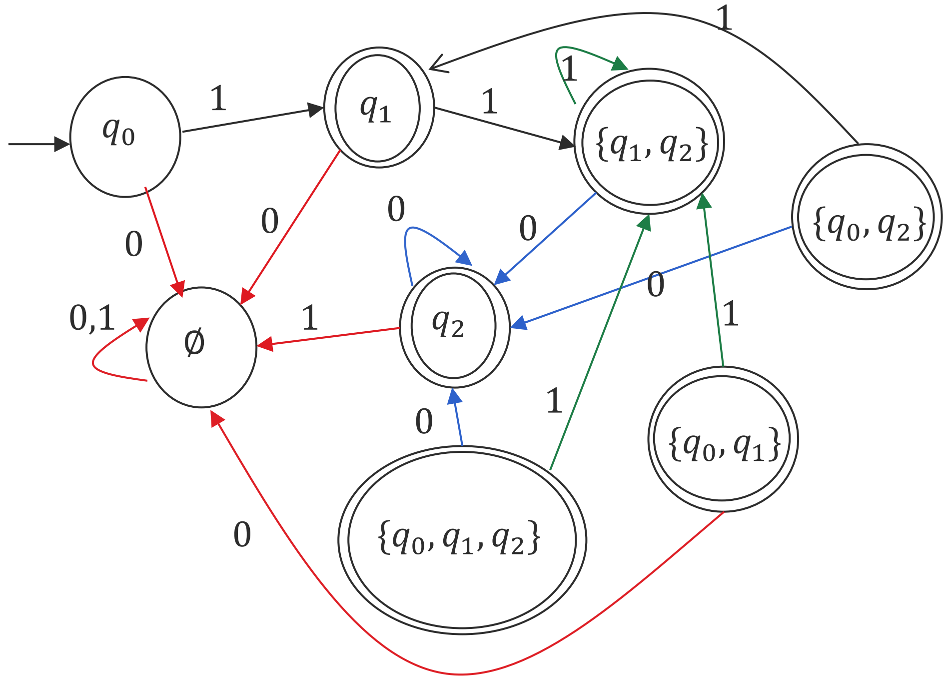
ونطبق نفس الطريقة لباقي الحالات ويمكن معرفة الانتقال فورا من الجدول .

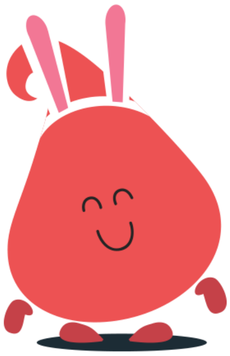
	0	**1**
$\emptyset$	$\emptyset$	$\emptyset$
$\longrightarrow \{q_0\}$	$\emptyset$	$\{q_1\}$
$*\{q_1\}$	$\emptyset$	$\{q_1, q_2\}$
$*\{q_2\}$	$\{q_2\}$	$\emptyset$
$*\{q_0, q_1\}$	$\emptyset$	$\{q_1, q_2\}$
$*\{q_0, q_2\}$	$\{q_2\}$	$\{q_1\}$
$*\{q_1, q_2\}$	$\{q_2\}$	$\{q_1, q_2\}$
$*\{q_0, q_1, q_2\}$	$\{q_2\}$	$\{q_1, q_2\}$

- الحالة الإبتدائية نفسها $\{q_0\}$

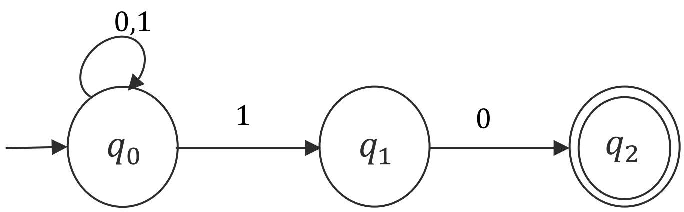
- الحالات النهائية هي $\{q_2\}$ و $\{q_1\}$ وجميع الحالات الجديدة التي تحتوي هذه الحالات النهائية تكون حالة نهائية أي $\{q_0, q_1\}, \{q_0, q_2\}, \{q_1, q_2\}$ جميعها حالات نهائية $\{q_0, q_1, q_2\}$.

3. **خطوة 3 :** رسم الأوتومات DFA :





تمرين: حول الاوتومات التالي من NFA إلى DFA:



1. الخطوة الأولى:

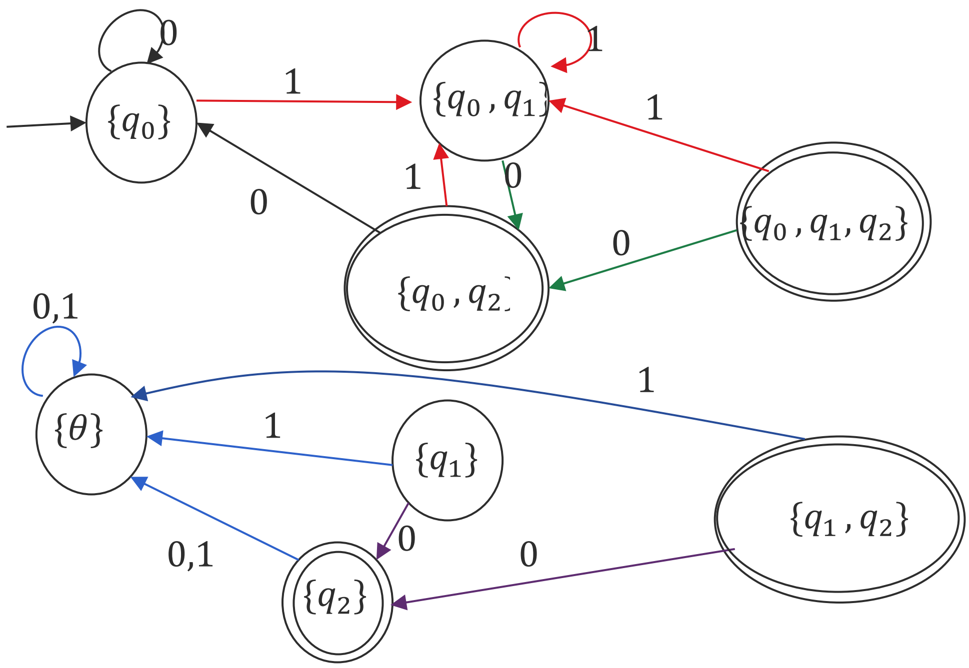
$$P(Q)=\{\emptyset,\{q_0\},\{q_1\},\{q_2\},\{q_0,q_1\},\{q_0,q_2\},\{q_1,q_2\},\{q_0,q_1,q_2\}\}$$

2. الخطوة الثانية:

ندرس التنقلات ونرسم جدول التنقلات:

	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\rightarrow\{q_0\}$	$\{q_0\}$	$\{q_0,q_1\}$
$\{q_1\}$	$\{q_2\}$	$\emptyset$
$*\{q_2\}$	$\emptyset$	$\emptyset$
$\{q_0,q_1\}$	$\{q_0,q_2\}$	$\{q_0,q_1\}$
$*\{q_0,q_2\}$	$\{q_0\}$	$\{q_0,q_1\}$
$*\{q_1,q_2\}$	$\{q_2\}$	$\emptyset$
$*\{q_0,q_1,q_2\}$	$\{q_0,q_2\}$	$\{q_0,q_1\}$

3. الخطوة الثالثة: نرسم مخطط التنقلات:



ملاحظة:

جميع الحالات $\{\theta\},\{q_1\},\{q_2\},\{q_1,q_2\},\{q_0,q_1,q_2\}$
حالات ميتة،لأن الحالات التي لا يمكن الوصول لها من الحالة الابتدائية هي حالات ميتة.

التحويل من $\mathcal{E}-NFA$ إلى NFA

تذكرة بتوابع الانتقال:

$\mathcal{E}-NFA \qquad \delta: Q \times \Sigma \cup \{\varepsilon\} \rightarrow P(Q)$

$NFA \qquad \delta: Q \times \Sigma \rightarrow P(Q)$

الاختلاف أنه في $\mathcal{E}-NFA$ لدينا انتقال وحيد على الأقل بدون مسح أي رمز $\mathcal{E}$ لذلك يجب التخلص من انتقالات $\mathcal{E}$ ونستخدم أثرها فقط.

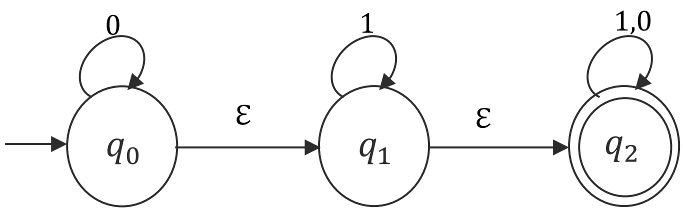
وللتحويل نحتاج لمعرفة مفهوم $\mathcal{E}-closure$

مفهوم الإغلاق $\mathcal{E}-closure(A)$ لحالة A

Set of all state can be reached without consuming any input symbols $\{\mathcal{E}\}$

أي $\mathcal{E}-closure$ لحالة A هي مجموعة جميع الحالات التي يمكن الوصول إليها من الحالة A بدون مسح أي رمز (وعلى الأقل تحوي الحالة نفسها).

مثال: ليكن لدينا المخطط التالي أوجد $\mathcal{E}-closure$ لجميع الحالات:



- من الحالة q_0 يمكن الانتقال $\mathcal{E}-closure\,\{q_0\} = \{q_0, q_1, q_2\}$ للحالتين q_1, q_2 بدون مسح أي رمز

$\mathcal{E}-closure\{q_1\} = \{q_1, q_2\}$

$\mathcal{E}-closure\{q_2\} = \{q_2\}$

ملاحظة:

$\mathcal{E}-closure$ تحوي على الأقل الحالة نفسها

خطوات التحويل من $\varepsilon - NFA \leftarrow NFA$

1. نحسب $\varepsilon - closure$ لجميع حالات $\varepsilon - NFA$
2. نرسم جدول الانتقالات ويتم تعبئة الانتقالات عند كل حالة في الجدول اعتمادا على الخوارزمية التالية

ε^* $transitions$ ε^*

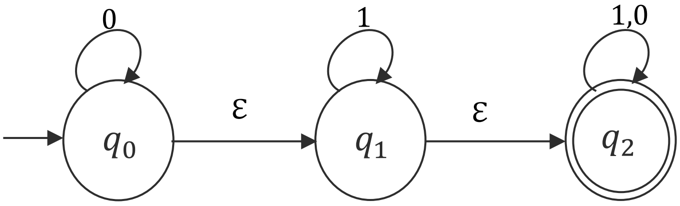
المرحلة c المرحلة B المرحلة A

- المرحلة A: نأخذ الإغلاق عند حالة معينة $\varepsilon - closure(q)$ فينتج مجموعة من الحالات أو الحالة نفسها على الأقل.
- المرحلة B: توجد انتقالات للحالات السابقة (التي نتجت عن الإغلاق في الحالة A) عند رموز الأبجدية.
- المرحلة C: نأخذ اجتماع الإغلاقات لجميع الحالات السابقة(التي نتجت عن التنقلات في المرحلة B).

> - تكون الحالة الابتدائية في NFA هي نفس حالة $\varepsilon - NFA$
> - الحالة النهائية في NFA هي نفس الحالة النهائية في $\varepsilon - NFA$ وكل حالة يمكن من خلالها الوصول للحالات النهائية في $\varepsilon - NFA$ بدون مسح أي رمز تكون حالة نهائية.

3. نرسم مخطط الانتقالات اعتماداً على الجدول.

مثال: حول الاوتومات التالي من $\varepsilon - NFA$ إلى NFA



1. نوجد $\varepsilon - closure$ للحالات q_0, q_1, q_2

$$\varepsilon - closure\{q_0\} = \{q_0, q_1, q_2\}$$
$$\varepsilon - closure\{q_1\} = \{q_1, q_2\}$$
$$\varepsilon - closure\{q_2\} = \{q_2\}$$

2. تعبئة جدول التنقلات عند كل حالة حسب الخوارزمية ε^* $transition$ ε^*

(C) (B) (A)

نبدأ بالحالة q_0:

- المرحلة A : الإغلاق عند الحالة q_0 تم إيجاده مسبقا.

$$\mathcal{E}-closure\{q_0\}=\{q_0,q_1,q_2\}$$

- المرحلة B: نوجد التنقلات عند كل دخل للحالات الناتجة عن الإغلاق (في المرحلة A) وهي q_0 & q_1 & q_2

عند الدخل 0:

- في الحالة q_0 نبقى في الحالة q_0
- في الحالة q_1 لا يوجد انتقال $\emptyset$
- في الحالة q_2 نبقى في q_2

الحالات الناتجة عن التنقلات عند الدخل q_0, q_2

- المرحلة C: نأخذ اجتماع الإغلاقات الناتجة عن المرحلة B وهي q_0, q_2

$$\mathcal{E}-closure\{q_0\}\cup\mathcal{E}-closure\{q_2\}$$
$$=\{q_0,q_1,q_2\}\cup\{q_2\}=\{\mathbf{q_0,q_1,q_2}\}$$

كيف تتم تعبئة الجدول؟

عند الدخل 1 :

- في الحالة q_0 لا يوجد انتقال $\emptyset$
- في الحالة q_1 نبقى في q_1
- في الحالة q_2 نبقى في q_2

	0	1
$\rightarrow * q_0$	$\{q_0,q_1,q_2\}$	$\{q_1,q_2\}$
$* q_1$	$\{q_2\}$	$\{q_1,q_2\}$
$* q_2$	$\{q_2\}$	$\{q_2\}$

فتكون الحالات الناتجة عن التنقلات هي $q_1 \& q_2$

- المرحلة C:نأخذ اجتماع الإغلاق للحالتان q_1 & q_2

$$\mathcal{E}-closure\{q_1\}\cup\mathcal{E}-closure\{q_2\}$$
$$=\{q_1,q_2\}\cup\{q_2\}=\{\mathbf{q_1,q_2}\}$$

الحالة q_1 :

- المرحلة A: $\mathcal{E}-closure\{q_1\}=\{q_1,q_2\}$
- المرحلة B: التنقلات للحالات الناتجة عن الإغلاق في المرحلة A أي الحالتان $q_1 \& q_2$

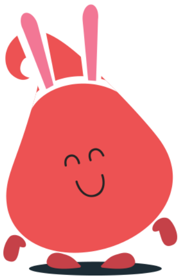
عند الدخل 0 :

- في الحالة q_1 لا يوجد انتقال $\emptyset$
- في الحالة q_2 نبقى في q_2

الحالة الناتجة هي q_2

المرحلة C: يوجد لدينا فقط حالة وحيدة من المرحلة B وهي q_2 نوجد الإغلاق لها وتم حسابه مسبقا.

$$\mathcal{E} - closure\{q_2\} = \{\boldsymbol{q_2}\}$$



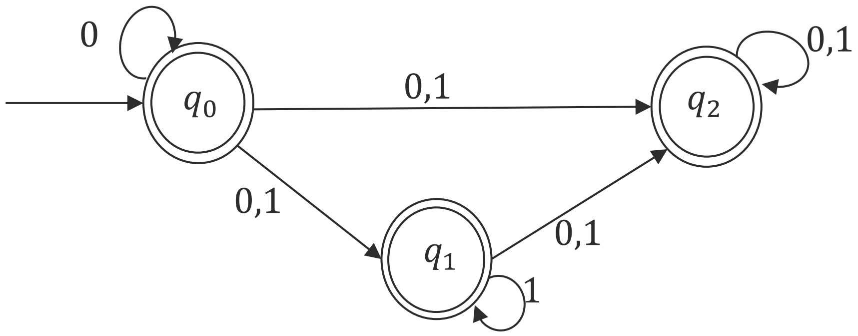
وهكذا نكمل دراسة بقية التنقلات للحالات q_1, q_2

- الحالة الابتدائية نفسها وهي q_0
- الحالة النهائية نفسها q_2 وبما أنه يمكن الانتقال من الحالتين q_1, q_0 للحالة النهائية q_2 بدون مسح أي رمز فهم أيضا حالات نهائية.

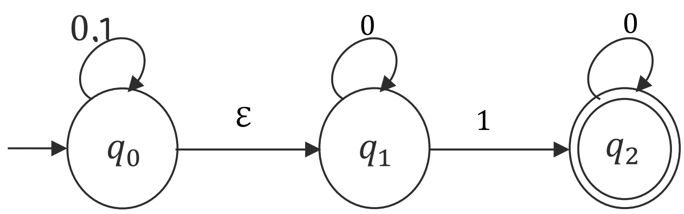
3. رسم مخطط الانتقالات:

عند التحويل من $\mathcal{E} - NFA$ إلى NFA لا يزداد عدد الحالات

أي في الحالة q_0 عند الدخل 0 ننتقل للحالة q_2 أو q_1 أو q_0 لا يتم التعامل معهم على أنهم حالة وحيدة عند الرسم.



مثال 2:



نتبع نفس خطوات التمرين السابق:

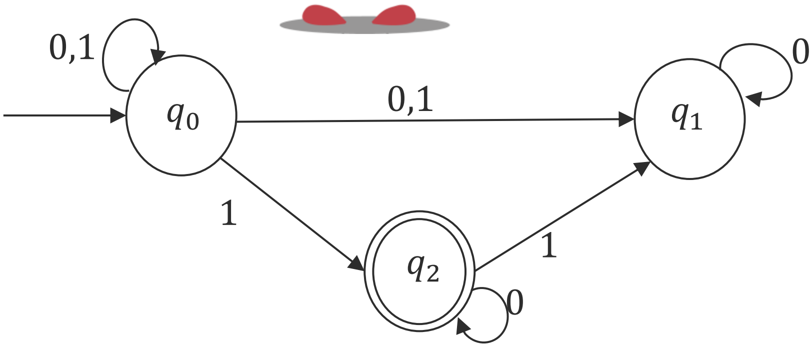
4. $\mathcal{E} - closure\ \{q_0\} = \{q_0, q_1\}$
$\mathcal{E} - closure\{q_1\} = \{q_1\}$
$\mathcal{E} - closure\{q_2\} = \{q_2\}$

3. ندرس التنقلات عند $\mathcal{E} - closure$ لكل حالة ونرسم الجدول:

	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$
q_1	$\{q_1\}$	$\{q_2\}$
$* \; q_2$	$\{q_2\}$	$\emptyset$

فقط q_2 هي الحالة النهائية لأنه لا يمكن من q_1 أو q_0 الوصول لها بدون مسح أي رمز.

4. نرسم أوتومات NFA:

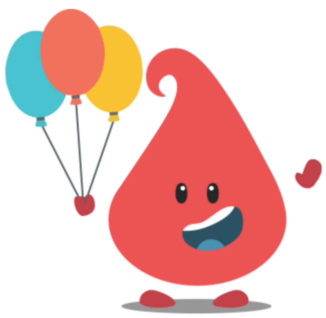


بما أن الأوتومات NFA فلا نرسم الحالة $\emptyset$

التعابير المنتظمة Regular expression (RE)

Regular expression (RE) = تكافؤ Finite automata (FA)

Regular language (RL)



✓ يتم التعرف على اللغات الطبيعية RL من خلال FA machine و Regular expression تعطي طريقة جديدة للتعرف على اللغات الطبيعية حيث يكون التعبير المنتظم عبارة عن ترميز مضغوط يمثل لغة كاملة مضمن String.

مثلا : يمكننا التعبير عن مجموعة التسميات المقبولة للمتحول في لغة برمجة ما كما يلي:

$$(a-z \,||\, A-Z \,||\, \_)(\, a-z \,||\, A-Z \,||\, \_ \,||\, 0-9\,)$$

التعبير المنتظم السابق يعني أن المتحول يجب أن يبدأ بأحد الحروف الأبجدية سواء كان كبيرا أو صغيرا أو ب $underscore(\_)$ وثم يمكن وليس إجباريا ان يأتي بعد المحرف الاول محارف أخرى أو أرقام

▪ **ملاحظة:**
تستخدم RE في محركات البحث من أجل المطابقة.

التعريف الرياضي Regular Expression(RE) يتحدد بعدة قواعد:

نعرف بشكل عودي كما يلي:

1. $\emptyset$ هي تعبير منتظم يعبر عن اللغة التي لا تحتوي أي كلمة $L = \{\,\}$
2. Ɛ هو تعبير منتظم يعبر عن اللغة التي تحتوي فقط Ɛ على كلمة خالية $L = \{\mathcal{E}\}$
3. مهما يكن a من الأبجدية $\sum$ فإن $\{a\}$ هو تعبير نظامي
   $$\forall\, a \;\in \Sigma : a \to \{a\}$$
4. مهما يكن s و r تعبيران نظاميان فإن:
   $$r^*\,,\quad r+s\,,\quad r.s$$

تعابير نظامية أيضاَ أي حالة يمكن اشتقاقها من الحالات السابقة تعتبر تعبير نظامي.

حيث $r.s$ تعني عملية وصل r بـ s ، و $r+s$ تعني or، و r^* تعني مُكرِّرات r.

▪ **ملاحظة:**
يمكن أن نعبر عن العملية $(r+s)$ بالرمز $(r|s)$ أيضاَ.

الإغلاق Kleene closure: و تعني صفر أو أكثر من التكرارات

$$a^* = \{\mathcal{E}, a, aa, aaa, \ldots..\}$$

ظهور a ولا مرة Ɛ أو ظهورها لمرة أو أكثر

$$a.a^* = a, aa, aaa, \ldots \ldots. = a^+$$

ظهور a لمرة واحدة على الأقل.

تذكرة:

$$a\mathcal{E} = a$$

لأن ε عنصر حيادي في عملية الوصل

أمثلة عن التعابير المنتظمة:

- **المثال الأول:**

$$(01)^* = \mathcal{E}, 01,0101,010101, \ldots \ldots \ldots$$

هو عبارة عن تكرارت 01 ولا مرة $\mathcal{E}$ أو لمرة أو لأكثر من مرة.

- **المثال الثاني:**

ماهو التعبير النظامي لجميع الكلمات والتي المحرف الثالث في الكلمة هو 1 .

$$(0+1)(0+1)1(0+1)^*$$

- **المحرف الأول إما 1 أو 0**
- **المحرف الثاني إما 1 أو 0**
- **المحرف الثالث واحد حصراً**
- **امحرف الرابع إما $\mathcal{E}$ أو 1 أو 0 أو أي تكرار منهما.**

- **المثال الثالث:**

ماهو التعبير النظامي لجميع الكلمات التي تحوي عدد فردي من الواحدات:

إما 1 أو 111 أو 11111 أو ... ...

$$(\underbrace{0^*\,1\,0^*\,1\,0^*}) ^*\, 0^*\,1\,0^*$$

هذه السلسلة تتكرر إما ولا مرة $\mathcal{E}$ أو مرة أو أكثر.

- **المثال الرابع:**

ماهو التعبير النظامي لجميع الكلمات التي تحوي محرف 1 لمرة واحد فقط.

$$0^*10^*$$

مع الوقت ستدرك أن الانشغال بعملك وتفاصيل يومك نجاة،
وأن الأيام المملوءة نعمة، مهما كانت ثقيلة وصعبة، وأن
وجود رفقة من حولك تعينك على أمر دُنياك أمان وسكينة..