مقدمة في الأوتومات

م. محمد تقالة

اللغات الصورية

مقدمة

الأوتومات: هي آلات (أجهزة) مجردة تقدم نمذجة للواقع أي تقوم بتقديم model والذي يقوم بتبسيط الواقع بهدف فهم المشكلة،فمثلاً تقوم بعمليات على اللغات المدخلة لتحدد فيما إذا كانت مجموعة من الأحرف أو الكلمات تنتمي لهذه اللغة أو لا تنتمي لها عن طريق نمذجة قواعد هذه اللغة كي نستطيع تمثيلها بلغة نستطيع التعامل معها.

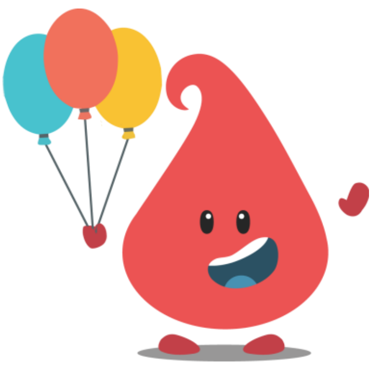
- كل لغة لها أبجدية (محارف)، ولها قواعد، ويؤدي تسلسل المحارف إلى تشكيل كلمة.

إما أن يكون لهذه الكلمة معنى في هذه اللغة فنقول أن هذه الأحرف شكلت كلمة ضمن اللغة، مثال: قالت

أو أن تكون هذه الكلمة ليس لها معنى في هذه اللغة فنقول أنها مجرد أحرف لا يمكن تفسيرها، مثال: الت

- تصنيف اللغات الصورية Formal Languages: (من الأبسط إلى الأعقد)

1. اللغات المنتظمة Regular Languages (RL): أبسط أنواع اللغات وأسهلها
2. اللغات خارج السياق Context-Free-Languages(CFL): لغات تحمل الكلمة فيها المعنى ذاته(معنىً ثابت) بأي سياق وضعت فيه، مثل: if في اللغات البرمجية والتي تحمل معنى واحد دائماً
3. اللغات السياقية Context Languages(CL): وهي اللغات الطبيعية المحكية، فيختلف فيها معنى الكلمة بحسب موقعها من الجملة (فاعل، مفعول به، ...) أي بحسب سياقها مثل: العربية، الانكليزية ..
4. اللغات القابلة للعدّ عودياً Enumerated Recursively Languages



تعاريف ومصطلحات:

1. الأبجدية Alphapet: يرمز لها ب Σ

وهي عبارة عن مجموعة منتهية من الرموز مثل:أبجدية النظام الثنائي $\sum=\{0,1\}$

2. الكلمة \ السلسلة word/string: يرمز لها ب w

هي سلسة من رموز الأبجدية Σ مثل: 01101 ويُحدد فيما إذا كانت تنتمي ل اللغة او لا بناءً على قواعد هذه اللغة.

3. الجمل:

بالنسبة للغات الطبيعية فهي تحوي جمل أما اللغات البسيطة لا تحتوي.

اللغات المنتظمة(البسيطة)

نعبر عنها بخماسية: $(\Sigma , Q , q_0 , F , \delta)$

1. Σ: أبجدية اللغة (محارفها) مثل: العربية: (أ، ب، ... ، ي) ، الانكليزية: $(A\ to\ Z)$ ، لغة الـassembly: (0,1)
2. Q: حالات الأوتومات المنتهية .
3. q_0: الحالة الابتدائية التي يبدأ عندها الأوتومات، عددها واحد حصراً.
4. F: الحالات النهائية ممكن واحدة أو أكثر حسب المسألة.
5. δ: تابع الانتقال: $\delta: Q \times \Sigma \to Q$

وهذا يعني إذا كنا في حالة محددة ثم أضفنا محرف انتقلنا إلى حالة جديدة.

- مثال 1 : لدينا لغة $L = \{ a^n\, b^m : n > 0 , m \geq 0 \}$ (لغة فيها المحرفين a, b) أنشئ أوتومات ل اللغة، كلماتها :تكرار a بعدد n مرة ويأتي بعده b مكرر m مرة بوجود شرطين $n > 0$ و $m \geq 0$

- فرضاً: هل الكلمات التالية تنتمي ل اللغة؟

ab ← نعم تنتمي، aa ← نعم تنتمي، bb ← لا تنتمي

لماذا السلسلة bb لا تنتمي إلى هذه اللغة؟

بسبب وجود الشرط $n > 0$ اي ان المحرف a لا يمكن ان يكون معدوماً.

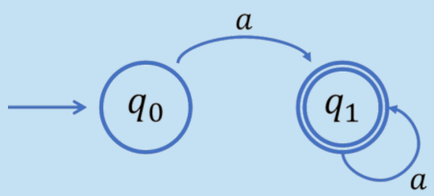
أي ببساطة يلزمنا أوتومات يحتوي على الأقل انتقال ل a، وعدة انتقالات أو ولا انتقال ل b .

- (مثال توضيحي):لو كان الشرط $n > 0 , m > 0$ يصبح لدينا :

abaa ← هذه السلسلة لا تنتمي (يجب أن تبدأ بـa وتنتهي بـ b) ، aab ← نعم تنتمي، aa ← لا تنتمي.

- التمثيل بالرسم

- أبجدية اللغة المعطاة هي عبارة عن الدخلين a و b ،وكل من الدخلين يمثل انتقالاً.
- الحالة الابتدائية q_0 نرمز لها بالسهم (→) و كل دائرة تمثل حالة Q



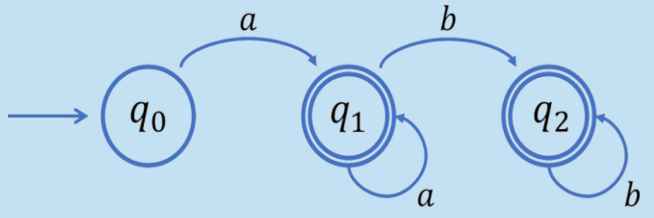
لنبدأ ب a: بفرض السلسلة تحوي a فقط.

- يتم الانتقال من q_0 إلى q_1 إذا جاء a ونعين q_1 حالة نهائية
- نرمز بدائرتين للحالة النهائية والتي هي في مثالنا هنا q_1.

وبفرض السلسلة مكونة من aa أو عدة تكرارات من a مثلاً ..aaa

- عند الدخل الأول سننتقل من q_0 إلى q_1 وعند الدخل الثاني نضع سهم عائد من q_1 إلى نفسها ونعينها كحالة نهائية.

أما عند السلسل ab :



- فسنحتاج لإنشاء حالة جديدة من أجل الانتقال b ولتكن q_2 ونعينها كحالة نهائية.

وبفرض وجود تكرار من b مثلاً abb :

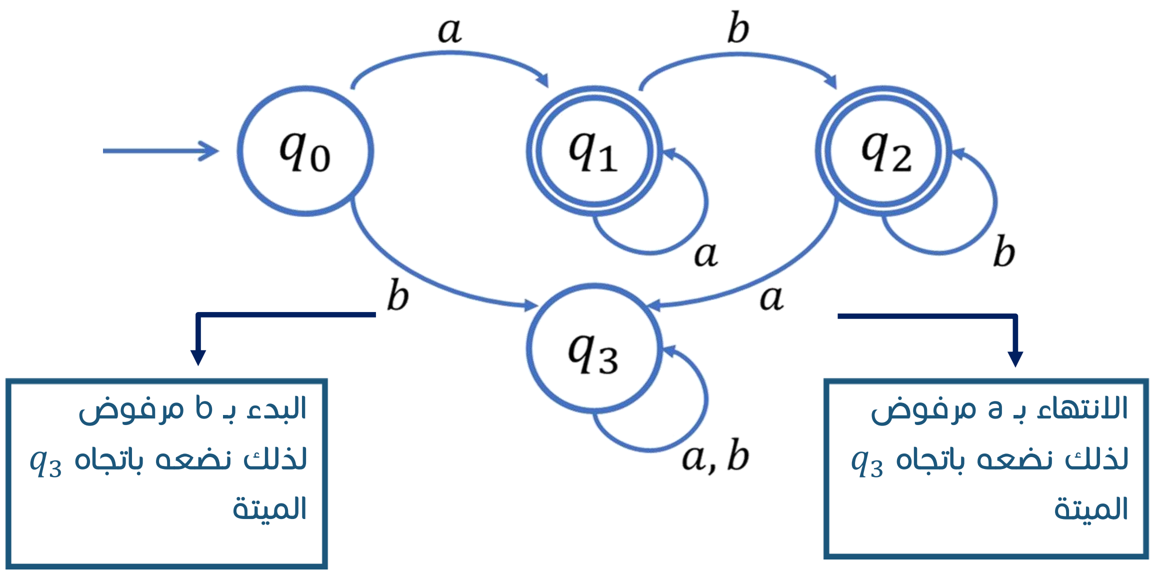
- نرسم سهماً عائداً من q_2 إلى نفسها لتمثيل التكرارات ونعينها كحالة نهائية.

الهدف من الاوتومات هو الاختصار قدر المستطاع

يوجد ثلاث أنواع للغات البسيطة: $\varepsilon - \mathrm{NFA}$ NFA DFA

- DFA: يعني أن الأوتومات منتهي حتمي أي بلحظة معينة إذا جاء حرف معين هناك حالة واحدة فقط سينتقل إليها (كما في المثال السابق) ⇐ كل حالة تحتاج محرف أبجدي كي تنتقل إلى حالة جديدة.
- NFA: المحرف الواحد ينتقل إلى حالة واحدة أو أكثر
- $\varepsilon - \mathrm{NFA}$: يتم الانتقال لحالة جديدة بدون محرف ،سيتم شرحها أكثر فيما بعد.

الرسمة السابقة لم تنتهي بعد لأننا عالجنا الحالات التي تنتمي للغة فقط وبقي الحالات التي لا تنتمي للغة (يجب تعريفها) ⇐ يجب معالجة كل الحالات.



- **الشرح:**
- بما أننا عالجنا حالة q_0 مع a، يجب معالجة حالة q_0 مع b (أي يجب معالجة كل حالة مع كل من المحرفين a,b) لأن النوع هو DFA.
- ننشئ حالة جديدة q_3 بسبب الانتقال b وتسمى بالحالة الميتة لعدم وجود انتقال يبدأ منها.
- لا نُنشئ حالة جديدة عند الانتقال a من q_2 لأنه يصح جعل الانتقال a إلى q_3 (لا فائدة من إنشاء حالة جديدة ولتكن q_4).
- q_3 إذا جاءها a أو b تبقى نفسها لأن عندها اللغة ميتة (وضع أي محرف عندها لا يوصل لكلمة نظامية)

ملاحظات:

1. a^* : وتعني تكرار a إما a غير موجود أو موجود a مرة واحدة أو موجود عدد لا نهائي من a (من 0 إلى اللا نهاية)
2. a^+ : تكرار a على الأقل مرة واحدة. $a^+ = a \, . \, a^*$

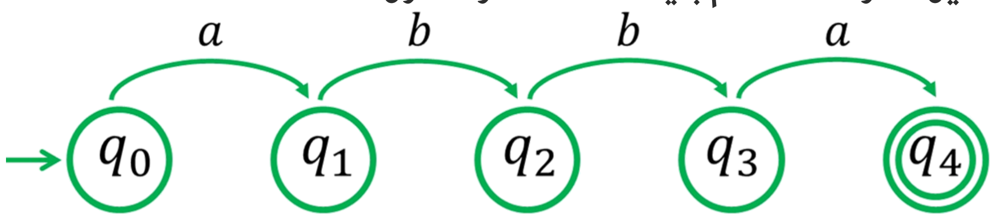
- مثال 2 : لدينا لغة $L = \{ W : W \in \{a, b\}^* \text{ and contain abba } \}$

- نعرف أوتومات لغة فيها عدد المحارف $\{a, b\}^*$ لا على التعيين.

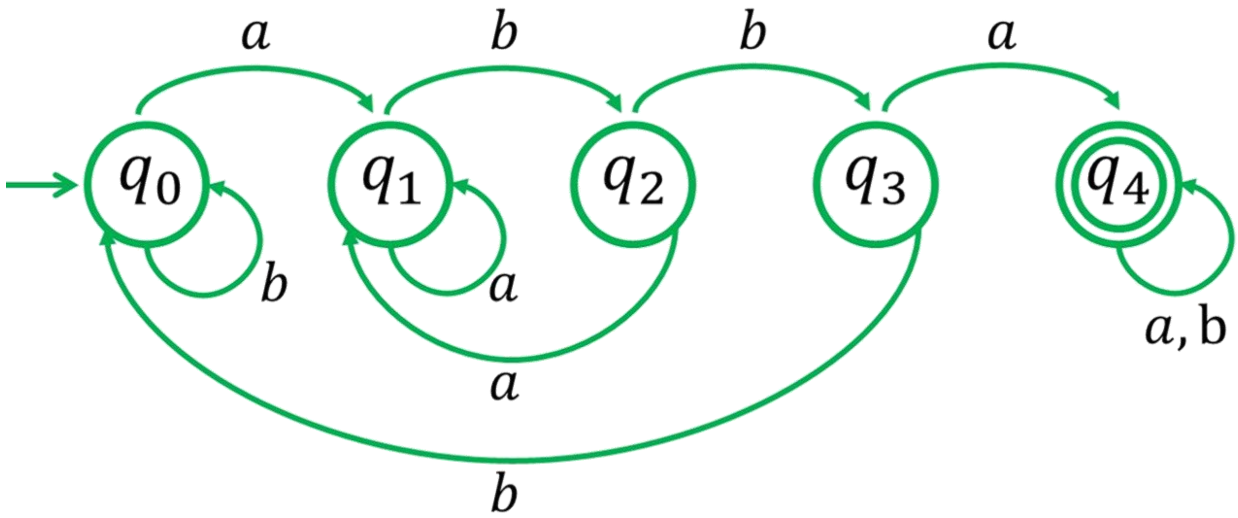
$\{a, b\}^*$ تعني أن كل الحالات مقبولة، ولكن هناك شرط مضاف هو أن تحوي الكلمة على abba ولا يهم مكانها ولكن إن وجد start بدلاً من contain في تعريف اللغة فيجب حصراً أن تأتي abba بداية الكلمة

-من الممكن وجود أكثر من حل أو تمثيل للمسألة. (المهم تمثيل كل الحالات)

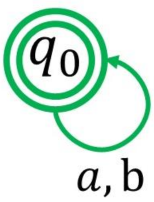
- سنقوم بتمثيل الشرط abba ثم بقية الحالات (الشرط الأول)



- هكذا نكون حققنا شرط احتواء الأوتومات على abba وبعدها سنقوم بمعالجة الحالات المتبقية، مع مراعاة الشرط السابق



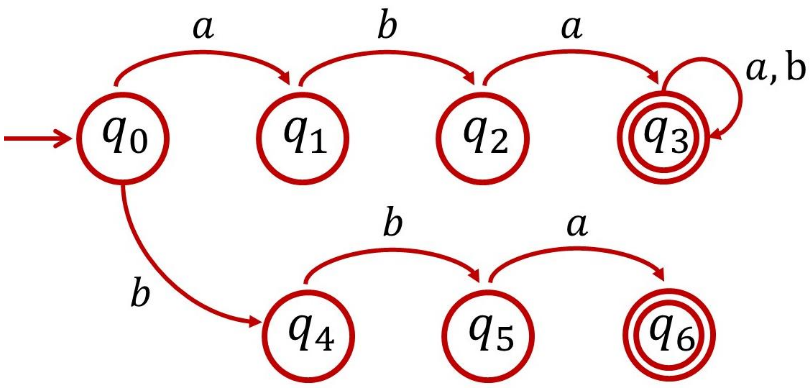
- لو كان نص المسألة or بدلاً من and فإن تحقق أحد الشرطين صحيح وبما أن الشرط الأول يشمل جميع الحالات يصبح شكل الأوتومات :



- q_0 : عند b ستبقى عند نفس الحالة تنتظر a
- q_1 : عند a ستبقى ضمن نفس الحالة تنتظر b بعد a
- q_3 : عند b تصبح السلسلة abbb وليست abba لذلك سنعود إلى الحالة الابتدائية q_0 للبحث عن السلسلة المطلوبة.

- مثال 3 : صمم أوتومات يقبل جميع السلاسل التي تبدأ ب aba أو السلاسل التي تنتهي ب bba

بما أن الأوتومات إما يقبل السلاسل التي تبدأ ب aba أو السلاسل التي تنتهي ب bba
فسنعبر عن كل سلسلة بمسار ونجعل الحالة الابتدائية شاملة للمسارين:
- الأول سيمثل السلسلة التي تبدأ ب aba (يحقق الشرط الأول)
- الثاني سيمثل السلسلة التي تنتهي ب bba(يحقق الشرط الثاني)



- بما أن الاوتومات هو DFA فيجب رسم التنقلات على باقي رموز الأبجدية لكل حالة :

➢ الحالة q_0 : تم رسم تناقلاتها سابقا

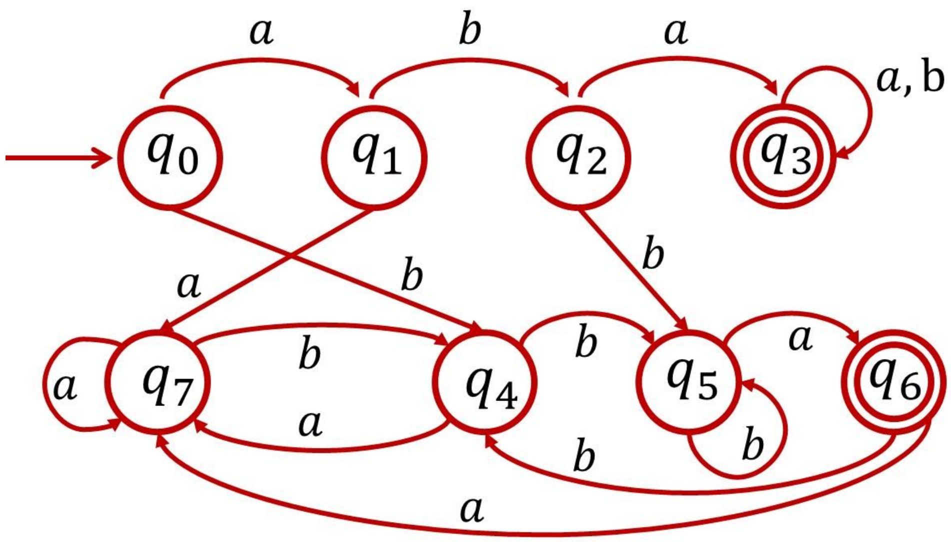
-عند الدخل a تنتقل للحالة q_1

-عند الدخل b تنتقل للحالة q_4

➢ الحالة q_1 :

-عند الدخل a

لا يمكن رسم هذا الانتقال بالمسار الأول لأنه ستكون جميع السلاسل غير مقبولة (لأن شرط المسألة الأول لن يتحقق)،فمثلاً لو انتقلنا عند الدخل a للحالة q_2،السلسلة aaa مرفوضة وبالتجريب على باقي الحالات في المسار الأول نلاحظ أن الشرط لن يتحقق.



وبالمسار الثاني فرضاً لو انتقلنا عند الدخل a للحالة q_4 تكون السلسلة aaba مرفوضة
وبالتجريب على باقي الحالات في المسار الثاني نلاحظ أن الشرط الثاني لن يتحقق.

لذا سنلجأ لزيادة عدد الحالات وإضافة حالة جديدة ولتكن q_7 بحيث يكون الانتقال من q_1 عند الدخل a للحالة q_7

- عند الدخل b ننتقل للحالة q_2

➢ الحالة q_3 :

-عند الدخلين a,b نبقى في الحالة نفسها لأنه طالما بدأت السلسلة ب aba لايهم بماذا ستنتهي.

وبنفس الطريقة ندرس التنقلات عند بقية الحالات بحيث اي انتقال من حالة لحالة أخرى عند رمز ما تكون السلسلة عنده غير مقبولة نبحث عن حالة أخرى للانتقال لها محققة الشروط السابقة.

- نريد التعبير عن تابع الانتقال بجدول طريقة كتابته كالتالي:

	$Q\backslash\Sigma$	a	b
الحالات			
الحالة الابتدائية	$\rightarrow q_0$	q_1	q_4
	q_1	q_7	q_2
	q_2	q_3	q_5
حالة نهائية	$* q_3$	q_3	q_3
	q_4	q_7	q_5
	q_5	q_6	q_5
حالة نهائية	$* q_6$	q_7	q_4
	q_7	q_7	q_4

- q_0 إذا جاءها انتقال a تذهب لـ q_1 وإذا جاءها انتقال b تذهب لـ q_4 وهكذا في باقي الجدول.
- يهمنا الجدول السابق لنحول من NFA إلى DFA

تمارين للحل:

1. سلاسل تحوي أربع أصفار فقط من أبجدية تتألف من 0 و 1 وهذه الأصفار ليس بالضرورة متتالية
2. سلاسل تنتهي بـ 1101 من أبجدية 0,1
3. سلاسل لا تحوي 110 من أبجدية 0,1

انتهت المحاضرة.