المجلة الدولية للهندسة وتقنية المعلومات
journal homepage:www.ijeit.misuratau.edu.ly

مقدمة للغات الرسمية ونظريات الأتمتة

عبد المجيد مختار عفط
جامعة مصراته ، علوم حاسوب ، مصراته ، ليبيا
majid.afat@it.misuratau.edu.ly

الملخص— **هذه الورقة تقدم شرح مفصل حول اللغات الرسمية والطرق المختلفة لتمثيل هذه اللغات، كذلك تعريف الأتمتة وتصنيفاتها والربط بين اللغات الرسمية ونظريات الأتمتة. بالإضافة إلي ذلك توضح الخوارزميات المستخدمة للتحويل بين تصنيفات الأتمتة المختلفة. بالإضافة إلي توضيح الفرق بين اللغات التامة والغير تامة والضوابط التي تحدد كلا منهما. كما تقدم تعريف المشكلة في هذا المجال واستخدام العمليات علي اللغات التي تساعد في تكوين لغات كبيرة ومعقدة يمكن استخدامها في تطوير البرمجيات الضخمة. هذا بالإضافة إلي إعطاء بعض المفاهيم الأساسية الواجب معرفتها للخوض في مجال اللغات الرسمية ونظريات الأتمتة.**

الكلمات المفتاحية: **أتمتة، DFA، NFA، PDA، التعبير القياسي، العمليات علي اللغات.**

1. المقدمة

الأتمتة هي نتاج البشرية من خلال التطوير الفكري، وذلك لجعل الأمور منجزه بشكل تلقائي أو آلي أحيانا لإظهار بعض العظمة وفي أغلب الحالات للوصول إلي حد كبير من الإتقان . ففي هذه الأيام يتم استخدام الأتمتة لإنجاز بعض المهام التي لا يمكن للبشر القيام بها أو لتسريع الإنتاجية .

أما في مجال علوم الحاسوب تعتبر الأتمتة هي البرمجة المجردة(Abstract computing) وهي البرمجة من غير وسيط أو مساعد الذي يكون في الغالب المترجم(Compiler). لذا من المهم جدا استثمار المزيد من الوقت لدراسة الأتمتة لتطوير برمجيات ضخمة كالمترجم وغيرها من البرمجيات التي تكون قريبة من لغة الالة.

2. اللغات الرسمية

قبل تعريف اللغة يجدر التعرف على مجموعة من المفاهيم الأساسية للغات الرسمية ومن بنها الآتي:

المجموعة "هي تجميع لأشياء مختلفة من مجال أو مجالات مختلفة" [1].

المجموعة الهجائية ∑ وهي عبارة عن مجموعة منتهية من الرموز كالأمثلة التالية:

- أحرف اللغة العربية {أ،ب،ت،...،ي}.
- أحرف اللغة الإنجليزية {a,b,c,…,z}.
- الأرقام الثنائية{0,1}.

قوة المجموعة: هي عبار عن مجموعة من الكلمات بطول أس المجموعة فإذا كان ∑={0,1} فإن:

- $\sum^1$={0,1}
- $\sum^2$={00,01,11,10}
- $\sum^0$={ε}

حيث ε تعرف بالسلسلة الفارغة وهي عبار عن كلمة طولها 0 وطول الكلمة عبارة عن عدد أحرفها.

$\sum^* = \sum^0 \cup \sum^1 \cup \sum^2 \ldots$

اللغة:

عبارة عن مجموعة من الكلمات المختارة من مجموعة هجائية [2].

هذا التعريف ينطبق على كل اللغات بما في ذلك اللغات الطبيعية.

فإذا كانت لدينا مجموعة هجائية وكانت L جزئية من $\sum^*$، إذا تعتبر L لغة. كما يلاحظ في الأمثلة التالية:

- مجموعة الكلمات في اللغة العربية.
- مجموعة الأرقام الثنائية التي تحتوي على عدد زوجي من الأصفار و الواحدات.
- مجموعة من الأرقام الثنائية التي تحتوي علي عدد س من الأصفار وعدد س من الواحدات.

يختلف مفهوم اللغة هنا عن نظيره في اللغات الطبيعية من حيث المكون، فاللغات الطبيعية تكون مجموعة من المفردات التي لها معني وتربطها مجموعة من القواعد النحوية لتركيب تلك المفردات لتكوين الجملة. أما اللغة في نظريات الأتمتة فتعتبر مجموعة من الكلمات التي من الممكن أن تكون فيما بينها أحيانا أنماط تربطها.

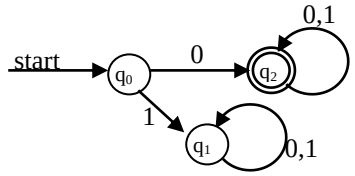
3. تعريف المشكلة في نظريات الأتمتة

دائما يتم التفكير في المشكلة على أنها كيفية تحويل المدخلات إلى مخرجات. أما في نظريات الأتمتة فتتلخص المشكلة على عضوية الكلمة في لغة معينة [2].

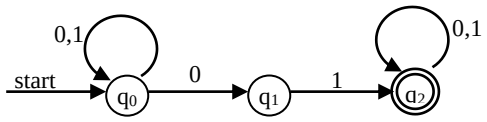
4. الأتمتة المنتهية

عبارة عن مجموعة من الحالات بحيث يتم التنقل من حالة إلى أخري استجابة لمدخل خارجي [2].

تصنف الأتمتة المنتهية على حسب المدخلات إلى صنفين أساسيين حتمي Deterministic Finite Automata DFA وتعني أن المدخل الواحد ينقل الحركة الى حالة واحده في المرة الواحدة كما موضح في (الشكل 1)، وغير حتمي NON-deterministic Finite Automata DFA ويعني من الممكن الوصول لأكثر من حالة مع المدخل الواحد في المرة الواحدة كما موضح في (الشكل 2) [2].



شكل 1. المخطط الانتقالي لمجموعة الأرقام الثنائية التي قيمتها زوجية.



شكل 2. المخطط الانتقالي للأرقام الثنائية التي تحتوي على صفر متبوعا بواحد.

استلمت الورقة بالكامل في 11 اغسطس 2020 وروجعت في 30 سبتمبر 2020 وقبلت للنشر في 8 اكتوبر 2020

ونشرت ومتاحة على الشبكة العنكبوتية في 18 اكتوبر 2020

5. مكونات الأتمتة المنتهية

تتكون الأتمتة المنتهية من خمسة أشياء أساسية وهي تعرف ب -5**tuple** $(Q, \sum, q0, \delta, F)$ حيث:

- Q عبارة عن مجموعة منتهية من الحالات.
- $\sum$ هي مجموعة الهجاء الخاصة باللغة.
- q0 هي حالة البداية.
- F مجموعة الحالات النهائية وتكون جزئية من مجموعة الحالات.
- δ هي دالة التنقل δ(q,a)=p، والتي تعني أن الحالة q مع المدخل a تؤدي الي الحالة p.

تكون 5-tuple أعلاه للأوتومات الحتمي والغير حتمي باختلاف في دالة التنقل بحيث تكون في الأتومات الغير حتمي كالتالي:

δ(q,a)={p1,p2,...}

بمعني أن الحالة q مع المدخل a تنقل الأتومات لمجموعة من الحالات هذه المجموعة تكون جزئية أو مساوية لمجموعة الحالات Q.

6. محددات التصميم

ملاحظة: عند تصميم أي أتومات للغة معينة يجب مراعات التالي:

1. الكلمات المقبولة تنهي عملية التنقل إلي حالة نهائية.
2. الكلمات الغير مقبولة تنهي عملية التنقل إلي حالة غير منتهية.
3. لا يجب ان تتوقف عملية التنقل إلا بعد الانتهاء من الكلمة، وهو ما يعرف ب Die والذي يحدث عندما يتم استقبال مدخل ولا يوجد حالة للانتقال إليها.

إذا لم تتوفر النقاط أعلاه في التصميم فيعتبر التصميم خاطئ ويجب إعادة النظر في التصميم.

7. استخدامات الأتمتة المنتهية

تعتبر الأتمتة المنتهية هي تمثيل للحوسبة المجردة Abstract computing [2]. والحوسبة المجردة تعني إجراء عمليات البرمجة او الحوسبة من دون وسيط مساعد. فمثلا المبرمج يقوم بالبرمجة بمساعدة المترجم، أما مصممي المترجمات ففي الغالب يشتغلون على الآلة مباشرة. تستخدم الاتمتة المنتهية كنموذج في المجالات التالية:

- تصميم برمجيات التحقق من الدوائر الكهربائية.
- في بناء محلل المفردات في المترجمات.
- البحث عن كلمة في ملف أو الويب.
- تصميم برمجيات للتحقق من الأنظمة التي لها عدد محدود من الحالات، مثل بروتوكولات الاتصالات.

من الممكن استخدام الاتمتة المنتهية في معالجة اللغات الطبيعية.

8. خوارزمية التحويل من NFA إلي DFA

يعتبر الأتومات الغير حتمي **NFA** أسهل في التصميم ولكن من الصعب بل من المستحيل أحيانا تتبعه لمعرفة عضوية الكلمة في لغة معينة، بينما الاتومات الحتمي **DFA** يعتبر سهل جدا في التتبع ومن ثم من الممكن برمجته باستخدام الحواسيب التقليدية. لذلك يتم تصميم أوتومات غير حتمي ومن ثم تحويله إلي أوتومات حتمي باستخدام خوارزمية التحويل.

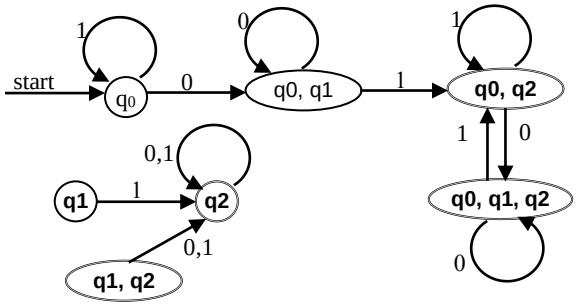
تعرف خوارزمية التحويل ب **Subset construction algorithm** وهي تقوم بإيجاد جميع المجاميع الجرئية من مجموعة الحالات ومن ثم إيجاد دالة الانتقال لكل مجموعة. وفيما يلي تفاصيل الخوارزمية:

- $Q_D = \{S : S \leq Q_N\}$
- $F_D = \{S : S \leq Q_N \text{ and } S \cap F_N \neq \emptyset\}$
- لكل S من Q_D ولكل مدخل من $\sum$ دالة الانتقال تكون كالتالي:
- $\delta_D(S,a) = U_{p \text{ in } s}\, \delta_N(p,a)$
- $q_{oD} = q_{oN}$ تبقي حالة البداية في مكانها أي لا تتغير.
- بالنسبة لمجموعة الهجاء تكون نفسها.

مثال تحويل الأتومات المبين بالشكل 2:

جدول 1. جدول انتقال يبين عملية التحويل من NFA إلي DFA

1	0	
q0	q0, q1	q0<-
q2	Ø	q1
q2	q2	q2*
q0, q2	q0, q1	q0, q1
q0, q2	q0, q1, q2	q0, q2*
q2	q2	q1, q2*
q0, q2	q0, q1, q2	q0, q1, q2*



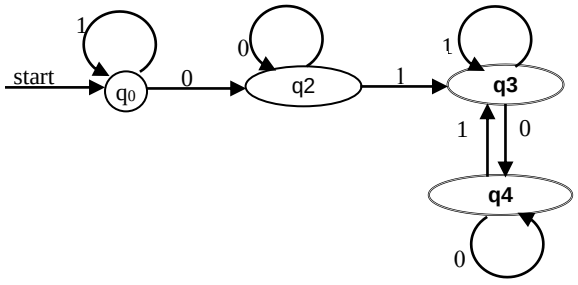
شكل 3. مخطط الانتقال بعد عملية التحويل للأتومات الموضح بالشكل 2.

يلاحظ في (الشكل 3) أن الحالات **{q1} ,{q2} ,{q1, q2}** حالات لا يمكن الوصول إليها وهي تعرف ب **Inaccessible state** لذا يتم تجاهلها، بالإضافة إلي ذلك عدد الحالات الناتج 2^3-1= 7حيث 3 هو عدد الحالات الأصلية. فإذا كان عدد الحالات 10 فعدد الحالات الناتج يكون 2^{10}-1=1023 حالة وهذا مكلف. لهذا يتم التعامل مع الحالات الممكن الوصول إليها فقط والمعروفة ب **Accessible state** كالتالي:

الأساس: q_0 تعتبر **Accessible state**.

الحث: إذا كانت **S Accessible state** إذا $U_{a\in \sum}\, \delta(S,a)$. بمعني جميع الحالات التي يتم الوصول إليها من **Accessible state** تعتبر **Accessible state**.

ملاحظة: يمكن تغيير أسماء الحالات باسماء جديدة لتكون سهلة في التعامل لتكون نتيجة التحويل كما هي بالشكل 4.

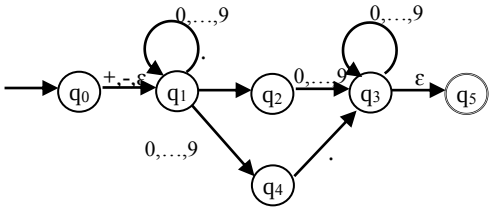


شكل 4. مخطط الانتقال بعد عملية التحويل للأتومات الموضح بالشكل 2.

يلاحظ أن الشكل الناتج من عملية التحويل يعطي نفس النتائج. أي بمعني ان الكلمات التي تكون مقبولة بالشكل 2 تكون مقبولة بالشكل 3، والغير مقبولة بالشكل 2 تكون غير مقبولة بالشكل 3.

9. أتومات غير حتمي مع القيمة ε

يعرف ب **Non-deterministic finite Automata with ε value(ε-NFA)** وهو يسمح لعملية التنقل بالاستمرار مع قيمة السلسلة الفارغة ε. ففي بعض الأحيان تستدعي الحاجة لأن يستمر الانتقال مع القيمة ε **[2]**. مثال علي ذلك الأرقام العشرية قد تكون البداية بإشارة الموجب أو السالب وقد لا تأتي الإشارة لذلك يتم وضع القيمة ε لتمثل عدم وجود الإشارة كما موضح بالشكل 5.



شكل 5. مخطط الانتقال لمجموعة الأرقام العشرية.

للرقم العشري أنماط مختلفة وهي الحالة الطبيعية وهي عبارة عن مجموعة من الأرقام والفاصلة العشرية ومن ثم مجموعة من الأرقام، ومن بن الأنماط أيضا فاصلة عشرية ومن ثم سلسلة من الأرقام، وقد تكون مجموعة من الأرقام متبوعة بفاصلة عشرية، بالإضافة إلى ذلك قد يبدأ الرقم بالإشارة أو لا. كل الأنماط السابقة تتماشي مع (الشكل 5).

10. خوارزمية التحويل من ε-NFA إلي DFA

لا يمكن تتبع ال ε-NFA لذا يجب تحويله إلى DFA. ولكن قبل الخوارزمية يجب التعرف على Eclose. وهو عبارة عن مجموعة من الحالات التي يمكن الوصول إليها بواسطة سلسلة من قيمة ε.

مثال : Eclose(q_0) في الشكل 5 {q_0,q1}.

يجب ملاحظة أن كل حالة تنتمي ل Eclose الخاص بها.

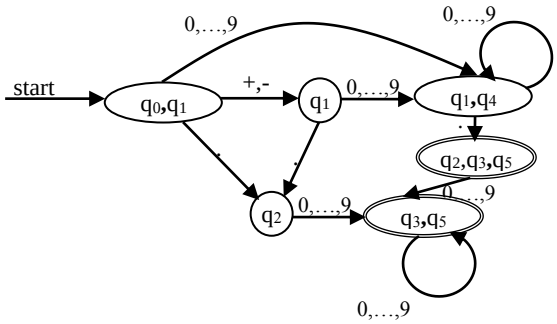
يمكننا ال Eclose من التخلص من قيمة ε. وفيما يلي استعراض لخطوات الخوارزمية:

- Q_D={S : S ⊆Q_E and S = ECLOSE(S)}
- q_D = ECLOSE(q0)
- F_D={S : S ∈ Q_D and S ∩ F_E ≠ Ø}
- δ_D(S,a) = $U_{\{ECLOSE(P) : P\in \delta(t,a) \text{ for some } t\in S\}}$.

مجموعة الحالات Q_D تحتوي علي الحالات الت يمكن الوصول إليها فقط. الجدول أدناه يوضح تطبيق الخوارزمية علي الأتومات في الشكل 5:

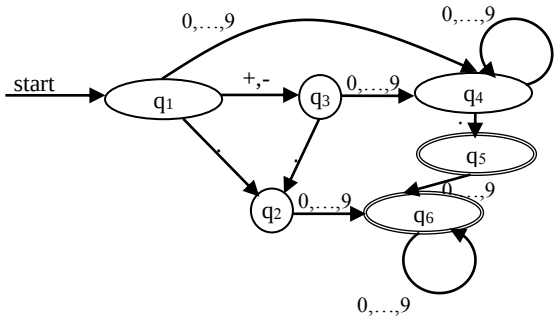
جدول 2. جدول انتقال يبين عملية التحويل من ε-NFA إلي DFA

0,…,9	.	+,-	
q_1,q_4	q_2	q_1	q_0,q_1<-
q_1,q_4	q_2	Ø	q_1
q_3,q_5	Ø	Ø	q_2
q_1,q_4	q_2,q_3,q_5	Ø	q_1,q_4
q_3,q_5	Ø	Ø	q_3,q_5*
q_2,q_3,q_5	Ø	Ø	q_2,q_3,q_5*



شكل 6. مخطط الانتقال بعد عملية التحويل للأتومات الموضح بالشكل 5.

بعد الانتهاء من عملية التحويل يمكن تغيير أسماء الحالات بأسماء مختصره كما موضح في المخطط التالي.



شكل 7. النتيجه النهائية لعملية التحويل.

بعد إجراء عملية التحويل يلاحظ أن جميع الأنماط التي تم الإشارة إليها تتماشي مع المخطط بالشكل 7، بالإضافة إلي ذلك تم التخلص من قيمة ε.

11. العمليات علي اللغات الرسمية

توجد ثلاث عمليات يمكن إجرائها علي اللغات وهذه العمليات تمكن من إنتاج لغة معقدة. بصورة أخري في بعض الأحيان تكون اللغة المطلوب تصميم مخطط انتقال لها معقدة ومن الصعب إجراء التصميم مباشرة لذلك يتم تقسيم المهمة إلي مجموعة من المهام ومن ثم تصميم المخطط لكل مهمة جزئية و أخيرا يتم إجراء العملية المطلوبة لربط تلك المهام الجزئية. وفيما يلي العمليات الثلاث على اللغات:

1. الإتحاد(**Union**):

LUM={w: w∈L or w∈M }

2. الدمج(**Concatenation**):

L.M={w: w=xy, x∈L and y∈M }

3. القوة(**Powers**):

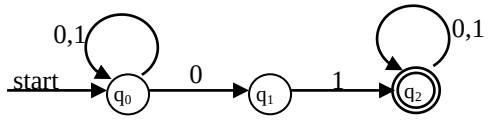
L^0={ε} L^1=L L^{k+1}=L.L^K

4. **kleene closure**:

$L^*=U^{\infty}_{i=0}\ L^i$

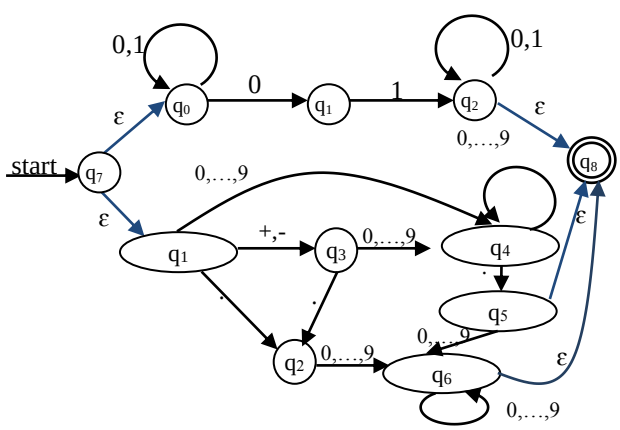
في المثال التالي يتم استعراض عملية الاتحاد لأنها تمكن من عملية ربط مجموعة من المخططات الانتقالية مع بعضها البعض.

الشكل(**8**) يبين مخطط الانتقال للغة **{$L=\{w: w=x01y , x,y\in\{0,1\}^*\}$**، وهي عبارة عن لغة الأرقام الثنائية التي تحتوي علي 0 متبوعا ب 1.



شكل 8. مخطط الإنتقال للغة $L=\{w: w=x01y , x,y\in\{0,1\}^*\}$.

يمكن ربط اللغة المبينة بالشكل 7و8 وذلك بتوحيد حالات البداية في حالة واحدة، وكذلك الحالات النهائية بحالة واحدة ويتم الربط باستخدام قيمة ε كما موضح أدناه:



شكل 9. عملية الاتحاد بين لغتين.

اللغة الناتجة هي عبار عن جميع الأرقام الثنائية التي تحتوي علي 0 متبع ب 1 كذلك جميع الأرقام العشرية. بصورة أخري تم إنتاج مخطط إنتقال **ε-NFA** جديد يقبل الأعداد الكسرية والارقام الثنائية التي تحتوي على 0 متبع ب1.

يمكن تحويل المخطط بالشكل 9 الى مخطط حتمي **DFA** باستخدام **Subset construction algorithm**.

بالنسبة لعملية الدمج فيتم ربط المخطط الأول بالثاني باستخدام القيمة ε، وتغيير الحالات النهائية بالمخطط الأول إلي حالات غير نهائية.

ملاحظة: **L.M ≠ M.L**.

12. التعابير القياسية Regular Expressions

التعبير القياسي هو عبارة عن صيغة يندرج تحتها مجموعة من التعابير، ويتم استخدامها للتعرف على مفردات اللغة. يتركب التعبير القياسي من حروف الهجاء الخاصة بلغة معينة والتي يتم ربطها بمجموعة من العمليات التالية:

1. النجمة: وتعني تكرار التعبير صفر أو اكثر من المرات مثل ***1**.
2. النقطة: وتعني دمج التعابير وفي بعض الأحيان لا تكتب مثل 01 أو **0.1**.
3. الزائد: وتعني "أو" أي بمعني إحدى التعابير **0+1**.

التعابير القياسية صممت لتمثيل اللغات القياسية باستخدام الأدوات أو العمليات الرياضية الأولية. هذا التمثيل يتضمن مجموعة من السلاسل النصية من المجموعة الهجائية الخاصة باللغة الت تكون مترابطة بالعمليات (+،*،) **[3]**.

التعابير القياسية تعتبر نموذج جيد لتمثيل مجموعات معينة من السلاسل بأسلوب جبري **[4]**.

بالإضافة إلي ذلك يمكن أن تكون مجموعة من التعابير الجزئية داخل الأقواس، العمليات أعلاه مرتبة حسب أولوية التنفيذ. وفيما يلي تعريف تفصيلي على التعابير:

الأساس **Basis**:

تعتبر القيمة ε تعبير قياسي بحيث اللغة التي يمثلها التعبير هي {ε } أو بصورة أخري L(ε)={ε} .

تعتبر القيمة Ø تعبير قياسي بحيث L(Ø)= Ø.

إذا كان **a**∈∑ إذا يعتبر المدخل a تعبيرا بحيث L(a)={a}.

الحث Induction:

إذا كان E مجموعة من التعابير، فيعتبر E تعبير بحيث L((E))=L(E).

إذا كان E و F تعبيران فإن E+F تكون تعبيرا بحيث

L(E+F)=L(E) U L(F).

إذا كان E و F تعبيران فإن E.F تكون تعبيرا بحيث(E.F)=L(E).L(F).

إذا كان E تعبيرا فيعتبر *E تعبيرا بحيث L(E*)= (L(E))* .

بصورة عامة كل جزئية من التعبير يمكن اعتبارها تعبيرا.

مثال:

L= {w: w∈ {0,1}* and 0,1 alternate in w }

وهي عبارة عن لغة من الأرقام الثنائية التي يكون فيها الصفر والواحد بصورة متعاقبة كالتالي:

L= {01,10,010,101,0101,101010,…}

يمكن تصميم تعبر للغة بأكثر من طريقة كالتالي:

(01)*1+(01)*+0(10)*+(10)*

كل تعبير من التعابير الأربعة في التعبير السابق يشكل الأنماط المختلفة التي تمثل المفردة داخل اللغة.

أو بصورة مختلفة (ε+0)*(01)(ε+1).

أو بصورة أخري (ε+1)*(10)(ε+0).

13. تحويل التعبير القياسي إلي DFA

لكل تعبير قياسي E أتومات حتمي (DFA) A بحيث L(E)=L(A).

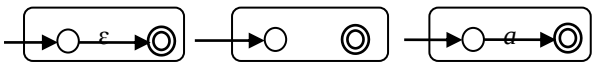
لكل تعبير قياسي E أتومات غير حتمي (ε-NFA) N بحيث L(E)=L(N).

لذلك يتم تحويل التعبير إلي ε-NFA ومن ثم يتم تحويلة إلي DFA.

التحويل من تعبير إلي ε-NFA:

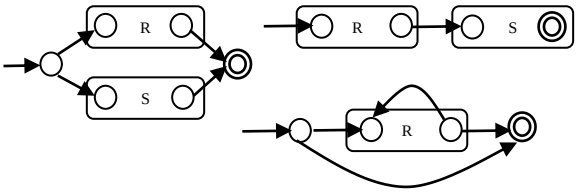
يتم العمل علي كل جزئية من التعبير وتحويلها إلي مخطط ومن ثم يتم ربط المخططات بالقيمة ε.

فيما يلي المخططات للمدخل a و ε و Ø.



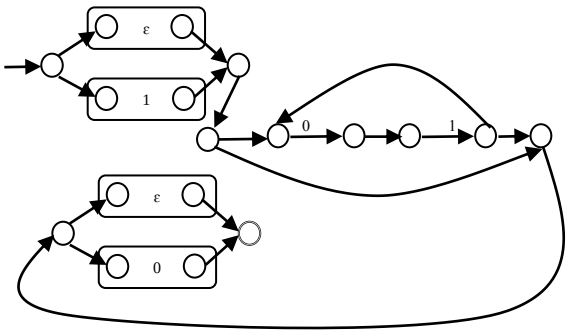
شكل 10. الأوتومات لمدخل و ε و Ø.

كذلك يتم تحويل العمليات كالتالي:



شكل 11. الأوتومات ل RS و R+S و R*.

مثال: تحويل التعبير (ε+0)*(01)(ε+1) إلى DFA.



شكل 12. تحويل تعبير قياسي إلي **ε-NFA**.

بعد ذلك يتم استخدام خوارزمية التحويل من **ε-NFA** إلي **DFA**.

14. الأوتومات الدافع للأسفل Push Down Automata(PDA)

PDA تساعد علي تمثيل بعض اللغات الغير قياسية التي لا يمكن تمثيلها بالأتومات المحدود [5].

أتومات الدافع للأسفل هو عبارة عن ε-NFA بالإضافة إلي مكدس (stack). المكدس هو عبارة عن نوع من هياكل البيانات الذي تتم الإضافة push والحذف pop من اتجاه واحد للعنصر المتواجد في قمة المكدس top.

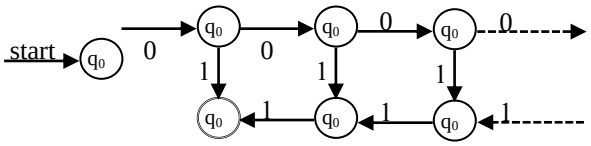
يوجد نوعان من اللغات اللغة التامة وهي اللغة التي يمكن تصميم أوتومات منتهي سواء أكان حتمي أو غير حتمي(DFA/ NFA/ ε-NFA) لتمييز الكلمة بها وتعرف ب Regular language. واللغات الغير تامة والتي من المستحيل تصميم أوتومات منتهي لها وذلك لأن عدد الحالات غير منتهي، كما موضح بالمثال التالي:

مثال:

لغة من الأرقام الثنائية التي تحتوي علي سلسلة بطول n من العدد صفر متبوعة بسلسلة طولها n من العدد واحد. والتي تصاغ على نموذج المجموعة التالية:

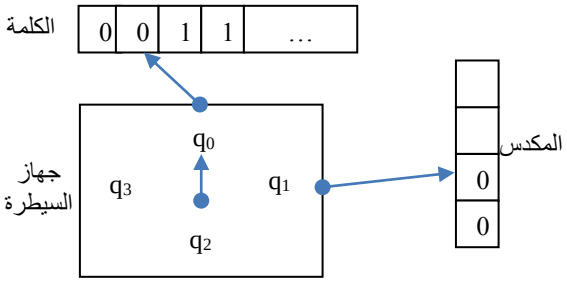
$L=\{w \in \{0,1\}^* : w=0^n1^n, n>0\}$

لا يمكن تصميم أوتومات منتهي لهذه اللغة لأن عدد الحالات سيكون غير منتهي كما موضح بالشكل التالي:



شكل 13. أتومات غير منتهي للغة $L=\{w \in \{0,1\}^* : w=0^n1^n, n>=0\}$.

يستخدم الأوتومات الدافع للأسف في تمييز الكلمة في اللغات الغير تامة. وذالك بوضع ضوابط محددة لكل لغة. الشكل التالي يبين جهاز الأتومات مع المكدس والكلمة المراد التحقق منها.



شكل 14. جهاز الأوتومات الدافع للأسفل للغة $L=\{w \in \{0,1\}^* : w=0^n1^n, n>=0\}$.

ضوابط أو آلية الجهاز:

1. إذا كان الرمز 0 وكانت الحالة **q0** وكان المكدس فارغا او علي قمته 0 يتم إضافة الرمز المقروء 0 إلي المكدس ومؤشر جهاز السيطرة يبقي في على حالته.
2. إذا كان الرمز المقروء 0 وكان علي قمة المكدس الرمز 1 أو المدخل 0 والحالة لا تساوي **q0** فإن مؤشر جهاز السيطرة ينتقل إلى حالة غير نهائية **q2** ويبقي هناك حتي نهاية الكلمة.
3. إذا تم قراءة الرمز 1 وكان علي قمة المكدس 0 فيتم سحب الرمز 0 من المكدس، ومؤشر جهاز السيطرة إذا كان لا يساوي **q2** ينتقل إلى محطة نهائية ض **q1**.
4. إذا انتهت الكلمة وكان المكدس غير فارغ فينتقل مؤشر جاز السيطرة إلي حالة غير نهائية، وإن كان المكدس فارغ فينتقل المؤشر إلي حالة نهائية ويعني ان الكلمة مقبولة.

يمكن تحويل الضوابط أعلاه إلي كود برمجي كالتالي:

```
for(int i=0;word[i]!='\0';i++)
  {
   if((word[i]=='0')&&(q==0)&&((top==-1)||(stack[top]=='0')))
     push(stack,top,'0');
     else
if(((word[i]=='0')&&(stack[top]=='1'))||((word[i]=='0')&&(q!=0))) q=2;
     else
     if((word[i]=='1')&&(stack[top]=='0'))
     {
        if (q!=2) q=1;
        pop(top);
     }
     else if (top==-1) q=2;
     }
  if ((top!=-1)||(q==2)) q=0;
  else  q=1;
  if (q==1) cout<<word<<"is accepted";
  else cout<<word<<"is rejected";
```

15. مكونات الأوتومات الدافع للأسفل (PDA)

يتكون الأتومات الدافع للأسفل من ستة مكونات:

حيث: $P=(Q,\sum,\Gamma , \delta,q0,F)$

- Q مجموعة منتهية من الحالات.
- $\sum$ مجموعة الهجاء الخاصة باللغة.
- Γ مجموعة الرموز الموجودة في المكدس.
- δ دالة الانتقال بحيث $(Q,\sum^*,\Gamma^*)=(Q,\sum *, \Gamma *)$.
- q0 حالة البداية.
- F مجموعة الحالات النهائية.

تعتبر المكونات أعلاه هي مكونات جهاز الأوتومات الدافع للأسفل، وكمثال علي ذلك اللغة التالية:

$$\mathbf{L=\{\sum=\{r,t,z\}:w=wzw^R , w\in\{r,t\}^*\}}$$

Q={r,t} $F=\{q_1\}$ $\sum$**={r,t}**

δ = **$(q_0,r,e)=(q_0,e,r)$**
$(q_0,r,e)=(q_0,e,r)$
$(q_0,z,e)=(q_1,e,e)$
$(q_0,r,r)=(q_1,e,e)$
$(q_0,t,r)=(q_1,e,e)$

في كل من التنقل الأول والثاني يتم الابتداء بحالة البداية عند قراءة أحد الرمزين **r,t**. ففي البداية يكون المكدس فارغا. بعدها تتم عملية الاضافة(**push**) للمكدس للرمز المقروء.
في التنقل الثالث يتم الانتقال لحالة النهاية بعد قراءة الرمز**z**.
في التنقل الرابع والخامس تتم عملية الحذف(**pop**).

16. الاستنتاجات

من خلال الدراسة السابقة يتضح أن بعض اللغات لايمكن تصميم أوتومات منتهي محدود لها مباشرة بل يتم تصميم أوتومات منتهي غير محدود ومن ثم تحويله باستخدام خوارزميات التحويل الى اوتومات منتهي محدود ليتم التعامل معه وبرمجته. كذلك بعض اللغات الأخري لا يمكن بأي حال من الأحوال تصميم أوتومات منتهي لها وهي ماتعرف باللغات الغير قياسية حيث يتم استخدام أوتومات الدفع السفلي للتعامل معها.

تساعد العمليات على اللغات في تصميم أوتومات يدمج مجموعة من اللغات مع بعضها وهو ما يساعد فى انتاج أوتومات معقد.

تمثل الأتمتة الحوسبة المجردة(**Abstract Computing**) لذا ينصح باستثمار المزيد من الوقت لدراستها حيث لا يمكن الإستغناء عنها فى البرمجيات القريبة من لغة الآلة.

المراجع

أ. *المراجع العربية*

[1] طارق الشيخ مجيد. (2005). النظرية الاحتسابية. جامعة إربد الأهلية. دار مجدلاوي للنشر والتوزيع.

ب. المراجع الأجنبية

[2] Jhone E.Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. (2001). Introduction to automata theory, languages and computation. By Addison- Wesley 2nd Edition.

[3] S.P. Eugene Xavier. (2005). Theory of Automata, Formal Languages and Computation. By New Age International (P) Ltd.

[4] K.I.P. MISHRA,N. (2008). CHANDRASEKARAN. Theory of Computer Science Automata, languages and Computation . third Edition.

[5] John C. Martin. (2011). Introduction to languages, and the theory of computation. By McGraw-Hill 4th Edition.