نظرية الحوسبة

الجمهورية العربية السورية
وزارة التعليم العالي
جامعة تشرين
كلية العلوم

نظرية الحوسبة

الدكتور
محمد حسن
أستاذ في قسم الرياضيات- جامعة تشرين

ثناء موسى
قائمة بالأعمال - جامعة تشرين- فرع طرطوس

لطلاب السنة الثالثة - رياضيات تطبيقية

٢٠١٤ - ٢٠١٥ م
1436 -1435 هـ

محتوى الكتاب

مقدمة

عند دراسة نظرية الحوسبة يتم التركيز على الأقسام الثلاثة الرئيسية التي تكون هذا العلم وهي :

1. نظرية التشغيل الذاتي Automaton Theory.
2. نظرية التحسيب Computability Theory.
3. نظرية التعقيد الحسابي Complexity Theory.

إن هذه الأقسام الرئيسية الثلاث تلتقي بسؤال واحد هو: "ما هي القدرات الأساسية للحاسوب وما هي حدود هذه القدرات ؟ " وكل واحد من هذه الأقسام يجيب على هذا السؤال من زاوية مختلفة ، حيث تعتمد الإجابة على طريقة تفسير كل قسم لهذا السؤال.

فنظرية التشغيل الذاتي : تتعامل مع تعريفات وخصائص النماذج الرياضية. وهذه النماذج تلعب دوراً مهماً في كثير من المجالات التطبيقية لعلوم الحاسوب. أحد هذه النماذج يسمى " التشغيل الذاتي المنت هي " Finite Automaton والذي يستخدم في معالجة النصوص وفي المترجمات وكذلك في تصميم معدات الحاسوب التي تمتلك ذاكرة محدودة جداً، كالحواسيب التي توضع داخل الآلات الكهروميكانيكية مثل الأبواب الآلية. ونظرية التحسيب : تركز على تصنيف المسائل وفرزها إلى مسائل قابلة للحل أو غير قابلة للحل. ولهذا تهتم بالنماذج النظرية للحاسوب التي تساعد في إيجاد تراكيب لحواسيب حقيقية.

أما نظرية التعقيد الحسابي : فتركز على طريقة الوصول إلى مخطط يمكن من خلاله تصنيف المسائل حسب صعوبة حل هذه المسائل حاسوبيا. بوساطة هذا المخطط يمكن عرض طريقة للدلالة على أن مسألة ما صعبة الحل حاسوبياً حتى و لو لم يكن بالإمكان إثبات ذلك.

كانت كلمة الحوسبة " Computing" أساساً تستخدم في حالة التعامل مع العدد والحساب (Calculating and Counting) ، أي العلم الذي يتعلق بإجراء الحسابات الرياضية.وقد أصبحت لاحقا تشير إلى عملية الحساب واستخدام الآلات الحاسبة وكانت العمليات الإلكترونية التي تجرى ضمن عتاد الحاسوب نفسه،إضافة إلى الأسس النظرية التي تؤسس لعلوم الحاسوب.

تعرف الحوسبة " Computation" بأنها سلسلة الخطوات الوسيطة intermediate steps التي نستخدمها في إنجاز خوارزمية مصممة لحل مشكلة أو مسألة بطريقة حاسوبية. ويمكن تعريفها أيضا أنها خوارزمية Algorithm ننفذها لتحويل مدخلات Inputs المسألة إلى مخرجات output أو نتائج (أي حلول للمسألة المطروحة). وهذا يعني أن الحاسوب يقوم بعملية حوسبة Computation عندما ينفذ برنامجاً.

ومن المعلوم، أن أي خوارزمية مكونة من مجموعة من العمليات الحسابية والمنطقية المتسلسلة، ونتيجة كل عملية تستخدم مدخلاً للعملية التالية. ويقوم البرنامج المعطى الممثل للخوارزمية بترتيب العمليات وتحديد شروط الانتقال من عملية إلى أخرى وكذلك إمكانية العودة إلى عملية سابقة أو الانتقال إلى عملية لاحقة ليست تالية (القفز إلى أعلى أو إلى أسفل).

مما سبق، يمكن تعريف الحوسبة بأنها إيجاد حلول لمسألة ما ابتداءً من معطيات مقدَّمة لها باستخدام خوارزمية. ويمكن أن يشمل هذا أيضاً إيجاد الخوارزميات المناسبة لحل نمط معين من المسائل. أي أن نظرية الحوسبة تبحث في تحليل المسائل Problems ومدخلاتها Inputs إضافة إلى الخوارزميات Algorithms المقدَّمة لحلها . هذه التعريفات كلها تشكل أساساً لنظرية التحسيب Computability Theory ونظرية التعقيد Computational Complexity Theory . تع دّ نظرية التحسيب أحد فروع المعلوماتية النظرية Theoretical Science Computer التي تدرس المسائل القابلة للحل حاسوبياً Computationally Solvable باستخدام نماذج مختلفة للحوسبة.

تختلف نظرية الحوسبة عن التخصصات المشابهة كنظرية التعقيد الحسابي، حيث إن الأخيرة تتعامل مع كيفية حل المسألة حاسوبياً بفعالية، في حين أن نظرية الحوسبة تبحث فيما إذا كانت هذه المسائل قابلة للحل Solvable أساسا باستخدام الحاسوب.

وتعدّ نظرية التعقيد الحسابي Computational Complexity ، إحدى أجزاء نظرية الحوسبة وتتعامل مع الموارد المطلوبة في عملية الحوسبة. وأكثر هذه الموارد شيوعاً هي الزمن (بمعنى كم عدد الخطوات أو ما يقابلها من الوقت يلزم لحل المسألة) والمكان (بمعنى ما حجم الذاكرة اللازمة لحل المسألة)، ويمكن أن يدخل بالحسبان موارد أخرى ، مثل: كم عدد المعالجات المتوازية اللازمة لإنجاز الحساب باستخدام برمجة متوازية.

تستخدم في نظرية التحسيب غالباً الآلات المجردة النموذجية التي تتألف من دخل وخرج ومجموعة عمليات مصرح بها تستعمل لتحويل الدخل إلى خرج . وتعدّ آلة تورينغ Turing Machine الأكثر شيوعا . يمكن أن نتصور آلة تورينغ أنها حاسوب منزلي

بسعة ذاكرة محدودة حيث لا يمكن الوصول إلا إلى قطاعات صغيرة متفرقة من هذه الذاكرة. وتمثل آلة تورينغ نموذجاً معقداً لعملية الحوسبة.

كما يمكن تعريف آلات مجردة أكثر تعقيداً بمجموعة تعليمات أوسع، مثل المسجلات ونماذج ذاكرة الحاسوب المختلفة . يدعى نموذج RAM أحد أكثر النماذج شيوعاً ومشابهة للحاسوب في وصفه الحالي، والذي يسمح بوصول عشوائي لمواقع الذاكرة المفهرسة.

يتم تعميم عمليات الحوسبة Computations لمعالجة المعلومات المختلفة حيث يمكن أن تكون هذه الحسابات بسيطة كحساب " كم يستغرق من الوقت قطع مسافة ما بالسيارة " كما يمكن أن تكون معقدة " كالحسابات التي تجرى للتنبؤ بالأحوال الجوية ومعرفة حالة الطقس ".

الهدف من هذا الكتاب الوصول لفهم عميق لخصائص وميزات العمليات الحسابية. وهذا الفهم العميق يمكن أن يستخدم في التنبؤ بمدى صعوبة العمليات الحسابية المطلوب إجراؤها وذلك من أجل اختيار الطرق والوسائل المناسبة لتسهيل وتطوير عملية تصميم هذه الطرق.

تغطي موضوعات هذا الكتاب الجوانب النظرية والعملية لمقرر نظرية الحوسبة لطلاب السنة الثالثة رياضيات تطبيقية وحرصنا في الوقت ذاته ليكون مرجعا وعونا نافعا لجميع الطلاب المهتمين في هذا العلم والله من وراء القصد.

المؤلفان

1- الفصل الأول

أساسيات نظرية الحوسبة

محتوى الفصل الأول :

1.1 . أساسيات نظرية الحوسبة.

2.1. الأبجديات والمتسلسلات (Strings and Alphabets).

3.1 . تنظيم المتسلسلات (Ordering of strings) .

4.1. تمثيل المعلومات (Representation of Information).

5.1. اللغات (Languages) .

6.1. القواعد (Grammars) .

7.1. البرنامج (Programs) .

الهدف من الفصل

يهدف هذا الفصل إلى اكتساب المعا رف التالية :

التعرف على : الأبجديات ، المتسلسلات ، تنظيم المتسلسلات، تمثيل المعلومات . اللغات و طرق تشكلها ، القواعد وطرق تعريفها ، البرامج وطرق تصميمها وكتابتها .

1.1. مقدمة.

تظهر دراسة الحوسبة أن هناك أنواع من المسائل لا يمكن حلها (غير قابلة للحل). أما المسائل القابلة للحل فإن بعضها يتطلب عملياً موارد لا يمكن توفيرها. ودراسة هذا النوع من المسائل لا يتحقق إلا من خلال دراسة الحوسبة التي تساعد في تحديد هذا النوع من المسائل كما تساعد في إيجاد الأدوات المناسبة لتحديد المسائل التي يمكن حلها وبشكل مناسب بالإضافة إلى إيجاد أدوات تصميم هذه الحلول. كما يمكن من خلال الحوسبة تطوير مصطلحات دقيقة معرفة جيداً للوصول إلى إدراك أو معرفة مسبقة عن هذه المسائل وطرق حلها.

إن القدرة على تمثيل المعلومات يع دّ أمراً حاسماً في عملية تبادل ومعالجة هذه المعلومات. لذلك عندما احتاجت المجتمعات البشرية لتبادل المعلومات حتى تتمكن من التواصل فيما بينها قامت بإيجاد لغات محكية لتحقيق التواصل المنشود. وعندما أصبحت بحاجة إلى مستوى أعلى وأعقد من التواصل قامت بتطور الكتابة Writing .

فاللغات المختلفة في شكلها الممكن تعتمد على مجموعة من الأصوات الرئيسية المحدودة كأساس للغة. أما الكلمات فيمكن تعريفها بأنها عبارة عن متواليات محدودة من هذه الأصوات. أما الجمل والعبارات فتنشأ عن متواليات محدودة من الكلمات. وفي النهاية الحديث ينشأ من متواليات محدودة من العبارات. وكل لغة مكتوبة تستخدم مجموعات من الرموز الأساسية المحدودة.

والحوسبة مثل اللغات الطبيعية تتعامل مع المعلومات في أكثر صورها عمومية.

وبناءً عليه فإن الحوسبة تتعامل مع مجموعات محدودة من الرموز، مثل الأعداد الأرقام ، الأحرف ورموز خاصة وبرامج.

2.1. الأبجديات والمتسلسلات

Strings and Alphabets

تسمى المجموعات المرتبة غير الخالية **أبجديات** إذا كانت عناصرها عبارة عن حروف أو أرقام أو رموز لها أشكال تمثيلية تعرف بها. أما المتوالية المحدودة المأخوذة من الأبجدية فتسمى **متسلسلة**. إذا كانت المتسلسلة تحوي على الرموز $s_0,s_1,s_2,\ldots,s_n$ فيمكن أن يرمز لها بالشكل $s_0s_1s_2\ldots s_n$.

أما المتسلسلة التي تحوي على صفر من الرموز فإنها تسمى متسلسلة خالية Empty String ويرمز لها بالرمز ε .

1.2.1. مثال:

إذا كانت $I_1 = \{a,b,\ldots,z\}$ وكانت $I_2 = \{1,2,\ldots,9\}$ فإن:

- Abc هي متسلسلة تنتمي إلى I_1 .
- 123 هي متسلسلة تنتمي إلى I_2 .
- Ab12 ليست متسلسلة تنتمي إلى I_1 لأنها تحوي رمزاً غير موجوداً في I_1 .
- ...314 ليست متسلسلة لأنها غير محدودة.
- المتسلسلة الخالية ε تنتمي إلى أي أبجدية.
- المجموعة الخالية Φ لا تعدّ أبجدية لأنها لا تحوي أي عنصر.

- مجموعة الأعداد الطبيعية ليست أبجدية لأنها غير محدودة.
- اتحاد الأبجديتين I_1 و I_2 ($I_1 \cup I_2$) يشكل أبجدية فقط عندما تأخذ عناصرها ترتيباً معيناً.
- الأبجدية التي تحوي عنصرين فقط تسمى أبجدية ثنائية Binary Alphabet والمتسلسلة التي تنتمي إلى هذه الأبجدية تسمى متسلسلة ثنائية. أما إذا احتوت الأبجدية عنصراً واحداً فقط فتسمى أبجدية أحادية Unary String .
- طول المتسلسلة String length يساوي عدد الرموز التي تحتوي عليها المتسلسلة. نستنتج أن طول المتسلسلة الخالية يساوي 0. وتلعب المتسلسلة الخالية الدور نفسه الذي يلعبه الرقم 0 في مجموعة الأعداد ونرمز لطول المتسلسلة X بالرمز $|X|$.

2.2.1.مثال:

- إذا كانت { 0 , 1 } أبجدية ثنائية و { 1 } أبجدية أحادية ، فإن 11 هي متسلسلة ثنائية تنتمي إلى الأبجدية الثنائية وفي الوقت نفسه هي متسلسلة أحادية تنتمي إلى الأبجدية الأحادية أما طول هذه السلسلة فيساوي 2. إذا كانت المتسلسلة بطول n فيمكن كتابتها بالشكل

$X = X_1 X_2 X_3 \ldots X_n$ حيث X_i تنتمي إلى I . وإذا كتبنا هذه الرموز بشكل عكسي فتسمى **متسلسلة عكسية** Reverse String ويرمز لها بالرمز X^R وتكتب بالشكل

$$X^R = X_n X_{n-1} X_{n-2} \ldots X_1 .$$

1.2.3.مثال:

إذا كانت $X = abcd$ فإن $X^R = dcba$.

تسمى المتسلسلة Z **متسلسلة فرعية** Substring من X إذا كانت تتكون من توالي رموز موجودة في X .

1.2.4.مثال:

تشكل cad متسلسلة فرعية من المتسلسلة abracada .

إذا كانت المتسلسلة X بطول m والمتسلسلة Y بطول n فإن XY هو تتابع أو توالي Concatenation المتسلسلتين ويتم الحصول على XY من إلحاق Y بنهاية X ، أي

$XY = x_1\ x_2 \ldots x_m\ y_1\ y_2 \ldots y_n$.

1.2.5.مثال:

إذا كانت $X = 01$ و $Y = 100$ فإن $XY = 01100$.

إن توالي المتسلسلة الخالية ε مع X يساوي X وتوالي ε مع نفسها يساوي ε.

إذا كانت $X = 011$ فإن المتسلسلات 011 , 101 , 110 هي متسلسلات تبادلية Permutations Strings للسلسلة X.

نمثل مجموعة جميع المتسلسلات التي تنتمي إلى الأبجدية I بالرمز I^* ويكون $I^* = I - \{\varepsilon\}$.

1.2.6.مثال :

- إذا كانت $X = 011$ فإن 011 , 11 , 01 , 1 , 0 , ε هي متسلسلات فرعية من X.
- المتسلسلات 01 , 0 , ε تسمى متسلسلات بادئة تامة Proper Prefixes strings والمتسلسلات 11 , 1 , ε فتسمى متسلسلات لاحقة تامة Proper Suffixes strings .

3.1 . تنظيم المتسلسلات Ordering of Strings .

تعدّ عملية البحث Searching أكثر العمليات التطبيقية العامة التي يتم إجراؤها على المعلومات Information . وتكمن أهمية هذه العمليات في تسهيل عملية البحث عن المعلومات وفي عملية تنظيمها بغض النظر عن الطريقة التي تتم فيها هذه العمليات . وفي معظم الحالات فإن الوصول إلى هذه الأهداف يعتمد على وجود علاقات يتم من خلالها تعريف عملية تنظيم مكونات أو كينونات المسألة المراد البحث فيها. وفيما يتعلق بالمتسلسلات فإن العلاقة التي يتم تكرارها هي مقارنتها أبجديا كما يحدث في تنظيم الأسماء في دليل الهاتف. ويمكن توضيح ذلك في المثال التالي:

1.3.1.مثال:

لدينا أبجدية $I = \{ 0 , 1 \}$ و 01 متسلسلة تنتمي إليها . من الواضح أن هذه المتسلسلة أصغر من المتسلسلة 01100 التي تنتمي إلى I لأن المتسلسلة 01 بادئة تامة من السلسلة 01100 . كما أن المتسلسلة 01100 أبجديا أ صغر من 0111 وذلك لأن المتسلسلتين تشتركان في أول ثلاثة رموز ولكن الرابع في الثانية أكبر

من الرمز الرابع في الأولى. كما أن مجموعة المتسلسلات التي تنتمي إلى I والمرتبة أبجدياً من الأصغر إلى الأكبر ولا يزيد طول أي منها عن 3 هي:

$$I^* = \{ \varepsilon , 0 , 00 , 000 , 001 , 01 , 010 , 011 , 1 , 10 , 100 , 101 , 11 , 110 , 111 \}$$

يكون هذا الترتيب الأبجدي سهلاً بالنسبة للمجموعات المحدودة، لأنه يمكن الوصول إلى أي من هذه المتسلسلات في هذه المجموعات في نهاية الأمر بغض النظر عن مكان وجودها . أما إذا كانت المجموعات كبيرة فيصعب ذلك . لذا نلجأ إلى ما يسمى التنظيم القانوني Canonical Ordering للمتسلسلات حيث تكون المتسلسلة مسبوقة بعدد محدود من المتسلسلات الأخرى.

حسب التسلسل القانوني تكون المتسلسلة X الأصغر قانونياً Canonically Smaller في الأبجدية I^* من المتسلسلة Y إذا تحقق أحد الشرطين التاليين:

1. إذا كان طول X أصغر من طول Y .
2. إذا كان الاثنان بنفس الطول ولكن X أصغر أبجدياً من Y .

2.3.1.مثال :

في الأبجدية $I = \{ 0 , 1 \}$ تكون المتسلسلة $X = 11$ أصغر قانونياً من المتسلسلة $Y = 000$ ، لأن طول X أقل من طول Y . كما أن المتسلسلة $X = 00$ أصغر من $Y = 11$ لأنها من الطول نفسه ولكن أبجدياً X أصغر من Y . لذلك إذا قمنا بترتيب المتسلسلات التي تنتمي إلى الأبجدية المعطاة في المثال السابق قانونيا وليس أبجديا فإن مجموعة هذه المتسلسلات تأخذ الشكل التالي:

$I^* = \{ \varepsilon , 0 , 1 , 00 , 01 , 10 , 11 , 000 , 001 , 010 , 011 , 100 , 101 , 110 , 111 \}$.

4.1. تمثيل المعلومات

. Representation of Information

من التعريفات والتوضيحات السابقة للأبجدية والمتسلسلات يمكن النظر إلى تمثيل المعلومات على أنه مخطط ترتيب الكائنات Objects داخل متسلسلات حسب قواعد وقوانين معينة .

إن الأطراف أو الجهات parties التي تقوم بتبادل جزء من المعلومات التي تنجز عمليات التمثيل والتحويل. فالتمثيل Representation يقوم به المرسل Sender أما التحويل Interpretation فيقوم به المستقبل Receiver وكذلك الأمر بالنسبة للعملية Process فليس المهم إذا كانت هذه الأطراف عناصر بشرية أم برامج Programs . لذا فمن وجهة نظر الأطراف المستخدمة فإن اللغة هي فقط عبارة عن مجموعة من المتسلسلات وهذه الأطراف هي التي تضع الحدود لتمثيل وتحويل هذه المتسلسلات.

5.1. اللغات Languages .

إذا كانت I أبجدية وكانت L مجموعة جزئية Subset من I^* فإنه يمكن القول إن L هي **لغة** Language تنتمي إلى I وكل عنصر من L يسمى **كلمة** Word أو **جملة** Sentence أو **متسلسلة** String.

تشكل المجموعات التالية : $\{0,1\}^*$, $\{\varepsilon, 10\}$, $\{0, 11, 001\}$ مجموعات جزئية من $\{0,1\}^*$ لذلك تعدّ جميعها لغات تنتمي إلى الأبجدية $\{0,1\}$.

المجموعة الخالية Φ والمجموعة $\{\varepsilon\}$ هما لغتان تنتميان إلى الأبجديات كلها.

Φ هي لغة لا تحوي أية متسلسلة.

$\{\varepsilon\}$ هي عبارة عن لغة تحتوي المتسلسلة الخالية ε .

اتحاد Union اللغتين L_1 , L_2 يمثل على النحو التالي $L_1 \cup L_2$ وهو عبارة عن اللغة التي تحتوي على جميع المتسلسلات الموجودة في L_1 أو L_2 أي :

$\{x : x \in L_1 \text{ Or } x \in L_2\}$.

تقاطع Intersection اللغتين L_1 , L_2 يمثل بالشكل $L_1 \cap L_2$ وهو عبارة عن اللغة التي تحتوي على جميع المتسلسلات الموجودة في L_1 و L_2 بللوقت نفسه أي :

$\{x : x \in L1 \text{ And } x \in L_2\}$.

متمة Complementation اللغة L التي تنتمي إلى I ونرمز لها L' هي عبارة عن جميع المتسلسلات الموجودة في I وليست موجودة في L حيث تحقق :

$\{x : x \in I^* \text{ And } x \notin L\}$.

1.5.1.مثال :

إذا كانت $L1 = \{\varepsilon, 0, 1\}$, $L_2 = \{\varepsilon, 01, 11\}$ فإن

$L_1 \cup L_2 = \{\varepsilon, 0, 1, 01, 11\}$

$L_1 \cap L_2 = \{\varepsilon\}$

$L_1' = \{ 00 , 01 , 11 , 000 , 001 , \dots \}$.

متمة Complementation بين L_1 , L_2 يرمز له بالشكل $L_1 - L_2$ وهو عبارة عن جميع المتسلسلات الموجودة في L_1 وغير موجودة في L_2 وتعطى بالشكل:

$\{ x : x \in L_1 , x \notin L_2 \}$.

2.5.1.مثال :

إذا كانت $L_1 = \{ \varepsilon , 1 , 01 , 11 \}$, $L_2 = \{ 1 , 01 , 101 \}$ فإن

$L_2 - L_1 = \{ 101 \}$, $L_1 - L_2 = \{ \varepsilon , 11 \}$.

ضرب Cross Product اللغتين L_1 , L_2 يمثل بالشكل $L_1 \times L_2$ وهو عبارة عن المتسلسلة التي تحوي جميع الثنائيات المرتبة (x , y) بحيث $x \in L_1 , y \in L_2$ أي :

$\{ (x , y) : x \in L_1 , y \in L_2 \}$.

تركيب Composition L_1 مع L_2 يمثل بالشكل $L_1 L_2$ وهو عبارة عن لغة تحقق العلاقة

$\{ x y : x \in L_1 , y \in L_2 \}$.

3.5.1.مثال :

إذا كانت $L_1 = \{ \varepsilon , , 0 , 1 \}$, $L_2 = \{ 01 , 11 \}$ فإن

$L_1 \times L_2 = \{ (\varepsilon , 01) , (\varepsilon , 11) , (0 , 01) , (0 , 11) , (1 , 01) ,$
$(1,11) \}$

$L_1 L_2 = \{ 01 , 11 , 001 , 011 , 101 , 111 \}$

$L - \Phi = L$

$\Phi - L = \Phi$

$\Phi L = \Phi$

$\{ \varepsilon \} L = L$

إذا كانت L لغة فإن L^i تمثل **التركيب** Composition المتكون من تكرار L عدد i من المرات ، حيث $L^0 = \{ \varepsilon \}$.

تسمى المجموعة $L^0 \cup L^1 \cup L^2 \cup L^3 \ldots$ نهاية L ويرمز لها L^* .

تسمى المجموعة $L^1 \cup L^2 \cup L^3 \ldots$ نهاية L الموجبة ويرمز لها L^+ .

4.5.1. مثال :

إذا كانت $L_1 = \{ \varepsilon , 0 , 1 \}$ وكانت $L_2 = \{ 01 , 11 \}$ فإن

$$L_1^2 = \{ \varepsilon , 0 , 1 , 00 , 01 , 11 \}$$

$$L_2^3 = \{ 010101 , 010111 , 011111 , 110101 , 110111 , 111101 , 111111 \}.$$

6.1. القواعد Grammars :

بشكل عام ، إن اللغة التي يمكن تعريفها بوساطة نظام شكلي Formal System تمتلك عدداً محدوداً من القواعد والأحكام المستنتجة تسمى لغة شكلية Formal Language

يعدّ توصيف اللغة من خلال قواعد محددة أمراً مريحاً. وتظهر الفائدة من ذلك عند استخدام عدد قليل من القوانين أو الأحكام Rules لوصف لغة تحوي عدد اً كبيراً من الجمل.

وبشكل خاص، سنأخذ بالحسبان فقط القواعد التي تتكون من مجموعات محددة من القواعد أو الأحكام وهذا النوع يسمى قواعد من النوع 0 (Type 0 Grammars) أو قواعد بناء التعابير Phrase Structure Grammars . أما اللغة الشكلية التي تنتجها هذه القواعد فتسمى لغات من النوع 0 .

نرمز لجميع القواعد من النوع 0 بالرمز G ، ويمكن تعريفها بأنها نظام رياضي يتكون من الرباعية (N , I , P , S) حيث :

N - أبجدية عناصرها رموز غير طرفية Nonterminal Symbols.

I - أبجدية عناصرها رموز طرفية Terminal Symbols.

P - عبارة عن علاقة من تشكيل محدود $(N \cup I^*)$ وتسمى عناصرها قوانين إنتاج Production Rules . كما أن كل من هذه القواعد (α , β) في P والتي تمثل على النحو التالي: $\alpha \rightarrow \beta$ التي تمتلك رمزاً واحداً على الأقل غير طرفي.

S – هو أحد رموز N ويسمي البداية Start .

1.6.1. مثال:

إذا كانت (N , I , P , S) قواعد من النوع 0 ، وكانت

$I = \{ a , b \} \ , \ N = \{ S \} \ , P = \{ S \rightarrow a S b , S \rightarrow \varepsilon \}$

فإن هذه القاعدة تمتلك عنصراً واحداً غير طرفي هو S ، وعنصري طرفيين هما a , b بالإضافة إلى قاعدتي إنتاج هما $S \rightarrow \varepsilon$, $S \rightarrow aSb$.

ملاحظة: لا تعدّ الرباعية (N_1 , I_1 , P_1 , S) قواعد إذا كانت N_1 مجموعة الأعداد الطبيعية أو إذا كانت I_1 فارغة ، لأنهما في هذه الحالة لا تعتبر أبجديات.

7.1. البرنامج Programs .

يعرف البرنامج بأنه متسلسلة من الأوامر أو التعليمات Instructions التي تتطابق مع خطوات الخوارزمية المقترحة لحل المشكلة المطلوب حلها. وعند إجراء عمليات الحوسبة فإنه يتم استخدام إحدى اللغات البرمجية المعروفة لكتابة البرنامج مستخدما بذلك جميع القواعد المعتمدة في اللغة البرمجية المختارة Programming Language .

2- الفصل الثاني

الحاسبة المنتهية (الأتومات المنتهي)

Finite Automaton

محتوى الفصل

2. الحاسبة المنتهية (الأتومات المنتهي **Finite Automaton**)

1.2. الحاسبة المنتهية المحددة **(Deterministic Finite Automaton)**

2.2. تكافؤ الحاسبات (Equivalence Automatons).

3.2. اختزال الحاسبة (Reducibility Automaton).

4.2. الحاسبة المنتهية غير المحددة **non Deterministic Finite Automaton)** (< NFA>).

أهداف الفصل

يهدف الفصل إلى اكتساب المعارف التالية :

مفهوم الحاسبة المنتهية ، طرق تمثيل الحاسبة ، اللغة الممثلة بحاسبة منتهية ، تكافؤ الحاسبات ، اختزال الحاسبات ، تحليل الحاسبات ، الشكل القياسي للحاسبة ، الحاسبة المنتهية غير المحددة <NFA > ، تكافؤ حاسبة منتهية محددة مع حاسبة منتهية غير محددة .

1.2. الحاسبة المنتهية المحددة

(Deterministic Finite Automation).

تبدأ نظرية الحوسبة باستخدام حاسوب مثالي يسمى النموذج الحاسوبي Computational Model. لأن الحاسوب الحقيقي معقد جداً لدرجة أنه لا يسمح بتطبيق النظريات الرياضية عليه مباشرة. من أجل هذا نعرف نموذج التشغيل الذاتي المحدد بأنه مكوّن من مجموعة أجزاء هي: مجموعة الحالات ، مجموعة قواعد الانتقال من حالة إلى حالة أخرى وفق عنصر الإدخال ، أبجدية إدخال تبين أية عناصر يسمح بها كمدخلات، حالة ابتدائية وكذلك مجموعة حالات القبول(الحالات النهائية أو الهدفية).

يستخدم التعريف الشكلي ما يسمى دالة الانتقال Transition Function ويرمز لها بـ δ وذلك لتعريف قواعد الانتقال. مثلاً ، إذا كان المخطط يظهر عملية الانتقال من حالة x إلى حالة y بواسطة سهم وذلك من أجل عنصر الدخل 1 ، فهذا يعني أنه إذا كان النموذج في الحالة x فعند قراءة رقم الدخل 1 فسوف ينتقل إلى الحالة y . ويتم التعبير عن هذا الفعل باستخدام دالة النقل بالشكل

$\delta(x, 1) = y$.

إن هذا التعبير يمثل الاختصار الرياضي للحدث، وعند وضع كل المكونات مع بعضها البعض فإننا نصل إلى التعريف الشكلي لنموذج التشغيل الذاتي المحدود الذي نسميه اختصارا الحاسبة المنتهية المحددة (الأتومات المنتهي المحدد).

1.1.2. تعريف .

نعرف الحاسبة المنتهية المحددة (Deterministic Finite Automaton)

< DFA > بأنها كل خماسية بالشكل : $(S , I , \delta , s_0 , F)$ ، حيث :

S مجموعة منتهية ، تسمى مجموعة الحالات (States set).

I مجموعة منتهية ، تسمى مجموعة الدخل أو أبجدية الدخل (Alphabet).

δ دالة النقل وتعرف بالشكل

$\delta : S \times I \rightarrow S$ (Transition Function).

$s_0 \in S$ الحالة الابتدائية (initial State) .

F مجموعة الحالات النهائية (الهدفية) $F \subseteq S$ (Final states).

يمكننا استخدام رموز التعريف الشكلي ، وهذا يعطي وصفاً دقيقاً لنموذج التشغيل الذاتي المحدد. فإذا رمزنا ب M للنموذج المعرف أعلاه فيمكن وصف هذا النموذج بالشكل :

$M = (S , I , \delta , s_0 , F)$ ، وندعوه حاسبة منتهية محددة أو أو تومات منتهٍي محدد .

2.1.2. مثال .

لتكن $M=(S , I , \delta , s_0 , F)$ حاسبة منتهية محددة معرفة بالشكل التالي :

$S = \{ s_0 , s_1 , s_2 \}$, $I = \{ 0 , 1 \}$, $F = \{ s_2 \}$

$\delta (s_0 , 0) = s_1$, $\delta (s_1 , 0) = s_2$, $\delta (s_2 , 0) = s_2$

$\delta (s_1 , 1) = s_0$, $\delta = (s_2 , 1) = s_2$, $\delta(s_0 , 1) = s_0$

من الواضح أن M تمثل حاسبة منتهية محددة .

يمكن أن نتصور مفهوم الحاسبة المنتهية المحددة كنظام اختياري بغض النظر عن طابعه الفيزيائي، حيث تكون حالة النظام ودوافعه الخارجية التي تؤثر عليه وحيدة التعيين. كما يفترض التعريف 1.1.2. بأنه مخصص في النظام حالة معينة كحالة ابتدائية ومجموعة حالات نهائية (هدفيه).

3.1.2. طرق تمثيل الحاسبة المنتهية المحددة

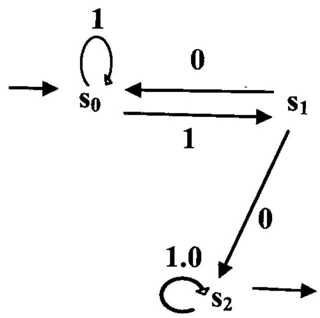
يمكن تمثيل بنية حاسبة منتهية محددة معينة بشكل مناسب . وسنقدم في هذا الكتاب ثلاث طرق أساسية لتمثيل الحاسبة ، نستعرضها على الحاسبة المعطاة في المثال 2.1.2.

a) جدول الدالة (function table) :

δ	0	1
$\rightarrow s_0$	s_1	s_0
s_1	s_2	s_0
$\leftarrow s_2$	s_2	s_2

يعطي هذا الجدول دالة نقل الحاسبة المعطاة في المثال، حيث يشار إلى الحالة الابتدائية بالسهم $\rightarrow$ ، ويشار إلى الحالة النهائية بالسهم $\leftarrow$. وعندما تكون الحالة الابتدائية ضمن مجموعة الحالات النهائية يشار إليها بالرمز $\leftrightarrow$.

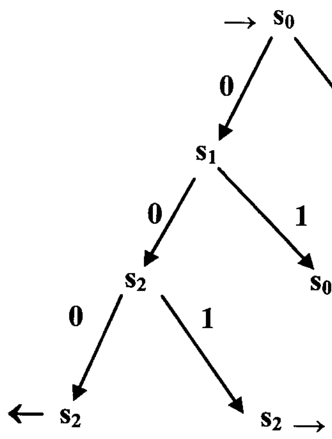
مخطط الحالات **(States Diagram) .**



(شكل 2.1)

إن رؤوس المخطط تقابل حالات الحاسبة ، كما تمثل الأسهم التنقلات الممكنة بين الحالات حسب قيم أبجدية الدخل . أما الحالة البدائية والحالات النهائية فيشار إليها كما في الجدول السابق .

b) شجرة الحالات (States Tree) .



(شكل 2.2)

من كل رأس لا يمثل ورقة في شجرة الحالات يخرج عدد من الأسهم يساوي عدد رموز الدخل . يقابل جذر الشجرة الحالة الابتدائية للحاسبة. أما الرؤوس التالية فتقيّم بالتوافق مع دالة النقل . ينطلق من الرأس أسهم فقط إذا لم يكن أي رأس مقيّم أ سابقاً بللرمز نفسه. وإذا ظهرت حالة لأول مرة على بعد ما من الجذر عند عدد من الرؤوس بللوقت نفسه، يكفي إخراج سهم من أحد هذه الرؤوس .

من المعلوم ، أنه لا يمكن تمثيل كل حاسبة بشجرة حالات كما وصفنا أعلاه، ولكن الأمثلة على ذلك عملياً غير مهمة كثيراً (أنظر 1.3.2 ، 2.3.2). كذلك يرمز للحالات النهائية بشكل مشابه للطرق السابقة. أما الحالة الابتدائية فليس من الضروري تمييزها. من الملاحظ أن مخطط الحالات ينتج من شجرة الحالات ، حيث تطابق الرؤوس المقيّمة بللرمز نفسه.

من خلال وصف الحاسبة يكون من السهل معرفة تأثيرها على الأحداث، أي السلاسل المختلفة لرموز الدخل . مثلاً : الحاسبة المعطاة في المثال 2.1.2 تعمل وفق سلسلة الرموز 011001 ، حيث تصل من الحالة الابتدائية s_0 وبالتدريج إلى الحالات s_1 , s_1 , s_0 , s_0 , s_1 , s_2 . إن سلسلة التغيير تقود الحاسبة من الحالة الابتدائية s_0 إلى الحالة النهائية s_2 . وإن السلاسل بهذه الخصائص كثيرة جداً. بتفحص دقيق لهذه الحاسبة نتحقق أن جميع هذه السلاسل تحوي السلسلة الجزئية 00 والتي تسمى السلسلة الصغرى.

يمكن توضيح سلاسل الدخل بطرق متعددة، أبسطها تصور سلسلة رموز الدخل بشكل تتالي الأحداث التي عملت عليها الحاسبة. بعض هذه الأحداث تقود الحاسبة من الحالية البدائية إلى بعض الحالات النهائية.

نستطيع الآن صياغة هذه المفاهيم ، حيث نتحدث عن الكلمات بدلاً من الأحداث، وعن اللغات بدلاً من مجموعات الأحداث . إن هذه المصطلحات تؤكد فاعلية الحاسبة ويتضح ذلك أثناء تعريف قواعد اللغة البرمجية في الفصول اللاحقة.

4.1.2. **تعريف** .

لتكن I مجموعة منتهية محددة اختيارية (أبجدية)، نرمز بـ I^+ لمجموعة جميع السلاسل المنتهية غير الخالية والمشكلة من عناصر المجموعة I . ونرمز ب ε للسلسلة الخالية .

نعرف I^* بالشكل $I^* =_{df} I^+ \cup \{\varepsilon\}$.

سنتحدث عن I^* كمجموعة جميع الكلمات فوق الأبجدية I ، وعن I^+ كمجموعة الكلمات غير الخالية فوق I . كما نرمز للسلسلة:

$i_1 , i_2 , \ldots, i_n \in I^*$

بالشكل $i_1 i_2 \ldots i_n \in I^*$.

وسندعو كل سلسلة من هذا النوع كلمة في الأبجدية I ، وندعو ε الكلمة الخالية.

إذا كانت $u = i_1 i_2 \ldots i_n \in I^*$ ، $v = j_1 j_2 \ldots j_m \in I^*$ ، عندئذ ندعو الكلمة

$uv = i_1\, i_2 \dots i_n\, j_1\, j_2 \dots j_m$ متتالية الكلمتين u , v . و بشكل خاص يكون $\varepsilon u = u \varepsilon = u$.

5.1.2. **تعريف** .

إذا كانت I أبجدية منتهية و $L \subseteq I^*$ ، عندئذ نسمي L لغة على الأبجدية I .

يمكن عرض مثالاً على ذلك اللغة العربية ، فهي تتشكل من أبجدية منتهية ، يتشكل منها كلمات هذه اللغة . ومن كلمات اللغة العربية تتشكل الجمل الصحيحة المركبة وفق قواعد اللغة المعتمدة. وينتمي إلى أبجدية اللغة العربية إضافة إلى رموز أحرفها ، رموز الأرقام ورموز اعتراضية أخرى ورمز الفراغ .

6.1.2. تعريف .

لتكن $M = (S , I , \delta , s_0 , F)$ حاسبة منتهية محددة . نوسع الدالة δ إلى دالة نقل عامة بالشكل:

: $\delta = S \times I^* \to S$

$$1) \quad \delta (s , \varepsilon) = s , \quad \forall s \in S ;$$

$$2) \quad \delta (s , wi) =_{df} \delta (\delta (s , w) , i) \quad ; \forall\ i \in I , s \in S , w \in I^* .$$

إن اللغة الممثلة (المعرفة) بالحاسبة M ، تسمى لغة وتعرف بالعلاقة :

$$L (M) = \{ w ; w \in I^* \ \& \ \delta (s_0 , w) \in F \} .$$

نقول إن الكلمة $w \in I^*$ مقبولة بالحاسبة M فقط إذا كان $w \in L (M)$.

7.1.2. تعريف .

نقول إن اللغة L ممثلة بحاسبة منتهية محددة، إذا كان يوجد حاسبة منتهية محددة M ، حيث يكون

$L = L(M)$. وسنرمز فيما يلي بـ Σ لمجموعة جميع اللغات الممثلة بالحاسبة المنتهية المحددة .

تطبيق (1.2.) .

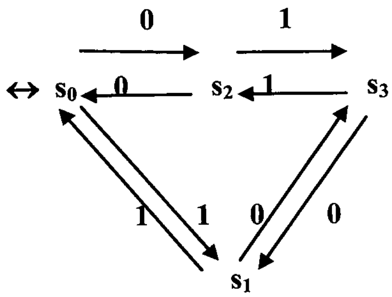
لتكن M حاسبة منتهية محددة معطاة بالشكل

$$M = (\{ s_0 , s_1 , s_2 , s_3 \}, \{ 0 , 1 \} , \delta , s_0 , \{ s_0 \})$$

حيث دالة النقل ممثلة بالجدول التالي :

δ	0	1
$\leftrightarrow s_0$	s_2	s_1
s_1	s_3	s_0
s_2	s_0	s_3
s_3	s_1	s_2

لنمثل الحاسبة M بمخطط (بيان موجه، رؤوسه تمثل حالات الحاسبة وأضلاعه تمثل الأحداث الجارية)



(شكل 3.2.)

لنتحقق الآن من قبول أو عدم قبول الحاسبة M للكلمتين 001100 , 01110 .

$$\delta(s_0\,,01110\,) = \delta(\delta(\,s_0\,,0111),0) = \delta(\delta(\delta(\,s_0\,,011),1),0) =$$

$$= \delta(\delta(\delta(\delta(\,s_0,01),1),1),0) = \delta(\delta(\delta(\delta(\delta(\,s_0,0),1),1),1),0) =$$

$$= \delta(\delta(\delta(\delta(\,s_2,\,1),1)1),0) = \delta(\delta(\delta\,(\,s_3,\,1),1),0) =$$

$$= \delta(\delta(\,s_2,\,1),0) = \delta(\,s_3,0) = s_1 \notin F\,.$$

ينتج أن الكلمة 01110 غير مقبولة بالنسبة للحاسبة M .

لنتحقق بأن الكلمة 001100 مقبولة بالنسبة للحاسبة M بطريقة أخرى :

$$\rightarrow \mathbf{s_0} \Rightarrow_0 \mathbf{s_2} \Rightarrow_0 \mathbf{s_0} \Rightarrow_1 \mathbf{s_1} \Rightarrow_1 \mathbf{s_0} \Rightarrow_0 \mathbf{s_2} \Rightarrow_0 \mathbf{s_0} \in \mathbf{F}\,.$$

ينتج أن الكلمة 001100 مقبولة بالنسبة للحاسبة M .

تطبيق (2.2) .

أنشئ حاسبة منتهية محددة تقبل جميع السلاسل من الأبجدية { 1 , 0 } والتي تمثل التمثيل الثنائي للأعداد التي تقبل القسمة على 5 .

لحل المسألة ، نشكل مجموعة الحالات التي تحوي الحالات التالية :

s_0 : تمثل الحالة البدائية .

s_1 : تمثل الحالة عندما يكون $(x \bmod 5 = 1)$.

s_2 : تمثل الحالة عندما يكون $(x \bmod 5 = 2)$.

s_3 : تمثل الحالة عندما يكون $(x \bmod 5 = 3)$.

s_4: تمثل الحالة عندما يكون $(x \bmod 5 = 4)$.

s_5 : تمثل الحالة عندما يكون $(x \bmod 5 = 0)$ ، وتمثل الحالة النهائية.

لنفرض أن الحاسبة موجودة في الحالة s_m . هذا يعني أن العدد الذي تعّرفه الحاسبة حتى الآن هو x ، حيث $x \bmod 5 = m$ وعدد الخانات الثنائية فيه يساوي $| x |$. وباعتبار أن الحاسبة محددة يجب أن ينطلق من الحالة s_m حدثان :

1) حدث موافق لرمز الدخل 0 ينقل الحاسبة إلى الحالة s_n .

2) حدث موافق لرمز الدخل 1 ينقل الحاسبة إلى الحالة s_p .

عند الوصول إلى الحالة s_n ، يصبح عدد الخانات الثنائية في العدد الجديد y يساوي $| y | = | x | +1$ وقيمته $y = 2 x$ (انزياح خانة ثنائية إلى اليسار وإضافة 0) .

ويكون :

$$n = y \bmod 5 \Rightarrow n = 2x \bmod 5 \Rightarrow n = 2 (x \bmod 5) \bmod 5 \Rightarrow$$

$$n = 2m \bmod 5 .$$

في حين ، عند الوصول إلى الحالة s_p ، يصبح عدد الخانات الثنائية في العدد الجديد y يساوي $| y | = | x | +1$ وقيمته $y = 2 x+ 1$

(انزياح خانة ثنائية إلى اليسار وإضافة 1) .

ويكون :

$$p = y \bmod 5 \Rightarrow p = (2x + 1) \bmod 5 \Rightarrow$$

$$p = 2\,(x \bmod 5) \bmod 5 + 1 \Rightarrow$$

$$p = 2m \bmod 5 + 1 .$$

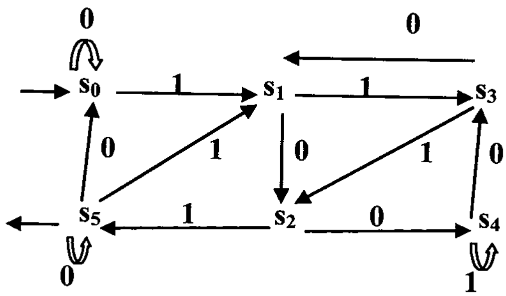
بتعويض m بالقيم 5 , 4 , 3 , 2 , 1 , 0 تحسب جميع قيم أرقام الحالات الناتجة عند الحدثين 0 ، 1 (أي جميع قيم n , p) كما يظهر في الجدول التالي :

m	0	1
0	0	1
1	2	3
2	4	5
3	1	2
4	3	4
5	0	1

ويصبح جدول دالة النقل كما يلي :

δ	0	1
s_0	s_0	s_1
s_1	s_2	s_3
s_2	s_4	s_5
s_3	s_1	s_2
s_4	s_3	s_4
s_5	s_0	s_1

ويكون مخطط الحاسبة بالشكل :



(شكل 4.2)

8.1.2. تمارين .

1.1.2. عين سلسلة الحالات ، التي تنتقل إليها الحاسبة المعطاة في المثال 2.1.2 أثناء شغل كلمتي الدخل الآتيتين:

101100111 , 0110111001 .

2.1.2. ميزنا اللغة الممثلة بالحاسبة المعطاة في المثال 2.1.2. كمجموعة جميع الكلمات على الأبجدية { 0 , 1 } التي تحوي الكلمة الجزئية 00 . أوجد الخصائص المشابهة للغات الممثلة بالحاسبات الآتية:

(a

δ	0	1
↔ s_0	s_1	s_0

s_1	s_2	s_1
s_2	s_0	s_2

(b

δ	0	1
$\rightarrow s_0$	s_1	s_0
$\leftarrow s_1$	s_2	s_1
$\leftarrow s_2$	s_0	s_2

(c

δ	0	1
$\rightarrow s_0$	s_0	s_1
s_1	s_0	s_2
$\leftarrow s_2$	s_0	s_2

(d

δ	0	1
$\rightarrow s_0$	s_0	s_1
$\leftarrow s_1$	s_2	s_1
$\leftarrow s_2$	s_0	s_1

(e

δ	0	1
$\rightarrow$ A	B	A
B	A	C
$\leftarrow$ C	B	C

(f

δ	0	1
$\leftrightarrow 1$	1	2
2	1	3
$\leftarrow 3$	2	3

3.1.2. شكّل لكل من اللغات الآ تية الحاسبة التي تمثلها، حيث تكون أبجدية الدخل في جميع الحالات $\{0 , 1\}$.

تتشكل هذه اللغات من جميع الكلمات التي تحقق الشروط المبينة أدناه :

a) عدد الأصفار التي تحويها ، قابل للقسمة على 3.

b) عدد الواحدات التي تحويها ، قابل للقسمة على 2 أو 3 .

c) الكلمة تنتهي بتكرارين أو أكثر من الرقم 1 مسبوقة بالرقم 0 مرة واحدة على الأقل.

d) الرمزان الأوّلان في الكلمة متماثلان مع الرمزين الأخيرين.

e) الكلمة تنتهي بالرمز 1 وطولها $3k + 1$ من أجل $k \geq 1$.

4.1.2. لتكن M حاسبة تقبل جميع الكلمات في أبجدية معينة، أطوالها قابلة للقسمة على 5 . كم عدد الحالات الأصغر لهذه الحاسبة ؟

5.1.2. اكتب طريقة ، تستطيع من خلالها بناء برنامج لكل الحاسبات المنتهية التي تؤدي الدور نفسه.

6.1.2. يمكن تمثيل دالة النقل الموسعة δ لحاسبة اختيارية

$$M = (S , I , \delta , s_0 , F)$$

بمصفوفة النقل ، حيث نعرف لكل كلمة فيها $u \in I^*$ مصفوفة $M(u)$ كما يلي :

δ	s_1	....	s_j	....	s_n
s_1			...		
....			...		
s_1	...	...	a_{ij}	...	...
....			...		
s_n			...		

حيث :

$$S = \{ s_1 , s_2 , \dots , s_n \}$$

$$a_{ij} = \begin{cases} 1 \ ; & \delta(si , u) = sj \ , \\ 0 \ ; & \delta(si , u) \neq sj \ . \end{cases}$$

إذاً تكون كل مصفوفة $M (u)$ بالشكل $n \times n$ (n عدد عناصر S) ، حيث يكون العنصر في السطر – i والعمود – j مساويا 1 فقط عندما تنقل كلمة الدخل u الحاسبة M من الحالة s_i إلى الحالة s_j .

a) برهن أن لكل حاسبة M ، تكون $M (\varepsilon)$ مصفوفة أحادية .

b) برهن أنه لتعيين $M (u)$ بالنسبة لأي $u \in I^*$ ، يكفي معرفة $M (v)$ فقط من أجل $v \in I$.

اقتراح لحل المسألة :

إذا عدلنا الخوارزمية المعتادة لضرب المصفوفات ، حيث نأخذ بدلاً عن عملية الجمع + العملية $\oplus$ المعرفة كما يلي:

$$0 \oplus 0 = 0 \ ,$$

$$0 \oplus 1 = 1 \oplus 0 = 1 \oplus 1 = 1 \;;$$

عندئذ ، من أجل كل $u , v \in I^*$ ، تتحقق العلاقة

$$M(u)\,M(v) = M(u , v) \, .$$

وهذا أيضا يتحقق من أجل الحاسبة المنتهية غير المحدودة .

2.2 . تكافؤ الحاسبات . (Equivalence Automations)

1.2.2. تعريف :

لتكن $M \;,\; N$ حاسبتين منتهيين ، نقول إن M , N متكافئتان إذا كان

$$L(M) = L(N) \, .$$

ينتج من التعريف مباشرة ، أنه لا يمكن لحاسبتين أن يكونا متكافئتان ، إذا كانت اللغات الممثلة لهما غير منتهية. يمكن في كل صف تكافؤ حاسوبي إيجاد حاسبة أصغرية . وسنعرض الآن بعض خصائص الحاسبة المنتهية ، التي تساعدنا في برهان بعض الحقائق.

2.2.2. تعريف :

ليكن $M = (S , I , \delta , s_0 , F)$ حاسبة منتهية و $s \;,\; r \in S$. نقول أن الحالتين s , r متكافئتين ونرمز لهما $s \sim r$ ، إذا كان

$$\delta(s , w) \in F \Leftrightarrow \delta(r , w) \in F \;,\quad \forall\, w \in I^* \, .$$

تمثل العلاقة $\sim$ علاقة تكافؤ على S .

3.2.2. تعريف :

لتكن $M = (S , I , \delta , s_0 , F)$ حاسبة منتهية ، ولتكن $\equiv$ علاقة تكافؤ على M . عندئذ تسمى العلاقة $\equiv$ تتطابق حاسوبي Isomorphism إذا تحقق الشرطان:

$$\forall s , r \in S \ , \forall i \in I :$$

$$1) \quad s \equiv r \Rightarrow [s \in F \Leftrightarrow r \in F],$$

$$2) \quad [\delta (s , i) \equiv \delta (r , i)] .$$

4.2.2. مبرهنة :

إن تكافؤ الحالات المقدّم في التعريف 2.2.2. يشكل تطابقاً حاسوبياً .

البرهان :

$$s \sim r \Leftrightarrow^{(1)} (\forall w \in I^*) \ [\delta (s ,w) \in F \Leftrightarrow \delta (r ,w) \in F \ \Rightarrow^{(2)}$$

$$(\forall w \in I^+) \ [\delta (s ,w) \in F \Leftrightarrow \delta (r ,w) \in F \ \Leftrightarrow^{(3)}$$

$$(\forall i \in I) \ (\forall w \in I^*) [\delta (s ,iw) \in F \Leftrightarrow \delta (r ,iw) \in F \ \Leftrightarrow^{(4)}$$

$$(\forall i \in I) \ (\forall w \in I^*) \ [\delta (\delta(s , i),w) \in F \Leftrightarrow$$

$$\delta (\delta (r , i) , w) \in F \ \Leftrightarrow^{(5)}$$

$$(\forall i \in I) \ [\delta (s , i) \equiv \delta (r , i)] .$$

إن علاقات التكافؤ (1) ، (5) تنتج مباشرة من تعريف ~ ' والاقتضاء (2) والتكافؤ (3) تنتج من علاقة I^{+} , I^{*} ، أما التكافؤ (4) فينتج من تعريف توسيع دالة النقل . *

5.2.2. تعريف :

لتكن $M_1 = (S_1 , I , \delta_1 , s_1 , F_1)$ ، $M_2 = (S_2 , I , \delta_2 , s_2 , F_2)$ حاسبتين محددين . عندئذ تسمى الدالة $h : S_1 \rightarrow S_2$ تشاكلاً تبولوجياً Homomorphism للحاسبتين M_2 , M_1 إذا وفقط إذا تحققت العلاقات الآتية :

1) $h (s_1) = s_2$,

2) $h (\delta_1 (s , i)) = \delta_2 (h (s) , i) \; ; \; \forall \; s \in S_1 , i \in I$,

3) $s \in F_1 \Leftrightarrow h (s) \in F_2 \; ; \qquad \forall \; s \in S_1$.

نقول إن h تشاكل تبولوجي لـ S_1 على S_2 ، إذا كانت h دالة غامرة لـ S_1 على S_2 .

6.2.2. مبرهنة :

لتكن $M = (S , I , \delta , s_0 , F)$ حاسبة محددة و $\equiv$ تطابقاً حاسوبياً على S ، عندئذ :

1) تمثل الخماسية

$$M/_{\equiv} =_{df} (S/_{\equiv} , I , \delta_{\equiv} , [s_0]_{\equiv} , \{ [s]_{\equiv} \}_{\, s \in F}) ,$$

حيث $\delta_{\equiv}$ معرفة بالعلاقة $\delta_{\equiv} ([s]_{\equiv} , i) = [\delta (s , i)]_{\equiv}$ ، حاسبة محددة.

2) تمثل الدالة $h : S \rightarrow S/_\equiv$ المعرفة بالشكل :

$$\forall s \in S \; : h(s) = [s]_\equiv \,,$$

تشاكل تبولوجي حاسوبي لـ M على $M/_\equiv$. وتسمى الحاسبة $M/_\equiv$ حاسبة توزيع لـ M وفق العلاقة $\equiv$.

البرهان :

1) برهان أن $M/_\equiv$ تشكل حاسبة محددة ، ينتج ذلك مباشرة من تعريف الدالة

$$\delta_\equiv : S/_\equiv \times I \rightarrow S/_\equiv$$

حيث إن : $s \equiv r \Rightarrow \delta(s,i) \equiv \delta(r,i), \; \forall s \in S, i \in I$.

2) حسب تعريف h و $\delta_\equiv$ نجد أن :

$$h(\delta(s,i)) = [\delta(s,i)]_\equiv = \delta_\equiv([s]_\equiv \,, i) = \delta_\equiv(h(s), i)$$

ينتج أن h تمثل تشاكلاً تبولوجياً حاسوبياً ، لأن الخاصتين الأخيرتين للتشاكل التبولوجي الحاسوبي تنتجان بالنسبة لـ h مباشرة من تعريف $M/_\equiv$. *

7.2.2. مبرهنة :

ليكن h تشاكلاً تبولوجياً حاسوبياً للحاسبة $M_1 = (S_1, I, \delta_1, s_1, F_1)$ على الحاسبة $M_2 = (S_2, I, \delta_2, s_2, F_2)$ ، عندئذ يكون :

$$h(\delta_1(s,w)) = \delta_2(h(s), w) \;, \; \forall s \in S_1 \;, \; i \in I^* \,.$$

البرهان: نبرهن على هذا التأكيد بالتحريض حسب طول الكلمة w .

1) من أجل كلمة بطول 1 على الأكثر ، ينتج البرهان مباشرة من تعريف التشاكل التبولوجي .

2) نفرض الآن أن التأكيد محقق من أجل جميع الكلمات بطول k . ولتكن w كلمة بطول $k + 1$. عندئذ يمكن كتابة w بالشكل $w = vi$ ، حيث v كلمة بطول k .

حسب تعريف h وحسب فرض التحريض يكون

$$h(\delta_1(s, w)) = h(\delta_1(s, v), i) = \delta_2(h(\delta_1(s, v)), i) =$$

$$= \delta_2(\delta_2(h(s), v), i) = \delta_2(h(s), w) . *$$

8.2.2. **نتيجة** :

إذا وجد تشاكل تبولوجي حاسوبي من الحاسبة M_1 إلى الحاسبة M_2 ، عندئذ تكون الحاسبتان M_1 و M_2 متكافئتين .

البرهان . ليكن:

$$M_1 = (S_1, I, \delta_1, s_1, F_1) \quad \text{و} \quad M_2 = (S_2, I, \delta_2, s_2, F_2)$$

وليكن $h : S_1 \rightarrow S_2$ تشاكل تبولوجي .

حسب تعريف التشاكل التبولوجي الحاسوبي يكون $h(s_1) = s_2$. وحسب المبرهنة السابقة يكون

$$h(\delta_1(s_1, w)) = \delta_2(h(s_1), w) = h(\delta_1(s_1, w)) ,$$

$$\forall w \in L(M_1) .$$

ويكون بالتالي $w \in L(M_1) \Leftrightarrow \delta_1(s_1, w) \in F_1$ ،

وهذا يكافئ حسب التعريف 5.2.2.(3)

$$h(\delta_1(s_1, w))\ F_2 \Leftrightarrow \delta_2(s_2, w) \in F_2 \Leftrightarrow w \in L(M_2).$$

9.2.2. مبرهنة :

لتكن M حاسبة منتهية و ~ علاقة تكافؤ على فضاء حالات M (راجع 2.2.2.)، عندئذ تكون :

1) حاسبة التوزيع $M/_\sim$ مكافئة لحاسبة M .

2) لا تكون أي حالتين مختلفتين للحاسبة $M/_\sim$ متكافئتين .

البرهان.

1) التحقق من الجزء الأول من المبرهنة ينتج مباشرة من (4.2.2 ، 6.2.2 ، 8.2.2) .

2) لبرهان الجزء الثاني ، نفرض أن $[r]_\sim$ يكافئ $[s]_\sim$ ، وهذا يعني أنه

$$(\forall w \in I^*)\ [\delta_\sim([r], w) \in F_\sim \Leftrightarrow [\delta_\sim([s], w) \in F_\sim].$$

وهذا يعني حسب تعريف $M/_\sim$ أنه

$$(\forall w \in I^*)\ [\delta(r, w)]_\sim \in F_\sim \Leftrightarrow [\delta(s, w)]_\sim \in F_\sim].$$

ومن المعروف أن كل حالة مكافئة لحالة نهائية يجب أن تكون حالة نهائية . ينتج أنه

$$(\forall w \in I^*)\ [\delta(r, w) \in F \Leftrightarrow \delta(s, w) \in F],$$

أي أن $r \sim s$. وبذلك يتم البرهان على $[r] \sim [s] \Rightarrow [r] = [s]$.

ينتج من المبرهنة الأخيرة ، أن كل لغة يمكن تمثيلها بحاسبة منتهية يمكن أيضا تمثيلها بحاسبة منتهية لا يكون فيها أي حالتين متكافئتان . لذا، وجود عدد من الحالات

المتكافئة في حاسبة واحدة يزيد حجم هذه حاسبة دون فائدة. ولتصغير حجم فضاء الحالات لحاسبة ما دون التأثير على إمكانيات تميز هذه الحاسبة ، يمكن وضع بعض هذه الحالات بطريقة لا يمكن بلوغها عبر أية طريقة حسابية.

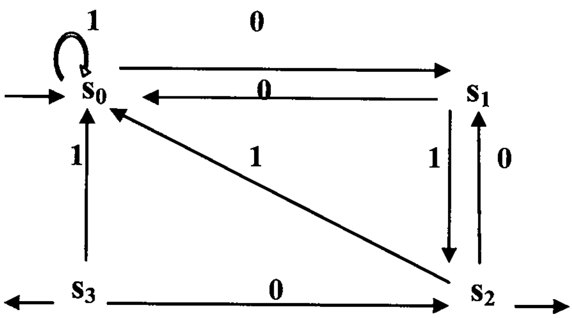
3.2. اختزال الحاسبة. (Reducibility Automaton)

1.3.2. تعريف.

نقول إن الحالة s للحاسبة $M = (S , I , \delta , s_0 , F)$ قابلة للبلوغ ، إذا وجد كلمة $w \in I^*$ ، حيث $\delta (s_0 , w) = s$. أما الحالات التي لا يمكن قبولها فتسمى حالات غير قابلة للبلوغ .

2.3.2. مثال .

لتكن M حاسبة معطاة بمخطط الحالات شكل 3.2. من الواضح أن الحالات القابلة للبلوغ هي s_0 , s_1 , s_2 ، أما الحالة s_3 فهي غير قابلة للبلوغ.



(شكل 3.2)

3.3.2. مبرهنة .

لتكن $M = (S, I, \delta, s_0, F)$ حاسبة منتهية محددة ، ولتكن $A \subseteq S$ مجموعة الحالات القابلة للبلوغ في الحاسبة M . عندئذ الحاسبة $N = (A, I, \delta_A, s_0, F \cap A)$ ، حيث $\delta_A =_{df} \delta / A \times I$ تجزئة للدالة δ على $A \times I$ ، تشكل حاسبة مكافئة للحاسبة M ولا تحوي حالات غير قابلة للبلوغ .

البرهان : من الواضح أن جميع الحالات التي تمر بها الحاسبة قبل أن تصل إلى الحالة القابلة للبلوغ تكون أيضا قابلة للبلوغ . ينتج أن

$$(\forall w \in I^*) \; [\delta(s_0, w) = \delta_A(s_0, w)] \qquad (1)$$

إن مجموعة الحالات النهائية في M ، التي يمكن الوصول إليها من الحالة البدائية ، تساوي المجموعة $F \cap A$ وبالتالي

$$(\forall w \in I^*) \; [\delta(s_0, w) \in F \Leftrightarrow \delta_A(s_0, w) \in F \cap A]$$

وبذلك يكون $L(M) = L(N)$. *

4.3.2. تعريف.

نقول إن الحاسبة M مختزلة (Reductive) ، إذا كانت تحقق الشرطين التاليين :

1. لا تحوي حالات غير قابلة للبلوغ ،
2. لا تكون أي حالتين مختلفتين منها متكافئتين .

5.3.2. تعريف .

نقول إن الحاسبة N مختزل الحاسبة M ، إذا تحقق الشرطان التاليان:

1. N تكافئ M .
2. N حاسبة مختزلة .

6.3.2. نظرية.

لكل حاسبة منتهية محددة يوجد حاسبة مختزلة .

البرهان : حسب المبرهنة 3.3.2. يوجد لكل حاسبة منتهية محددة M حاسبة مكافئة لها N ، وليس لها أي حالتين متكافئتين. إذا حذفنا من N جميع الحالات غير القابلة للبلوغ ، نحصل وفق المبرهنة 3.3.2 على حاسبة مختزلة للحاسبة N وبالتالي مختزل الحاسبة M . *

من الحائق الهامة والمفيدة ، أن كل مختزل لأي حاسبة يكون وحيد التعيين وبتشاكل تبولوجي محقق .

7.3.2. تعريف .

ليكن h تشاكلاً تبولوجياً حاسوبياً للحاسبة M على الحاسبة N . إذا كانت h دالة متباينة لفضاء حالات M على فضاء حالات N ، عندئذ نسمي h تشاكلاً تاماً حاسوبياً Isomorphism ، ونقول إن M يشاكل تماماً N .

8.3.2. نظرية.

تكون حاسبتان مختزلتان (بنفس أبجدية الدخل) متكافئتين إذا وفقط إذا كانتا متشاكلتين تماماً.

البرهان .

$\Leftarrow$ نفرض أن M , N حاسبتان متشاكلتان تماماً، عندئذ حسب النتيجة 8.2.2. يكونان متكافئتين .

$\Rightarrow$ لنبرهن الآن العكس:

لتكن $M_1 = (S_1 , I , \delta_1 , s_1 , F_1)$ و $M_2 = (S_2 , I , \delta_2 , s_2 , F_2)$ حاسبتين مختزلتين متكافئتين . ونشكل الآن تطبيق غامر $h : S_1 \rightarrow S_2$ ، ولنبرهن أنه تشاكل تبولوجي حاسوبي .

a) تعريف h : لنبرهن أنه لكل $s \in S_1$ يوجد فقط حالة واحدة $r \in S_2$ ، حيث

$$(\forall w \in I^*) \quad [\delta_1 (s , w) \in F_1 \Leftrightarrow \delta_2 (r , w) \in F_2] , \qquad (1)$$

إن وجود حالة r واحدة على الأكثر واضحة ، لأنه لو وجد حالتان r_1 , r_2 ، عندئذ ينتج من (1) أن $r_1 \sim r_2$, وهذا يناقض الفرض بأن M_2 مختزل .

نأخذ الآن حالة اختيارية $s \in S_1$ ، من المؤكد وجود $u \in I^*$ حيث $s = \delta_1 (s_1 , u)$ لأن s قابلة للبلوغ . إذا عرفنا $r = \delta_2 (s_2 , u)$ ، فإن الصيغة (1) تكافئ الصيغة

$$(\forall w \in I^*) \quad [\delta_1 (s_1 ,u w) \in F_1 \Leftrightarrow \delta_2 (s_2 ,u w) \in F_2] ,$$

وهذا محقق طبعاً ، لأن M_1 , M_2 حسب الفرض متكافئان.

لنضع الآن لكل $s \in S_1$ ، $h (s) = r$ ، حيث r , s يحققان (1) ، عندئذ يكون h التطبيق المعرف من S_1 إلى S_2 .

b) h يكون غامراً : من شكل العلاقة (1) ، ينتج أنه لكل $r \in S_2$ يوجد حالة واحدة فقط $s \in S_1$ ، حيث تتحقق العلاقة (1) . من هذا ينتج أن h تطبيق غامر.

c) h تشاكل تبولوجي :

1) من (1) ومن الفرض ، ينتج أن M_1 , M_2 متكافئان . من هذا ينتج أن $h (s_1) = s_2$.

2) لكل $s \in S_1$ ، يوجد $u \in I^*$ حيث $\delta_1 (s_1 , u) = s$.

ويتحقق من أجل كل كلمة $i \in I$ المساواة :

$$h (\delta_1 (s , i) = h (\delta_1 (s_1 , u i) .$$

وحسب بنية h يكون $h (\delta_1 (s_1 , u i) = \delta_2 (s_2 , u i)$.

وبالتالي :

$$h (\delta_1 (s , i) = \delta_2 (\delta_2 (s_2 , u) , i) .$$

وبما أن :

$$\delta_2 (s_2 , u) = h (\delta_1 (s_1 , u)) = h (s) ،$$

فإن:

$$h (\delta_1 (s , i) = \delta_2 (h (s) , i) .$$

3) من (1) ومن تعريف h ، نجد خاصة من أجل $w = \varepsilon$ ، أن

$$s \in F_1 \Leftrightarrow h (s) \in F_2 .$$

وبالتالي يكون h تشاكلاً تبولوجياً *

ينتج مما سبق ، أن مختزل ي حاسبتين منتهيتين متكافئتين يكونا ن متطابقين حتى بالتشاكل التام للحالات . إذا لكل لغة ممثلة بحاسبة منتهية يوجد حاسبة صغرى وحيدة التعيين تمثل هذه اللغة.

إن لهذا التأكيد من الناحية النظرية أهمية كبيرة، ولكن من الناحية العملية ليس له معنى واضح . لأنه من غير المفهوم كيفية إيجاد طريقة تشكيل مختزل الحاسبة . وحتى من غير المعروف فيما إذا كانت هذه الطريقة العامة للاختزال موجودة أم لا .

نبين الآن ، أنه يوجد خوارزمية بسيطة وعملية مستعملة للاختزال. حيث يتبين من النظريات السابقة أنه يمكن تشكيل مختزل لحاسبة معطاة إذا استطعنا التحقق مما يلي:

(1). إذا كانتا حالتين اختياريتين للحاسبة متكافئتين أم لا .

(2). تعيين الحالات غير القابلة للبلوغ .

بعدئذ نجمّع الحالات المتكافئة للحاسبة في حالة واحدة (أنظر المبرهنة 9.2.2.) ، ثم نحذف من الحاسبة الناتجة جميع الحالات غير القابلة للبلوغ (أنظر المبرهنة 3.3.2.). لهذا يكفي إيجاد خوارزميتين لحل المشكلتين السابقتين (1) ،(2) . ونبدأ بخوارزميه لتعيين الحالات غير القابلة للبلوغ. من المعلوم أن الحالات القابلة للبلوغ في الحاسبة هي الحالات التي يؤدي إليها مسار يبدأ من الحالة الابتدائية في مخطط الحالات . ولتحقيق هذا الهدف نقدم الخوارزمية التالية :

9.3.2. خوارزمية (Algorithim).

لتكن $M = (S , I , \delta , s_0 , F)$ حاسبة منتهية .

خطوة 1 : ضع $S_j := \{ s_0 \}$; $j := 0$;

خطوة 2 : repeat

$$S_{j+1} := S_j \cup \{ s ; (\exists r \in S_j)(\exists i \in I) : [\delta(r,i) = s] \} ;$$

$$J := j + 1 ;$$

Until $S_j = S_{j+1}$;

A:=S_j;

10.3.2. تأكيد .

إذا بدأت الخوارزمية 9.3.2. العمل على حاسبة منتهية محددة اختيارية $M = (S, I, \delta, s_0, F)$ ، فإنها تنتهي من أجل $j = n$ عملية على الأكثر ، حيث n عدد حالات الحاسبة M . والمجموعة A الناتجة تحوي تماماً جميع الحالات القابلة للبلوغ في الحاسبة M .

البرهان .

1. من الواضح أن $S_0 \subseteq S_1 \subseteq \ldots \subseteq S_{n-1} \subseteq S_n$. إذا كان $S_0 \subset S_1 \subset \ldots \subset S_{n-1} \subset S_n$ ، فإن S_n تحوي $n + 1$ عنصراً على الأقل . وهذا يناقض الفرض كون $S_n \subseteq S$.

2. بقي أن نبرهن أن A تحوي جميع الحالات القابلة للبلوغ في الآلة M.

لتكن $s \in S$ حالة قابلة للبلوغ ، عندئذ تكون المجموعة $K_s = \{ w \in I^* ; \delta(s_0, w) = s \}$ غير خالية . نختار من K_s كلمة ما $w = i_1 \ldots . i_k$ بطول أصغري ، ولنبرهن أن $k \leq j$ و $s \in S_k$.

كل مجموعة S_p تتكون من جميع الحالات في S ، والتي تكون قابلة للبلوغ من s_0 وفق كلمة ما بطول p على الأكثر . إذاً يكفي البرهان بشكل خاص أن $k \leq j$.

نفرض مؤقتا العكس ، أي أن $k > j$. عندئذ لا تكون الحالة $\delta (s_0 , i_1 \; i_2 \; \ldots \; i_{j+1})$ قابلة للبلوغ بأي كلمة بطول أصغر من $j + 1$. إذا كانت قابلة للبلوغ بكلمة ما u بطول $| u | \leq j$ ، عندئذ تكون الحالة s قابلة للبلوغ بكلمة طولها أصغر من k ، وهذا يتناقض مع طريقة اختيار w .

لتكن $v \in I^*$ كلمة ما ، بحيث $w = i_1 \ldots . i_{j+1} v$ عندئذ يكون :

$$S = \delta (s_0 , w) = \delta (\delta (s_0 , i_1 \ldots . i_{j+1}) , v) = \delta (\delta (s_0 , u) , v) =$$

$$= \delta (s_0 , uv) , \quad | uv | < k .$$

وبالتالي $\delta (s_0 , i_1 \; i_2 \; \ldots \; i_{j+1}) \in S_{j+1} - S_j$. من الفرض $k > j$ ، نستنتج أن $S_j \subset S_{j+1}$ ، وهذا يناقض الفرض . وبالتالي يكون $k \leq j$.

وبذلك يتم برهان التأكيد . *

11.3.2. مثال .

بتطبيق الخوارزمية 9.3.2. على الحاسبة المعطاة في المثال 2.3.2 ، ينتج أن :

$$S_0 = \{ s_0 \} \; , \; S_1 = \{ s_0 , s_1 \} \; , \; S_2 = \{ s_0 , s_1 , s_2 \} \; ,$$

$$S_3 = \{ s_0 , s_1 , s_2 \} = A .$$

أما خوارزمية إيجاد صفوف التكافؤ - $\sim$ على فضاء الحالات لحاسبة ما فتكون أكثر تعقيدا ، ولكن تستخدم طرقاً مشابهة للخوارزمية السابقة . وسنعرض فيما يلي بعض النتائج المساعدة :

12.3.2. تعريف.

نعرف لكل حاسبة منتهية $M = (S, I, \delta, s_0, F)$ متوالية العلاقات

$\sim^0, \sim^1, \sim^2, \sim^3, \ldots$ على المجموعة S بالشكل التالي :

$$1.\ s \sim^0 r \Leftrightarrow_{df} (s \in F \Leftrightarrow r \in F)\ ,$$

$$2.\ s \sim^1 r \Leftrightarrow_{df} s \sim^{j-1} r\ , (\forall i \in I)[\delta(s, i) \sim^{j-1} \delta(r, i)],\ \ j \geq 1\ .$$

13.3.2. مبرهنة .

لتكن $M = (S, I, \delta, s_0, F)$ حاسبة منتهية محددة ، عندئذ تتحقق التأكيدات التالية:

1) كل من العلاقات $\sim^j$ ، حيث $j \geq 0$ ، تشكل علاقة تكافؤ على S .

نرمز بـ $D_j =_{df} S / \sim^j$ لتحليل المجموعة S وفق العلاقة $\sim^j$.

2) لكل $j \geq 0$ ، يكون التحليل D_{j+1} تنعيم (تجزئة) للتحليل D_j .

3) لكل $j \geq 0$ ، تتحقق العلاقة $D_j = D_{j+1} \Rightarrow D_j = D_{j+t}$ ، بالنسبة لكل $t \geq 1$.

4) إذا كان $|S| = n$ ، عندئذ يوجد $k \leq n - 1$ ، حيث يكون :

$D_k = D_{k+1}$.

5) لكل k ، حيث $D_k = D_{k+1}$ يكون :

$$(\forall s, r \in S)[s \sim^k r \Leftrightarrow s \sim r].$$

أي أن $\sim^k = \sim$.

البرهان.

تنتج صحة التأكيدين 1). و 2). مباشرة من التعريف 12.3.2.

3) من التعريف 12.3.2 والفرض أن $D_j = D_{j+1}$ ، ينتج أن

$$s \sim^{j+1} r \Leftrightarrow s \sim^j r \;\&\; (\forall i \in I)[\delta(s,i) \sim^j \delta(r,i)] \Leftrightarrow$$

$$\Leftrightarrow s \sim^{j+1} r \;\&\; (\forall i \in I)[\delta(s,i) \sim^{j+1} \delta(r,i)] \Leftrightarrow s \sim^{j+2} r.$$

وبذلك برهنا ، أنه لكل $j \geq 0$ ، تتحقق العلاقة:

$$D_j = D_{j+1} \Rightarrow D_{j+1} = D_{j+2}.$$

وبالتدريج نبرهن أن $D_j = D_{j+1} \Rightarrow D_j = D_{j+t}$ ، بالنسبة لكل $t \geq 1$.

4) نرمز لعدد صفوف التحليل D_j بالرمز c_j . إذا كانت التحليلات

$D_0, D_1, \ldots, D_k$ مختلفة عن بعضها البعض ، عندئذ يكون

$1 \leq c_0 \leq c_1 \leq \ldots \leq c_k \leq n$. ينتج من ذلك أن $k \leq n+1$.

وبالتالي يوجد $k \leq n+1$ ، حيث يكون $c_k = c_{k+1}$ ، أي $D_k = D_{k+1}$.

5) من التعريف 12.3.2. يمكن بالتدرج استنتاج أنه ، من أجل $j \geq 0$ و $s, r \in S$

اختيارية ، يكون $s \sim^j r$ عندما يكون

$$(\forall w \in I^*)\,[\,|w| < j \Rightarrow (\delta(s,w) \in F \Leftrightarrow (\delta(r,w) \in F)\,].$$

يتضح من هذا ، أنه لكل $s , r \in S$ يكون

$s \sim r \Leftrightarrow (\forall j \geq 0)\,[\,s \sim^j r\,]$. إذا كان $D_k = D_{k+1}$ ،

عندئذ استنادا إلى 3) يكون

$$(\forall j \geq 0)\,[\,s \sim^j r\,] \Leftrightarrow (\forall j)\,[\,0 \leq j \leq k \Rightarrow s \sim^j r\,] .$$

ولكن بما أن D_k تجزئة لجميع التحليلات السابقة ، نجد أنه:

$$(\forall j \geq 0) \quad [\,s \sim^j r\,] \Leftrightarrow [\,s \sim^k r\,] .$$

ينتج مما سبق أن : $[\,s \sim r\,] \Leftrightarrow [\,s \sim^k r\,]$.*

من المبرهنة 13.3.2. نستخلص بسهولة خوارزمية لتحليل فضاء الحالات إلى صفوف تكافؤ - ~ . نعرضها فيما يلي :

14.3.2. خوارزمية.

لتكن $M = (S , I , \delta , s_0 , F)$ حاسبة منتهية محددة معرفة وفق المبرهنة 13.3.2.

خطوة 1) put $j := 0$; make D_0 ;

خطوة 2) repeat

```
   make D_{j+1}    ;     j := j + 1 ;
Until  D_j = D_{j-1}    ;
put  D := D_j  ;
```

15.3.2. تأكيد .

تشكل الخوارزمية 14.3.2. بالنسبة لحاسبة منتهية محددة ، تحليلاً لفضاء حالاتها وفق علاقة التكافؤ ~ .

البرهان : ينتج مباشرة من المبرهنة 13.3.2.

16.3.2. مثال .

أنشئ التحليل ~/S للحاسبة M المعطاة بجدول الدالة التالي :

	a	b
→ 1	2	3
2	2	4
← 3	3	5
← 4	2	7
5	6	3
← 6	6	6
7	7	4
8	2	3
9	9	4

نتناول بالتفصيل بنية التحليلات D_j .

التحليل D_0 : تتوزع المجموعة S إلى صفين ، أحدهما يتكون من الحالات النهائية ، والثاني يضم بقية الحالات .

$$D_0 : \{1,2,4,7,8,9\} , \{3,5,6\} .$$

نرمز لصفوف D_0 بالأرقام II , I ولكل صف من D_0 ننشئ جدولاً على الشكل التالي :

	a	b
1	I	II
2	I	I
4	I	I
7	I	I
8	I	II
9	I	I

	a	b
3	II	II
5	II	II
6	II	II

التحليل D_1 : يشكل تجزئة للتحليل D_0 . حسب التعريف 12.3.2. يضم صف التحليل الحالات التي لها رمز الدخل نفسه والحالات التي نعبرها من صفوف D_0 نفسها.

$$D_1 : I = \{1,8\} , II = \{2,4,7,9\} , III = \{3,5,6\}.$$

	a	b
1	II	III
8	II	III

	a	b
2	II	II
4	II	II
7	II	II
9	II	II

	a	b
3	III	III
5	III	III
6	III	III

$$D_2 : I = \{1,8\} , II = \{2,4,7,9\} , III = \{3,5,6\} .$$

نلاحظ أن $D_2 = D_1$ وبالتالي يكون : $S/_{\sim} = D_2$. نرمز لصفوف التكافؤ $S/_{\sim}$ كما يلي:

$$P = \{ 1, 8 \} \ , \ Q = \{ 2 , 4 , 7 , 9 \} , \ R = \{ 3 , 5 , 6 \} \ .$$

عندئذ يمكن تمثيل الحاسبة $M/\sim$ بجدول الدالة التالي :

	a	b
$\rightarrow$ P	Q	R
Q	Q	Q
$\leftarrow$ R	R	R

من السهل التحقق ، أن جميع حالات هذه الحاسبة تكون قابلة للبلوغ. وبالتالي $M/\sim$ تكون في هذه الحالة مختزل الحاسبة M .

حسب النظرية 8.3.2. كل مختزلين متكافئين لحاسبة يكونا متشاكلين تماما بالحالات . أي أنهما يختلفان فقط بتسمية الحالات . وهذا الإشكال بالتسمية يمكن استبعاده ، إذا حولنا الحاسبة إلى شكل قياسي .

يتم نقل الحاسبة المنتهية المحددة $M = (S , I , \delta , s_0 , F)$ إلى **الشكل القياسي** بإجراء ، يتم بنتيجته تشكيل جدول للحاسبة يشاكل تماماً M وفق الخطوات التالية :

خطوة 1: نأخذ $n = | S |$ ، ولتكون مجموعة حالات الحاسبة الجديدة $\{ 1 , 2 , \ldots , n \}$

خطوة 2: نشكل تقابل h بين المجموعة $\{ 1 , 2 , \ldots , n \}$ والمجموعة S .

خطوة 3: نرتب خطياً المجموعة $I = \{ i_1 , i_2 , \dots , i_n \}$ ، ونشكل الجدول على النحو التالي :

	i_1	i_2	$\dots$	i_m
1				
2				
...				
n				

نملأ خلايا الجدول وفق التقابل $h : \{ 1 , 2 , \dots , n \} \to S$ وفق التدرج التالي :

1. $h(1) =_{df} s_0$.
2. نملأ الجدول من الأعلى إلى الأسفل ومن اليسار إلى اليمين (حسب دالة النقل δ للحاسبة M) . بفرض أننا الآن ملأنا الجزء الأول من الجدول بالقيم $h(1) , h(2) , \dots , h(j)$ ، حيث $j < n$. نستمر بإملاء الجدول ، ما دمنا نجد حالة تقابل الحالة s يجب نقلها إلى الجدول وغير موجودة بين الحالات $h(1) , h(2) , \dots , h(j)$. عندئذ نضع $h(j + 1) = s$ ، وفي المكان المناسب نضع $h(j + 1)$.
3. أخيراً نشير في العمود الأيسر إلى الحالة الأولية (أي 1) بسهم دخل وإلى الحالات النهائية بأسهم خرج .

17.3.2. مثال .

أنقل الحاسبة المعطاة بالجدول (a) إلى الشكل القياسي .

	a	b
A	B	A
$\leftrightarrow$ B	E	C
C	A	E
$\rightarrow$ D	E	F
$\leftarrow$ E	A	B
F	A	E

جدول (a)

بتعريف دالة التقابل h على النحو التالي:

h(1) = B. , h(2) = E , h(3) = C , h(4) = A | h(5) = D , h(6) = F .

نحصل على الحاسبة بالشكل القياسي المبين في الجدول (b) .

	a	**b**
$\leftrightarrow$ **1**	**2**	**3**
$\leftarrow$ **2**	**4**	**1**
3	**4**	**2**
4	**1**	**4**
$\leftarrow$ **5**	**2**	**6**
6	**4**	**2**

جدول (b)

نلاحظ هنا حقيقة لم نشر إليها سابقاً وهي : بعد إملاء الأسطر الأربعة الأولى من الجدول ، لا يكون من الواضح حسب الوصف السابق ، كيفية الاستمرار . في هذه الحالة يمكن أن نضع $h(5)$ مساوياً لإحدى الحالات المتبقية ، وهي هنا F , D . وعدم التعيين هذا يمكن أن يظهر فقط في الحاسبات التي تحوي حالات غير قابلة للبلوغ . إذا أنهينا العمل في لحظة ظهور هذا الغموض لأول مرة ، نحصل على الشكل القياسي للحاسبة M ، الذي استبعد منه الحالات غير القابلة للبلوغ. بهذا نحصل على النتيجة المستخلصة نفسها بتطبيق الخوارزمية 14.3.2 على الحاسبة المعطاة .

18.3.2. **مثال** . (التحقق من التكافؤ الحاسوبي) .

تحقق فيما إذا كانت الحاسبتان M_1 , M_2 المعطيتان بالجدولين التاليين متكافئتين .

M_1 :

		a	b
$\leftrightarrow$	s_0	s_0	s_5
	s_1	s_1	s_3
	s_2	s_2	s_7
	s_3	s_3	s_2
$\leftarrow$	s_4	s_6	s_1
	s_5	s_5	s_1
$\leftarrow$	s_6	s_4	s_2
	s_7	s_7	s_0

M_2 :

		a	b
	A	G	H
	B	A	B
	C	D	E
	D	B	D
	E	D	C
	F	E	F
$\leftrightarrow$	G	F	G
	H	G	A

نعلم مما سبق ، أن M_1 , M_2 يكونان متكافئين فقط ، عندما يكون الشكلان القياسيان لمختزليهما متساويين . ننشئ $M_1/\sim$ وفق الخوارزمية 14.3.2.

D_0 : I = { s_0 , s_4 , s_6 } , II = { s_1 , s_2 , s_3 , s_5 , s_7 } .

	a	**b**
s_0	**I**	**II**
s_4	**I**	**II**
s_6	**I**	**II**

	a	**b**
s_1	**II**	**II**
s_2	**II**	**II**
s_3	**II**	**II**
s_5	**II**	**II**
s_7	**II**	**I**

D_1 : I = { s_0 , s_4 , s_6 } , II = { s_1 , s_2 , s_3 , s_5 } , III = { s_7 } .

بالطريقة نفسها نستمر في تحليل الجداول فنجد :

D_2 : I = { s_0 , s_4 , s_6 } , , II = { s_1 , s_3 , s_5 } , III = { s_2 } ,

IV = { s_7 } . بالتدريج نجد التحليلين التاليين :

D_3 : I = { s_0 , s_4 } , II = { s_6 } , III = { s_1 , s_5 } , IV = { s_3 } ,

V = { s_2 } , VI = { s_7 } .

D_4 : { s_0 } , { s_4 } , { s_6 } , { s_1 } , { s_5 } , { s_3 } , { s_2 } , { s_7 } .

نلاحظ أنه لا يوجد أي حالتين للحاسبة M_1 متكافئتين . بقي لدينا استبعاد الحالات غير القابلة للبلوغ ونقل الحاسبة الجديدة إلى الشكل القياسي. يمكن إجراء كلا الوظيفتين معاً ، كما بينا في المثال السابق ، وتكون النتيجة كما يلي :

		a	b
↔	1	1	2
	2	2	3
	3	3	4
	4	4	5
	5	5	6
	6	6	1

$h(1) = s_0$

$h(2) = s_5$

$h(3) = s_1$

$h(4) = s_3$

$h(5) = s_2$

$h(6) = s_7$

والآن ننشئ التحليل $M_2/\tilde{\sim}$:

$D_0 : I = \{A, B, C, D, E, F, H\}$, $II = \{G\}$.

		b	a
↔	A	II	I
	B	I	I
	C	I	I
	D	I	I
	E	I	I
	F	I	I
	H	II	I

وبالتدريج نجد التحاليل التالية :

$D_1 : \{ A , H \} , \{ B , C , D , E , F \} , \{ G \} .$

$D_2 : \{ A , H \} , \{ B \} , \{ C , D , E , F \} , \{ G \} .$

$D_3 : \{ A , H \} , \{ B \} , \{ C , E , F \} , \{ D \} , \{ G \} .$

$D_4 : \{ A , H \} , \{ B \} , \{ C , E \} , \{ F \} , \{ D \} , \{ G \} .$

$D_5 : \{ A , H \} , \{ B \} , \{ C , E \} , \{ F \} , \{ D \} , \{ G \} .$

نلاحظ أن $D_4 = D_5$ ، وهذا يعني أن D_5 يشكل تحليل فضاء الحالات للحاسبة M_2 وفق العلاقة ~ . وبذلك نحصل على الحاسبة ~/M_2 ، حيث نمثلها بجدول الدالة، حيث نمثل كل صف تكافؤ - ~ بأحد عناصره .

	a	b
A	A	B
B	B	A
C	C	D
F	F	C
D	D	B
↔ G	G	F

أثناء نقل الحاسبة إلى الشكل القياسي ، يجب إعادة ترتيب أبجدية الدخل في الجدول حيث تكون بنفس ترتيب أبجدية الدخل في الحاسبة السابقة . عندئذ نحصل على الحاسبة المعطاة بالجدول التالي :

	a	b	
$\leftrightarrow$ 1	1	2	h (1) = G
2	2	3	h (2) = F
3	3	4	h (3) = C
4	4	5	h (4) = D
5	5	6	h (5) = B
6	6	1	h (6) = A

من الملاحظ أننا ، حصلنا على الشكل القياسي نفسه لمختزلي الحاسبتين M_1 , M_2 ، وبالتالي تكون الحاسبتان متكافئتين .

19.3.2. تمارين.

1.3.2. أنشئ الأشكال القياسية لمختزلات الحاسبات المنشأة في التمرين 3.1.2.

2.3.2. إذا استبعدنا أولاً من حاسبة M معطاة الحالات غير القابلة للبلوغ ، ثم أنشأنا للحاسبة الناتجة N التحليل $N/\sim$. عندئذ تكون الحاسبة $N/\sim$ مختزل الحاسبة M . برهن ذلك .

3.3.2. إذا كان h تشاكلاً تبولوجياً من M إلى N و g تشاكلاً تبولوجياً من N إلى P ، عندئذ يكون g.h تشاكلاً تبولوجياً من M إلى P . برهن ذلك .

4.3.2. اكتب برنامجاً لخوارزمية الاختزال .

5.3.2. يمكن تشكيل صفوف التحليل D_j بالنسبة للخوارزمية 14.3.2 ، حيث نضع في صف – D_j نفسه الحالتين $s , r \in S$ عندما يكون :

$$(\forall w \in I^*)\,[\,|w| \leq j\,] \Rightarrow (\delta(s, w) \in F \Leftrightarrow \delta(r, w) \in F)\ .$$

خمن تقريبا ، كم تستغرق هذه الطريقة مقارنة مع الطريقة المبينة في المثال 16.3.2. ؟

6.3.2. حاول اقتراح خوارزمية ، تقرر فيما إذا كان $L(M) \subseteq L(N)$ ، حيث إنّ M , N حاسبتان اختياريتان منتهيتان . (يمكن الاستعانة بالفصل الثالث) .

4.2. الحاسبة المنتهية غير المحددة <NFA>

(non Deterministic Finite Automaton).

يمكن تعميم مفهوم الحاسبة المنتهية المحددة إلى مفهوم الحاسبة المنتهية غير المحددة . وعندها لا يكون ضرورياً تعيين حالة أولية وحيدة في الحاسبة ، ولكن يمكن تعيين مجموعة حالات أولية يمكن أن تبدأ منها الحاسبة. إن مفهوم الحاسبة المنتهية غير المحددة ، الذي سنعرضها فيما يلي ، يتمتع بأهمية خاصة ليس فقط من الناحية النظرية ، ولكن سنرى أن لهذه الحاسبة تطبيقات هامة أيضاً.

1.4.2. تعريف .

نعرف الحاسبة المنتهية غير المحددة <NFA> بأنها خماسية بالشكل :

$M = (S , I , \delta , B , F)$ ، حيث :

S مجموعة الحالات (غير فارغة) .

I مجموعة رموز الدخل (غير فارغة) .

$\delta : S \times I \rightarrow P(S)$ دالة النقل ، حيث $P(S)$ مجموعة جميع المجموعات الجزئية للمجموعة S.

$B \subseteq S$ مجموعة الحالات الابتدائية .

$F \subseteq S$ مجموعة الحالات النهائية.

2.4.2 تعريف.

نقول إن الكلمة $w = i_1 \, i_2 \, , \, \dots \, , \, i_n \in I^*$ مقبولة بالحاسبة المنتهية غير

المحددة $M = (S , I , \delta , B , F)$ ، إذا أمكن إيجاد متوالية حالات

$s_1 , s_2 , \ldots , s_{n+1}$ من S ، حيث يكون

$s_1 \in B$ ، $s_{n+1} \in F$ ، $s_{j+1} \in \delta (s_j , i_j)$ ، من أجل كل

$j \in \{ 1 , 2 , \ldots , n \}$.

تكون الكلمة الفارغة ε مقبولة بالحاسبة M ، فقط عندما يكون $B \cap F \neq \phi$.

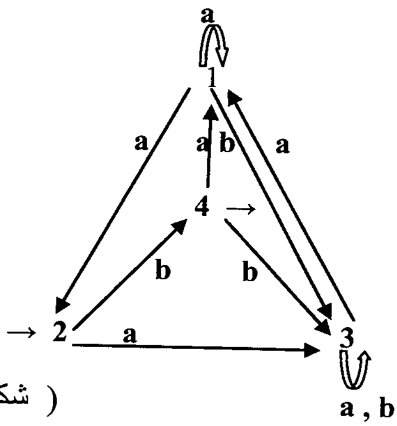
نرمز بـ $L (M)$ للغة الممثلة بحاسبة منتهية غير محددة M ، أي مجموعة الكلمات المقبولة بهذه الحاسبة . ونقول إن اللغة L مميزة بحاسبة منتهية غير محددة ، فقط عندما يوجد حاسبة منتهية غير محددة M ، حيث يكون $L (M) = L$.

نمثّل الحاسبة المنتهية غير المحددة عموماً بجدول الدالة أو بمخطط الحالات أو بشجرة ، بشكل مشابه للحاسبة المنتهية المحددة.

3.4.2 . مثال.

لتكن M حاسبة منتهية غير محددة معطاة بجدول الدالة أو بمخطط الحالات على النحو الآتي :

	a	b
1	1 , 2	3
→ 2	3	4
3	1 , 3	3
← 4	1	3



(شكل 4.2)

تقبل هذه الحاسبة مثلاً الكلمة abaa ، وتعبر الحالات 1, 1, 3, 1, 1 أو الحالات 1, 2, 4, 1, 1 أو الحالات 2, 3, 3, 1, 1 . بعض الحسابات على الكلمة abaa (مثلاً 1, 2, 4, 1, 2) لا تنتهي في حالة نهائية ، ولكن حسب التعريف 2.4.2 لا يغير ذلك شيئا. وبشكل مشابه نرى أن الكلمة b مقبولة ، لأنه يمكن الانتقال من الحالة 2 إلى الحالة 4 . ولكن يوجد حساب يبدأ في 1 ويقبل بالكلمة b .

كمثال على كلمة لا تنتمي إلى L (M) ، نقدم الكلمة bab أو bb ، بالنسبة للكلمة bb يوجد حسابان (2, 4, 3) , (1, 3, 3) ، ولكن أياً منها لا ينتهي في حالة نهائية.

يمكن أن ننظر إلى الحاسبة المنتهية المحددة كحالة خاصة من الحاسبة المنتهية غير المحددة. ينتج من ذلك ، أن كل لغة مميزة بحاسبة منتهية محددة تكون مميزة بحاسبة منتهية غير محددة. ولكن هل العكس صحيح ؟ أي هل كل لغة مميزة بحاسبة منتهية غير محددة تكون مميزة بحاسبة منتهية محددة ؟

للإجابة على هذا السؤال نقدم الخوارزمية والنظرية الآتيتين :

4.4.2.خوارزمية.

لتكن $N = (S , I , \delta , B , F)$ حاسبة منتهية غير محددة ، ولتكن $L (N)$ لغة مميزة بالحاسبة N . نشكل ممثل منته $M = (S', I , \delta' , s_0 , F')$ لتمييز $L (N)$ وفق الخطوات التالية :

خطوة 1 : نأخذ $P (S)$ مجموعة جميع المجموعات الجزئية لفضاء الحالات S .

خطوة 2 : نعرف الدالة $\delta' : P (S) \times I \rightarrow P (S)$ كما يلي:

لكل $i \in I$ و $k \in P (S)$ ، نأخذ $\delta' (k , i) = \cup_{s \in k} \delta(s , i)$.

خطوة 3 : نأخذ

$$S' = P (S) \ , \ s_0 = B \ , \ F' = \{ k \in P (S) \ ; k \cap F \neq \phi \} .$$

5.4.2. نظرية .

الخماسية $M = (S' , I , \delta' , s_0 , F')$ ، المعرفة في الخوارزمية السابقة ، تشكل حاسبة منتهية مميزة للغة $L (N)$.

البرهان:

من المعلوم أن

$$\varepsilon \in L (N) \Leftrightarrow B \cap F \neq \phi \Leftrightarrow B \in F' \Leftrightarrow \varepsilon \in L (M) .$$

ولنتحقق أنه من أجل كل كلمة w غير فارغة ، يكون

$$w \in L (M) \Leftrightarrow w \in L (N) .$$

1) نفرض أن $w = i_1 i_2 \dots i_n \in L(M)$, $n \geq 1$. عندئذ يوجد

$k_1 , \dots , k_{n+1} \in P(S)$ حيث $k_{n+1} \in F'$, $k_1 = B$ ، ولكل

$j \in \{1, 2, \dots, n\}$ يكون $k_{j+1} = \delta'(k_j, i_j)$.

نستنتج من ذلك أنه لكل r ، حيث $0 \leq r \leq n-1$ ، يوجد متوالية

s_{n+1} , $\dots$, s_{n-r+1} , s_{n-r} ، تحقق الشروط :

$s_j \in k_j$, $s_{n+1} \in \delta(s_j, i_j)$, $s_{n+1} \in F$, $s_{n+1} \in k_{n+1}$ ، من أجل

$n \geq j \geq n-r$.

نبرهن على ذلك بالتدريج وفق r .

I. نأخذ $r = 0$. ولنبرهن على وجود s_{n+1} , s_n حيث يكون

$s_{n+1} \in F$, $s_{n+1} \in \delta(s_n, i_n)$, $s_{n+1} \in k_{n+1}$, $S_n \in k_n$.

نعلم أن $k_{n+1} \in F'$. ينتج أن $k_{n+1} \cap F \neq \phi$. نختار

$s_{n+1} \in k_{n+1} \cap F$.

بما أن $s_{n+1} \in k_{n+1} = \delta'(k_n, i_n) = \cup_{s \in kn} \delta(s, i_n)$ ، فإنه يوجد

$s_n \in k_n$ ، حيث $s_{n+1} \in \delta(s_n, i_n)$.

II. نفرض أن التأكيد صحيح بالنسبة لـ r الذي يحقق الشرط $0 \leq r \leq n-1$. وهذا

يعني وجود متوالية $s_{n-r}, \dots, s_{n+1}$ بالخصائص المحددة .

ولنبرهن أن التأكيد محقق بالنسبة لـ $r+1$. لهذا يكفي البرهان على وجود

$s_{n-r-1} \in k_{n-r-1}$ ، حيث $s_{n-r} \in \delta(s_{n-r-1}, i_{n-r-1})$. وهذا ينتج من حقيقة أن

$s_{n-r} = \delta'(s_{n-r-1}, i_{n-r-1}) = \cup_{s \in kn-r-1} \delta(s, i_{n-r-1})$, $s_{n-r} \in k_{n-r}$.

وبذلك يتم برهان التأكيد لكل r ، حيث $(0 \le r \le n-1)$، خاصة من أجل $r = n$

1) إذاً يوجد متوالية $s_{n+1} , \ldots , s_1$ بالخصائص المطلوبة . ينتج مباشرة أن

$. i_1 \ldots i_n = w \in L (N)$ وبذلك يتم البرهان على أن

$. L (M) \subseteq L (N)$

2) لتكن $n \ge 1 , w = i_1 i_2 \ldots i_n \in L (M)$. إذاً يوجد متوالية حالات

$s_{n+1} , \ldots , s_1$ من S تحقق الشروط :

$$
\begin{array}{ll}
(a) & s_1 \in B \quad , \\
(b) & s_{j+1} \in \delta (s_j , i_j) , \quad 1 \le j \le n , \\
(c) & s_{n+1} \in F .
\end{array}
$$

نعرف المتوالية $k_{n+1} , \ldots , k_1$ بالشكل $k_{j+1} = \delta' (k_j , i_j)$ ، $k_1 = B$ من أجل $1 \le j \le n$. حسب الشرط (a) يكون $s_1 \in B$.

والشرط (b) يؤكد أن $s_t \in k_t$ ، من أجل كل $t \in (1 , \ldots , n+1)$.

وحسب (c) يكون $k_{n+1} \cap F \neq \phi$.

وهذا يعني أن $k_{n+1} \in F'$ و $i_1 i_2 \ldots i_n \in L (M)$. *

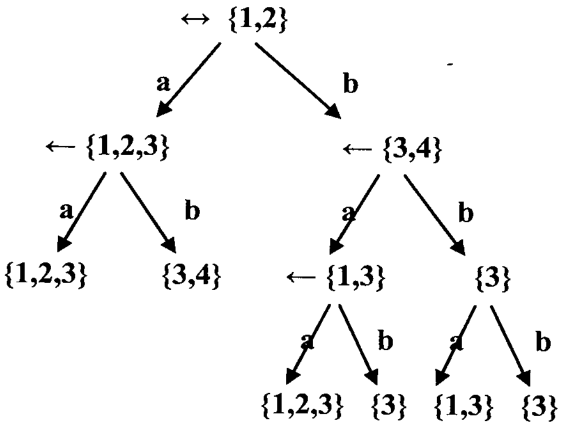
إن طريقة إنشاء الحاسبة M حسب الخوارزمية السابقة تقدم لنا خطة عملية لاستخلاص حاسبة منتهية من حاسبة منتهية غير محددة ومكافئة له . وللسهولة اخترنا

فضاء الحالات للحاسبة المحددة مساوياً لكامل (P (S . وكذلك أخذنا تلك الحالات من S' ، التي تكون قابلة للبلوغ من $B = s_0$.

6.4.2. **مثال.**

أنشئ حاسبة منتهية محددة مكافئة للحاسبة المنتهية غير المحددة المعطاة في المثال 3.4.2.

للسهولة نشكل شجرة الحالات للحاسبة كما يلي :



(شكل 5.2)

تحوي الحاسبة الناتجة 5 حالات ، أما (P (S فتحوي هنا 16 عنصراً . ومختزل الحاسبة ، الذي وجدناه ، يحوي 4 حالات .

بالحقيقة ، مسألة الانتقال من حاسبة منتهية غير محددة إلى حاسبة منتهية محددة مكافئة لها تملك أهمية تطبيقية كبيرة . وأحيانا يمكن بطريقة أساسية تسهيل مسألة إنشاء حاسبة منتهية ممثل للغة معطاة . لهذا الهدف نعرض المثال التالي:

7.4.2. مثال .

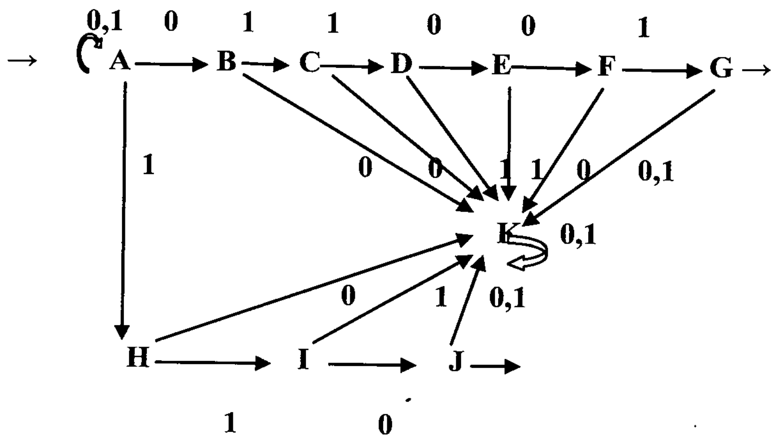
المطلوب اقتراح حاسبة منتهية محددة مميزة للغة بأبجدية { 0 , 1 } ومشكلة من جميع الكلمات المنتهية بالسلسلة 011001 أو بالسلسلة 110 .

لحل هذه المسألة يمكن التدرج بطريقتين :

a) إنشاء حاسبة محددة مباشرة. يستطيع القارئ بنفسه التحقق، حتى في هذه الحالة البسيطة، فإنه يحتاج إلى جهد كبير .

b) اقتراح حاسبة منتهية غير محددة تميز اللغة المعطاة ، بعدئذ وبمساعدة الإجراء المبين في الخوارزمية 4.4.2. يمكن استبدال الحاسبة المميزة غير المحددة بحاسبة مميزة محددة .

كمثال على ذلك الحاسبة نقدم مخطط الحالات التالي للحاسبة المطلوبة :



(شكل 6.2)

الفرعان ABCDEFG و AHIJ يقابلان الكلمات المنتهية بالسلسلة 011001 والسلسلة 110 .

إن عملية الانتقال من حاسبة منتهية غير محددة إلى حاسبة منتهية محددة مختزلة ، تعد وظيفة ميكانيكية ، يمكن استخدام الحاسوب لحلها .

8.4.2. تطبيق.

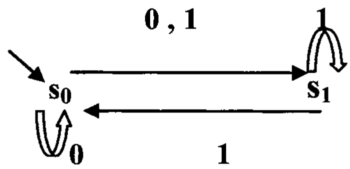
أوجد الحاسبة المنتهية المحددة المكافئة للحاسبة المنتهية غير المحددة التالية :

$$M = (\{ s_0 , s_1 \}, \{ 0 . 1 \}, \delta , s_0 , \{ s_1 \})$$

حيث تعطى الدالة δ بالجدول التالي:

δ	0	1
s_0	$\{s_0, s_1\}$	$\{s_1\}$
s_1	$\varnothing$	$\{s_0, s_1\}$

ويكون لهذه الحاسبة المخطط التالي



(شكل 7.2)

نشكل الحاسبة المنتهية المحددة المكافئة للحاسبة M بالشكل

$$M' = (S' , \{0 , 1\} , \delta , [s_0] , \{[s_1] , [s_0 , s_1]\})$$

$$S' = \{ \varnothing , [s_0] , [s_1] , [s_0 , s_1]\}$$

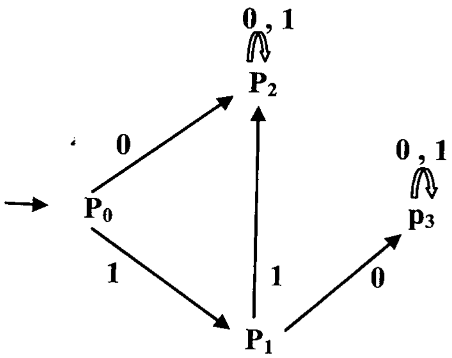
حيث تعرف دالة النقل كما في الجدول التالي:

δ'	0	1
$[s_0]$	$[s_0,s_1]$	$[s_1]$
$[s_1]$	$\varnothing$	$[s_0,s_1]$
$[s_0,s_1]$	$[s_0,s_1]$	$[s_0,s_1]$
$\varnothing$	$\varnothing$	$\varnothing$

وبإعادة التسمية للحالات نحصل على الجدول التالي الذي يعرف الدالة δ' :

δ'	0	1
$\rightarrow P_0$	P_2	P_1
$\leftarrow P_1$	P_3	P_2
$\leftarrow P_2$	P_2	P_2
P_3	P_3	P_3

الحالة الابتدائية هي p_0 والحالات النهائية هي p_1 , p_2 لاحتوائها على الحالة النهائية الوحيدة للحاسبة M وهي s_1 . يمكن تمثيل الحاسبة المنتهية المحددة الناتجة بالمخطط التالي :



(شكل 8.2)

9.4.2. تمارين.

1.4.2. أنشئ حاسبة منتهية غير محددة ، تقبل تماما جميع الكلمات من الأبجدية { a , b , c } ، التي تبدأ أو تنتهي بالسلسلة baac أو التي تحوي في أي مكان منها السلسلة abca .

2.4.2. اقترح خوارزمية ، تمكننا بالنسبة لكل حاسبة غير محددة M التأكد فيما إذا $L(M) \neq \phi$.

3.4.2. اقترح خوارزمية ، تمكننا بالنسبة لكل حاسبة غير محددة M على الأبجدية I التحقق فيما إذا $L(M) \subset I^*$.

4.4.2. فكر بأيةِ طريقة تستطيع ، بالنسبة لحاسبة منتهية غير محددة M ، برمجة إجراء يتقبل فقط كلمات من $L(M)$.

3. الفصل الثالث
الحاسبات التعاقبية
(Sequential Machines)

محتوى الفصل .

1.3. حاسبتا (آلتا) ميلي و مور التعاقبيتان

(Mealy and Moor Sequential Machines)

2.3. الدالة التعاقبية (Sequential function).

3.3. تحليل الحاسبات (Analysis machines).

الهدف من الفصل .

اكتساب المعارف التالية :

مفهوم حاسبتي (آلتي) ميلي و مور التعاقبيتين ، التكافؤ المسلكي للحاسبات ، الدالة التعاقبية ، تحليل الحاسبات ، وصل الحاسبات على التسلسل ، وصل الحاسبات على التفرع .

1.3. حاسبتا (آلتا) ميلي و مور التعاقبيتان

(Mealy and Moor Sequential Machines)

إن نموذج الحاسبة المنتهية ، الذي درسناه سابقاً ، لم يأخذ بالحسبان الطريقة التي تؤثر بها الحاسبة في الوسط المجاور . فإذا اعتبرنا التجهيزات الموصوفة حاسبة منتهية ، وأضفنا في كل لحظة إحدى إشارات الخرج الكثيرة ، فنحصل على حاسبة (آلة) تعاقبية.

1.1.3. تعريف.

نعرف حاسبة ميلي المنتهية التعاقبية بأنها خماسية

$M = (S , I , F , \delta , \lambda)$ ، حيث

S : مجموعة منتهية غير خالية (مجموعة الحالات ، فضاء الحالات).

I : مجموعة منتهية غير خالية (مجموعة رموز الدخل ، أبجدية الدخل).

F : مجموعة منتهية غير خالية (مجموعة رموز الخرج ، أبجدية الخرج).

δ : $\delta : S \times I \to S$ (دالة النقل).

λ : $\lambda : S \times I \to F$ (دالة الخرج).

خلافاً للحاسبة المنتهية المعرفة سابقا، لن نورد بالنسبة للحاسبة التعاقبية حالة ابتدائية أو مجموعة حالات نهائية . وأحيانا لا يميز في المراجع بين مصطلحات الآلة و الحاسبة ، وتسمى آلة ميلي التعاقبية حاسبة أيضاً . بالنسبة لآلة ميلي ، يرتبط رمز الخرج بالحالة الداخلية للآلة وبرمز الدخل .

وسنعرض نموذجاً آخر للآلة التعاقبية ، يرتبط فيها رمز الخرج فقط بحالة داخلية .

2.1.3. تعريف.

تسمى الآلة التعاقبية آلة مور ، إذا وجدت دالة $\mu : S \rightarrow F$ ، حيث

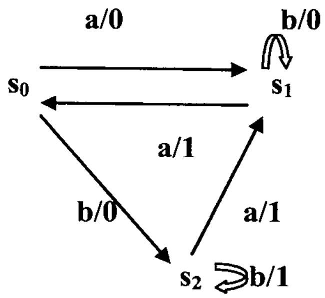
$$\forall (s,i) \in S \times I : \lambda(s,i) = \mu(\delta(s,i)).$$

يمكن تمثيل الآلة التعاقبية بشكل مشابه للحاسبة المنتهية ، بجدول دالة أو بمخطط حالات .

3.1.3. مثال.

a) نقدم مثالاً لآلة ميلي ، بأبجدية دخل $I = \{a, b\}$ وأبجدية خرج $F = \{0, 1\}$ ومجموعة حالات $S = \{s_0, s_1, s_2\}$ ، بجدول الدالة وبمخطط الحالات التاليين :

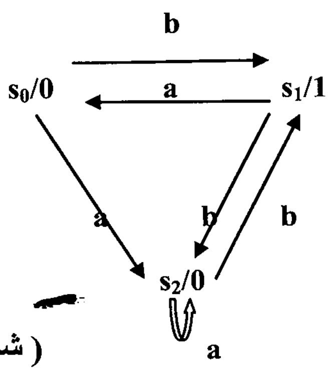
	a	b
s_0	$s_1 ; 0$	$s_2 ; 0$
s_1	$s_0 ; 1$	$s_1 ; 0$
s_2	$s_1 ; 1$	$s_2 ; 1$



(شكل 1.3)

b) كمثال لآلة مور ، بأبجدية دخل $\{ a , b \}$ وأبجدية خرج $\{ 0 , 1 \}$ ومجموعة حالات $S = \{ s_0 , s_1 , s_2 \}$ ، نقدم جدول الدالة ومخطط الحالات التاليين :

	a	b	μ
s_0	s_2	s_1	0
s_1	s_0	s_2	1
s_2	s_2	s_1	0



(شكل 2.3)

يمكن بالنسبة لآلة مور توسيع دالة النقل $\delta : S \times I \rightarrow S$ ودالة الخرج $\lambda : S \times I \rightarrow F$ ، حيث $(\lambda = \mu . \delta)$ ، بطريقة مناسبة إلى الدالتين :

$$\lambda\ S \times I^* \rightarrow F \quad , \quad \delta : S \times I^* \rightarrow S$$

كما نبين فيما يلي :

4.1.3. تعريف.

لتكن $M = (S , I , F , \delta , \mu)$ آلة مور .

نعمم دالتي النقل والخرج $\delta : S \times I^* \rightarrow S \quad , \quad \lambda : S \times I^* \rightarrow F$ على النحو الآتي :

من أجل كل $u \in I^*$, $i \in I$, $s \in S$ ، يكون :

$$\delta (s , \varepsilon) = s \quad ,$$

$$\delta (s , ui) = \delta (\delta (s , u) , i) ,$$

$$\lambda (s , u) = \mu (\delta (s , u)) .$$

من السهل التحقق من صحة المبرهنةالآتية :

5.1.3. مبرهنة .

إذا كانت $M = (S , I , F , \delta , \lambda)$ آلة مور و δ (λ ، على التوالي) التعميم الناسب لدالة النقل (الخرج ، على التوالي) ، عندئذ من أجل كل $s \in S$, $u , v \in I^*$ يكون :

$$\delta (s , uv) = \delta (\delta (s , u) , v) ,$$

$$\lambda (s , uv) = \lambda (\delta (s , u) , v) .$$

بالنسبة لآلة ميلي عموماً، لا يمكن تعريف قيمة $\lambda (s , \varepsilon)$ ، حيث تتحقق المبرهنة 5.1.3.

6.1.5. تعريف .

لتكن $M = (S , I , F , \delta , \lambda)$ آلة ميلي . عندئذ تعميم دالة النقل $\delta : S \times I^* \rightarrow S$ يعرف بشكل مطابق لما ذكرناه في ال تعريف 4.1.3. أما تعميم دالة الخرج $\lambda : S \times I^+ \rightarrow F$ فيعرف على النحو الآتي :

$$\forall i \in I , u \in I^* , s \in S \; : \; \lambda (s , ui) = \lambda (\delta (s , u) , i)$$

سنحدد فيما يلي الشرح فقط على آلة مور . ويمكن إجراء هذا التحديد دون التقليل من عمومية المسألة . وسنبرهن أنه لكل آلة ميلي M يوجد آلة مور مكافئة مسلكياً لها ، أي آلة تتفاعل مع مؤثرات الدخل عموما بطريقة M نفسها . وسنعرض أولاً الصيغة الدقيقة لمفهوم التكافؤ المسلكي للآلة.

7.1.3. تعريف.

لتكن $M = (S , I , F , \delta , \lambda)$ ، $M = (S_1 , I , F_1 , \delta_1 , \lambda_1)$ آلتي ميلي بأبجدية الدخل نفسها. ولتكن $s \in S , s_1 \in S_1$.

نقول إن الحالتين s , s_1 متكافئتان مسلكياً (ونكتب $s \sim_m s_1$) إذا كان

$$\lambda (s , u) = \lambda_1 (s_1 , u) \quad , \quad \forall u \in I^+ \quad .$$

8.1.3. تعريف.

1) تكون الآلة M_1 محتواة مسلكيا في الآلة M (ونكتب $M_1 \subset_{\sim m} M$) ، إذا كان من أجل كل $s_1 \in S_1$, يوجد $s \in S$ حيث $s_1 \sim_m s$.

2) تكون الآلتان M_1 , M متكافئتين مسلكياً (ونكتب $M \sim_m M_1$) ، إذا كان

$$M \subset_{\sim m} M_1 \quad \& \quad M_1 \subset_{\sim m} M .$$

9.1.3. نظرية.

يوجد لكل آلة ميلي $M = (S , I , F , \delta , \lambda)$ ، آلة مور $M' = (S' , I , F , \delta' , \mu)$ مكافئة مسلكياً لها.

البرهان . (خوارزمية إنشاء آلة مور M' المقابلة لآلة ميلي M)

يتم الانتقال من آلة ميلي M إلى آلة مور M' ، بتجزئة كل حالة للآلة الأصلية إلى عدد من الحالات ، حيث كل حالة جديدة ناتجة يمكن دمجها مع أحد رموز الخرج . سيكون للآلة M' أبجدية دخل I وأبجدية خرج F كما في الآلة M . بما أن $F \neq \phi$ ، فإنه يمكن اختيار $y_0 \in F$.

نعرف مجموعة الحالات S' ودالة النقل δ' ودالة الخرج μ' كما يلي :

$$S' =_{df} S'_1 \cup S'_2 \;;$$

$$S'_1 =_{df} \{ (s , y) \in S \times F ; (\exists (s^- , i) \in S \times I) [\delta (s^- , i) = s \;\&\; \lambda (s^- , i) = y] \} ;$$

$$S'_2 =_{df} \{ (s , y_0) ; s \in S \;\&\; \neg (\exists (s^- , i) \in S \times I) [\delta (s^- , i) = s] \} ;$$

$$\delta' ((s , y) , i) =_{df} (\delta (s , i) , \lambda (s , i)) ;$$

$$\lambda' ((s , y) , i) =_{df} \lambda (s , i) .$$

- تعريف S' صحيح ، لأن $\delta' ((s , y) , i) \in S'$ و $\lambda' ((s , y) , i) \in F$ بالنسبة لجميع $(s , y) \in S'$, $i \in I$.

- M' تكوّن أيضا آلة مور ، لأننا إذا عرفنا الدالة $\mu : S' \rightarrow F$ بالشكل (

$\mu (s , y)) = y$ ، عندئذ $\lambda' = \mu \circ \delta'$.

- بقي أن نتحقق أن $M' \sim_m M$. يصبح هذا واضحاً بعد برهان صحة العلاقة :

$$(\forall s \in S) (\forall y \in F) [(s , y) \in S' \Rightarrow s \sim_m (s , y)] \quad (*)$$

نبرهن أولاً أن :

$$(\forall u \in I^+) (\forall (s , y) \in S') [\delta' ((s , y) , u) =$$

$$= (\delta(s , u) , \lambda(s , u))] \quad (**)$$

1) بالنسبة لكلمة u بطول t تنتج هذه المساواة مباشرة من تعريف δ' .

2) إذا كانت $(**)$ محققة بالنسبة لكلمة $v \in I^+$ ، فإنها تتحقق من أجل الكلمة u

$= vi$ ، حيث $i \in I$ ، لأن

$$\delta' ((s , y) ,vi) = (\delta' (\delta'(s , y) , v) , i) =$$

$$\delta' ((\delta (s , v) , \lambda (s , v)) , i) =$$

$$= (\delta (\delta (s , v) , i) , \lambda (\delta (s , v) , i)) = (\delta (s , vi) , \lambda (s , vi)) .$$

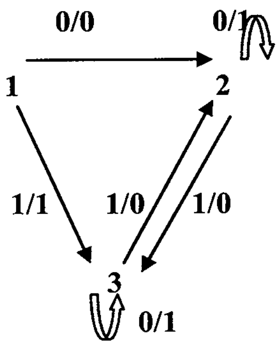
3) بطريقة برهان $(**)$ نفسها، نبرهن على تحقق العلاقة :

$(\forall u \in I^+) (\forall (s , y) \in S') [\lambda ((s , u) = \lambda'(s , y) , u)]$. وبذلك

يتم برهان $(*)$.

10.1.3. مثال.

ننشئ آلة مور M' المكافئة مسلكيا آلة ميلي M المعطاة بمخطط على النحو الآتي

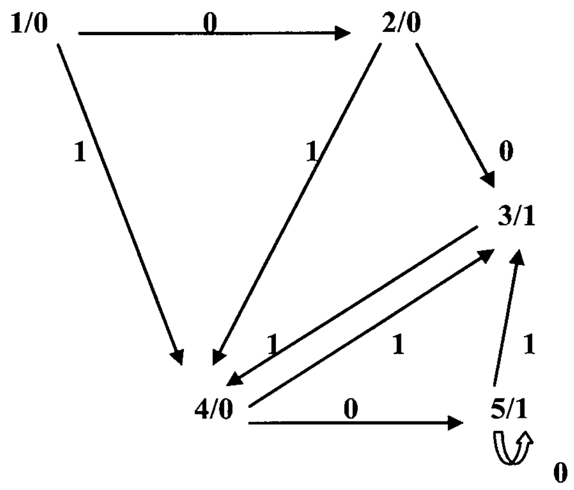


(شكل 3.3)

		0	1	
M' :	(1 , 0)	(2 , 0)	(3 , 0)	0
	(2 , 0)	(2 , 1)	(3 , 0)	0
	(2 , 1)	(2 , 1)	(3 , 0)	1
	(3 , 0)	(3 , 1)	(2 , 1)	0
	(3 , 1)	(3 , 1)	(2 , 1)	1

إذا رمزنا ، من أجل الوضوح ، للحالات (1,0), (2,0), (2,1), (3,0), (3,1) على الترتيب بالرموز

5 , 4 , 3 , 2 , 1 ، يمكننا تمثيل الآلة M' بالمخطط التالي:



(شكل 4.3)

تمكننا النظرية 9.1.3. من دراسة آلة مور دون التأكيد على عمومية المسألة ، لأننا عرفنا بالنسبة لآلة مور جواب الآلة على متوالية بطول صفر . وقد استخدمنا مفهوم تكافؤ الحالات وتكافؤ الآلات مدعمة بشروط ، لكي لا يسمح تعريف الحالات المتكافئة بالتجزئة برد الفعل على كلمة دخل فارغة.

11.1.3. تعريف.

لتكن $M = (S , I , F , \delta , \lambda)$ ، $M' = (S' , I , F' , \delta' , \lambda')$ آلتين بأبجدية الدخل I نفسها نقول :

1) يكونا الحالتين $s \in S$, $s' \in S'$ متكافئتين موريا (ونكتب $s \sim_m s'$)

إذا كان $\mu(s) = \mu' (s')$ & $s \sim_m s'$.

2) يمكن احتواء M في M' (ونكتب $M \subset_{\sim m} M'$) ، إذا كان

$(\forall s \in S)(\exists s' \in S') : (s \sim_m s')$.

3) M تكافئ M' مسلكيا (ونكتب $M \sim_m M'$) ، إذا كان

$M \subset_{\sim m} M'$ & $M' \subset_{\sim m} M$.

12.1.3. ملاحظة .

نلاحظ أنه عندما يكون $M = M'$ تكون العلاقة $\sim_m$ ، الواردة في البند (1) من التعريف السابق ، علاقة على المجموعة S . وفي هذه الحالة نتحدث عن تكافؤ الحالات للحاسبة نفسها.

13.1.3. تأكيد.

1) لتكن $M = (S, I, F, \delta, \mu)$ حاسبة، عندئذ تكون العلاقة $\sim_m$ علاقة تكافؤ على المجموعة S.

2) تكون العلاقة $\sim_m$ على مجموعة الحاسبات بأبجدية الدخل نفسها علاقة تكافؤ.

14.1.3. تعريف .

1) تكون الحاسبة $M = (S, I, F, \delta, \mu)$ حاسبة مختزلة ، إذا تحققت من أجل كل $s_1, s_2 \in S$ العلاقة $s_1 \sim_m s_2$ ، عندئذ يكون $s_1 = s_2$.

2) لتكن M حاسبة ما، عندئذ كل حاسبة M' ، حيث :

a) M' حاسبة مختزلة .

b) $M' \sim_m M$ ،

عندئذ تدعى M' حاسبة مختزلة للحاسبة M .

15.1.3. **نظرية.**

يوجد خوارزمية، تنشئ لكل حاسبة مور حاسبة مختزلة.

نترك برهان النظرية للقارئ كتمرين . لأن خوارزمية اختزال الحاسبات التعاقبية تكون مشابهة لخوارزميات اختزال الحاسبات التي تعرفنا عليها. وسنورد هنا فقط مثالاً بسيطاً لتوضيح ذلك.

16.1.3. **مثال.**

أنشئ الحاسبة المختزلة لحاسبة مور المعطاة بجدول الدالة التالي :

	0	1	
1	2	2	0
2	3	2	1
3	4	4	1
4	5	2	0
5	6	2	1
6	4	4	1

لنشكل متتالية التحليلات $D_0 , D_1 , D_2 , \ldots\ldots$

$$D_0 = (1 , 4)\ (2 , 3 , 5 , 6)$$

$$D_1 = (1 , 4)\ (2 , 5)\ (3 , 6)$$

$$D_2 = (1\ 4)\ (2 , 5)\ (3 , 6)$$

من الواضح أن $D_1 = D_2$ ، وبالتالي $D_1 = S/\sim_m$ ، وتكون الحاسبة المختزلة المطلوبة

	0	1	
1	2	2	0
2	3	2	1
3	1	1	1

بقية المبرهنات والتأكيدات في هذه الفقرة سنعرض ها دون برهان . ويستطيع القارئ كتابتها إذا فهم البراهين في الجزء الأول A ، وهي في الحقيقة مشابهة لها.

17.1.3. تعريف.

لتكن $M = (S, I, F, \delta, \mu)$ ، $M' = (S', I', F', \delta', \mu')$ حاسبتين . تدعى الثلاثية $h = (h_S, h_I, h_F)$ تشاكلاً تبولوجياً للحاسبة M إلى الحاسبة M' ، إذا تحققت الشروط التالية:

1) $h_S : S \to S'$, $h_I : I \to I'$, $h_F : F \to F'$,

2) $\forall s \in S$, $i \in I$: $h_S(\delta(s,i)) = \delta'(h_S(s), h_I(i))$,

3) $\forall s \in S$: $h_F(\mu(s) = \mu'(h_S(s)))$.

18.1.3. مبرهنة .

1) ليكن $h = (h_S , h_I , h_F)$ تشاكلاً تبولوجياً للحاسبة $M = (S , I , F , \delta , \mu)$

إلى الحاسبة $M' = (S' , I' , F' , \delta' , \mu')$ ،

عندئذ من أجل كل $s \in S , i_1, \ldots , i_n \in I^*$ يكون :

$$\text{a)} \quad h_S \,(\delta (s ,i_1 \ldots i_n)) = \delta' (h_S (s) , h_I (i_1) \ \ldots h_I (i_n))$$

$$\text{b)} \quad h_F \,(\lambda (s ,i_1 \ldots i_n)) = \lambda' (h_F (s) , h_I (i_1) \ \ldots h_I (i_n))$$

حيث : $\lambda' = \mu' \, o \, \delta'$, $\lambda = \mu \, o \, \delta$.

2) ليكن $h_1 = (h_S , h_I , h_F)$ تشاكلاً تبولوجياً للحاسبة M إلى الحاسبة M' و

$h_2 = (h_S' , h_I' , h_F')$ تشاكلاً تبولوجياً للحاسبة M' إلى الحاسبة M'' ، عندئذ

يكون

$h_2 \, o \, h_1 = (h_S' \, o \, h_S , h_I' \, o \, h_I , h_F' \, o \, h_F)$ تشاكلاً تبولوجياً للحاسبة M إلى

الحاسبة M'' .

19.1.3. تعريف.

1) يدعى التشاكل $h = (h_S , h_I , h_F)$ المعروض في التعريف 17.1.3. تشاكل

تبولوجي على الحاسبة M' ، إذا كانت الدوال h_S , h_I , h_F غامرة .

2) يسمى التشاكل $h = (h_S , h_I , h_F)$ تشاكلاً تاماً ، إذا كانت الدوال h_S , h_I , h_F متباينة .

20.1.3. مبرهنة.

إذا كان $h = (h_S , h_I , h_F)$ تشاكلاً تاماً للحاسبة M على الحاسبة M' ، عندئذ يكون $h^{-1} = (h^{-1}_S , h^{-1}_I , h^{-1}_F)$ تشاكلاً تاماً للحاسبة M' على الحاسبة M .

21.1.3. تعريف .

تدعى الثلاثية $(\equiv_S , \equiv_I , \equiv_F)$ تطابق على الحاسبة $M = (S , I , F , \delta , \mu)$ ، إذا تحققت الشروط التالية:

1) $\equiv_S$, $\equiv_I$, $\equiv_F$ تشكل ، على التوالي، علاقات تكافؤ على S , I , F .

2) $\forall s , s' \in S , \forall i , i' \in I : [s \equiv_S s' \ \& \ i \equiv_I i' \Rightarrow$

$\Rightarrow \delta (s , i) \equiv_S \delta (s' , i') \ \& \ \mu (s) \equiv_F \mu (s')]$

22.1.3. تعريف .

نعرف التحليل الآلي للحاسبة $M = (S , I , F , \delta , \mu)$ وفق التطابق $\equiv = (\equiv_S , \equiv_I , \equiv_F)$ بأنه الحاسبة

$M/_{\equiv} = (S/\equiv_S \ , \ I/\equiv_I \ , F/\equiv_F \ , \ \delta_{\equiv} , \mu_{\equiv} \ )$ ، حيث $\mu_{\equiv}$, $\delta_{\equiv}$ تعرف كما يلي :

$$\delta_{\equiv} ([s]_S , [i]) = [\delta (s , i)]_S \ ,$$

$$\mu_{\equiv} ([s]_S) = [\mu (s)]_F \ .$$

إن مفهومي التشاكل التبولوجي بين الحاسبات والتطابق على الحاسبات مرتبطان ببعضيهما ، كما يتضح من النظريتين التاليتين :

23.1.3. نظرية.

ليكن $M/_{\equiv}$ تحليل الحاسبة وفق التطابق $\equiv$ ،عندئذ تعطي ثلاثية الدوال

$$h_S : S \rightarrow S/\equiv_S \ ; \quad h_I : I \rightarrow I/\equiv_I \ ; \quad h_F : F \rightarrow F/\equiv_F$$

المعرفة بالعلاقات $h_S (s) = \ [s]_S \ ; \quad h_I (i) = \ [i]_I \ ; \quad h_F (F) = \ [F]_F$

تشاكلاً تبولوجياً $h_{\equiv}$ للحاسبة M على التحليل $M/_{\equiv}$.

24.1.3. نظرية.

ليكن $h = (h_S , h_I , h_F)$ تشاكلاً تبولوجياً للحاسبة $M = (S , I , F , \delta , \mu)$ على الحاسبة $M' = (S' , I' , F' , \delta' , \mu')$.

نعرف على المجموعات $S \ , \ I \ , \ F$ ،على التوالي، علاقات التكافؤ $\equiv_S \ , \ \equiv_I \ , \equiv_F$ حيث :

$$s_1 \equiv_S s_2 \ \Leftrightarrow_{df} \ h_S (s_1) = h_S (s_2) \ ,$$

$$i_1 \equiv_I i_2 \ \Leftrightarrow_{df} \ h_I (i_1) = h_I (i_2) \ ,$$

$$F_1 \equiv_F F_2 \quad \Leftrightarrow_{df} \; h_F(F_1) = h_F(F_2).$$

عندئذ تشكل الدالة $\equiv = (\equiv_S , \equiv_I , \equiv_F)$ تطابق على الحاسبة M ، وكذلك $M/_{\equiv}$ تكون متشاكلة تماما مع الحاسبة M' .

25.1.3. مبرهنة.

ليكن $\equiv = (\equiv_S , \equiv_I , \equiv_F)$ تكافؤ على الحاسبة $M = (S , I , F , \delta , \mu)$ ،

وليكن $s , s' \in S$ ، $u = i_1 , \dots , i_n \in I'$ ، $u' = i_1' , \dots , i_n' \in I^*$.

إذا كان $s \equiv_S s'$ & $(\forall j)[1 \le j \le n] \Rightarrow i_j \equiv_I i_j'$ ، عندئذ يكون

$$\delta(s , u) \equiv_S \delta(s' , u')$$

$$\mu(s , u) \equiv_F \mu(s' , u') .$$

إن التشاكل التبولوجي للحالات (التطابق) ، الذي سنعرضه فيما يلي ، يقدم أمثلة خاصة هامة للتشاكل التبولوجي بين الحاسبات (التطابق على الحاسبات) . وهذا الارتباط سيتضح في نهاية هذه الفقرة .

25.1.3. تعريف.

1. يدعى التشاكل التبولوجي $h = (h_S , h_I , h_F)$ للحاسبة M إلى الحاسبة M' تشاكلاً تبولوجياً للحالات ، إذا كان $h_I = id_I$, $h_F = id_F$ (حيث ، يرمز id_A للتطبيق المطابق على المجموعة A).

2. يدعى التشاكل التام $h = (h_S , h_I , h_F)$ تشاكلاً تاماً للحالات ، إذا كان $h_I = id_I$, $h_F = id_F$.

27.1.3. مبرهنة.

التشاكل التام العكسي لتشاكل تام للحالات يكون أيضا تشاكلاً تاماً للحالات.

28.1.3. نظرية.

ليكن $h = (h_S , h_I , id_F)$ تشاكلاً تبولوجياً لحالات الحاسبة

$M = (S , I , F , \delta , \mu)$ إلى الحاسبة $M' = (S' , I , F' , \delta' , \mu')$

(الحاسبتان بأبجدية الدخل نفسها) ، عندئذ يكون

$\forall s \in S , \ s \sim_m h_S (s)$ ، أي أن $M \subset_{\sim m} M'$.

29.1.3. نتيجة.

1. إذا كانت M , M' حاسبتين متشاكلتين تماما بالحالات ، عندئذ $M \sim_m M'$.
2. إذا كان $h = (h_S , id_I , id_F)$ تشاكلاً تبولوجياً لحالات مختزل الحاسبة $M = (S , I , F , \delta , \mu)$ إلى نفسه، عندئذ يكون h تشاكلاً تاماً وحتى تشاكل تام مطابق ، أي $h = (id_S , id_I , id_F)$.

30.1.3. تعريف .

تسمى العلاقة $\sim$ تطابق حالات على الحاسبة $M = (S , I , F , \delta , \mu)$ ، إذا كانت الثلاثية $(\sim , = , =)$ تشكل تطابقاً على M . أي أن :

1. $\sim$ علاقة تكافؤ على S .
2. $s \sim s' \Rightarrow (\forall i \in I) [\delta (s' , i) \sim \delta (s , i) \ \& \ \mu (s) \sim \mu (s')$.

نلاحظ أن تكافؤ الحالات $\sim_m$ يشكل تطابق حالات . إضافة إلى ذلك ، يشكل تحليل S وفق تطابق حالات اختياري تنعيم لتحليل S وفق $\sim_m$.

31.1.3. نظرية .

لتكن $M = (S , I , F , \delta , \mu)$ حاسبة و $\sim$ تطابق حالات ما على M ، عندئذ يكون $\forall s , s' \in S, \ s \sim s' \Rightarrow s \sim_m s'$.

32.1.3. نظرية.

تكون الحاسبة M حاسبة مختزلة إذا وفقط إذا أمكن تصوير كل حاسبة M' متكافئة معها تبولوجيا على M.

33.1.3. نظرية .

لتكن M , M' حاسبتين مختزلتين و $M \sim_m M'$ ، عندئذ تمثل M تشاكل حالات تام مع M' ، وبالوقت نفسه يكون هذا التشاكل وحيد التعيين .

بشكل مشابه للحاسبات المنتهية ، يمكن أن يكون للحاسبات المتعاقبة مختزلان للحاسبة نفسها مختلفان فقط بتسمية الحالات. كذلك بشكل مشابه للحاسبات المنتهية ، نستطيع إيجاد الشكل القياسي للحاسبة وبالتالي إيجاد الشكل القانوني للمختزل.

34.1.3. تمارين.

1.1.3. اكتب متوالية الخرج للحاسبة المعطاة في المثال 3.1.3. a جوابا لمتوالية الدخل :

a) abbabba

b) bbbbaab

إذا بدأت العمل في الحالة s_0 .

2.1.3. أنشئ لكل من الأمثلة التالية حاسبة ميلي بالصفات المذكورة ، مع العلم أن أبجدية الدخل والخرج في جميع البنود تكون $\{0,1\}$:

1) يكون الجواب على الرمزين الاختياريين الأوليين 0 . أما الرموز n من باقي الخرج فتساوي رمز الدخل (n-2) .

2) يعطى الخرج 1 حالما يكون العنصر الأول في متوالية الرموز 1 والمتبوعة بالمجموعة 00 ، وفي بقية الأوضاع تعطي 0 .

3.1.3. حاسبة بثلاثة حالات داخلية تقابل متوالية الدخل 00001000100010 بمتوالية الخرج 01010000101001 . أوجد مثل هذه الحاسبة . وتحقق فيما إذا يوجد حاسبة بحالتي دخل تحقق هذه الخواص .

4.1.3. لتكن $M=(S,I,F,\delta,\lambda)$ حاسبة ميلي و $s\in S$. نعرف الدالة المحققة بالحاسبة M انطلاقا من الحالة s (ونرمز لها φ_M^s) حيث :

$$1)\quad \varphi_M^s(\varepsilon)=\varepsilon,$$

$$2)\quad \varphi_M^s(i)=\lambda(s,i)\ ,\ \forall i\in I,$$

$$3)\quad \varphi_M^s(ui)=\varphi_M^s(u)\ \lambda(s,ui)\ ,\ \forall i\in I, u\in I^+.$$

برهن أنه من أجل كل حاسبتين

$$M'=(S',I,F',\delta',\lambda') \quad \text{و} \quad M=(S,I,F,\delta,\lambda)$$

ومن أجل كل $s\in S$ و $s'\in S'$ يكون :

$$1)\quad s \sim_m s' \Leftrightarrow \varphi_M^{s} = \varphi_M^{s'} \,,$$

$$2)\quad M' \subset_{\sim m} M \Leftrightarrow \{\varphi_{M'}^{s'} \,;\, s' \in S'\} \subseteq \{\varphi_M^{s} \,;\, s \in S\}.$$

5.1.3. أنشئ لحاسبة ميلي المعطاة في المثال a.3.1.3) حاسبة مور m- المكافئة لها ، (أي المكافئة مسلكيا) .

6.1.3. أنشئ مختزلا الحاسبتين :

a)

	0	1	
s_0	s_1	s_2	0
s_1	s_1	s_3	0
s_2	s_4	s_1	0
s_3	s_0	s_1	0
s_4	s_2	s_0	1
s_5	s_4	s_3	0

b)

	0	1	
s_0	s_1	s_2	1
s_1	s_3	s_0	0
s_2	s_4	s_5	0
s_3	s_0	s_2	0
s_4	s_2	s_5	0
s_5	s_0	s_3	0

7.1.3. أوجد حاسبة مور المختزلة m–المكافئة للحاسبة :

a)

	0	1
s_0	$s_0,0$	$s_2,0$
s_1	$s_3,1$	$s_4,0$
s_2	$s_0,0$	$s_1,1$
s_3	$s_4,0$	$s_5,1$
s_4	$s_5,0$	$s_5,0$
s_5	$s_2,1$	$s_4,0$

b)

	0	1
s_0	$s_1,0$	$s_2,0$
s_1	$s_3,0$	$s_4,0$
s_2	$s_4,0$	$s_0,0$
s_3	$s_1,0$	$s_0,0$
s_4	$s_5,0$	$s_6,0$
s_5	$s_2,0$	$s_6,0$
s_6	$s_1,0$	$s_{5,1}$

8.1.3. نعرف m – مختزل حاسبة ميلي بأنه حاسبة ، لا يكون فيها أي حالتين مختلفتين m – متكافئتان.

1. برهن أنه يوجد لكل حاسبة ميلي حاسبة m – مختزل بحيث تكون m – متكافئة معها.

2. اقترح خوارزمية للاختزال .

3. أنشئ حاسبة مور بحيث تكون مختزلة ولا تكون b – مختزلة.

4. أوجد m – مختزل للحاسبات في التمرين 7.1.3.

9.1.3. ما هو طول أقصر كلمة دخل ، والتي تميز (مسلكيا) الحالات s_2 , s_6 في الحاسبة التالية :

	0	1	
s_0	s_2	s_1	0
s_1	s_4	s_2	0
s_2	s_0	s_6	0
s_3	s_6	s_5	0
s_4	s_5	s_0	0
s_5	s_4	s_3	1
s_6	s_1	s_6	0

أوجد مثل هذه المتوالية.

10.1.3. برهن أنه في كل حاسبة مختزلة على n حالة ،من أجل كل مجموعة مكونة من r حالة ($2 \le r \le n$) يوجد ثنائية حالات مميزة بكلمة دخل طولها $n - 2$.

11.1.3. أوجد جميع حالات التكافؤ الموجودة بين حالات الحاسبتين التاليتين :

a)

	0	1	
s_0	s_0	s_1	0
s_1	s_1	s_2	1
s_2	s_3	s_1	1
s_3	s_2	s_4	0
s_4	s_2	s_3	0

	0	1	
s_5	s_5	s_6	0
s_6	s_7	s_5	0
s_7	s_7	s_9	1
s_8	s_9	s_8	0
s_9	s_8	s_7	1

b)

	0	**1**	
s_0	s_0	s_1	**1**
s_1	s_2	s_3	**1**
s_2	s_3	s_4	**0**
s_3	s_1	s_0	**0**
s_4	s_3	s_2	**0**

	0	**1**	
s_5	s_6	s_7	**0**
s_6	s_8	s_5	**1**
s_7	s_9	s_6	**1**
s_8	s_5	s_8	**0**
s_9	s_7	s_6	**0**

12.1.3. إذا كان يوجد تشاكل تبولوجي للحاسبة M على الحاسبة M' ، وبالوقت نفسه يوجد تشاكل تبولوجي للحاسبة M' على الحاسبة M ، فهل تكون الحاسبتان M , M' عندئذ متشاكلتين تبولوجيا ؟

13.1.3. لتكن M حاسبة معينة على أربع حالات تبدأ من حالة غير معروفة ، وليكن i أحد رموز الدخل . برهن أن كلمة الدخل المكونة من 29 رمزا i تنقل الحاسبة إلى الحالة نفسها مثل كلمة الدخل iiiii .

2.3. **الدالة التعاقبية** (Sequential Function) .

يمكن فهم الحاسبة التعاقبية كتمثيل منته لمجموعة دوال معينة من مجموعة كلمات إلى مجموعة كلمات . وسنعرض في هذه الفقرة بعض الخصائص التي تميز مثل هذه الدوال .

1.2.3. تعريف.

لتكن $M = (S , I , F , \delta , \lambda)$ حاسبة ميلي . تسمي الدالة $\varphi_M : S \times I^* \to F^*$ دالة التحويل للحاسبة M وتعرف كما يلي :

$$\varphi_M (s , \varepsilon) = \varepsilon \ , \quad \forall s \in S ,$$

$$\varphi_M (s , ui) = \varphi_M (s , u) \ \lambda (s , ui) \ , \quad \forall i \in I , s \in S .$$

2.2.3. تعريف.

نقول إن الدالة $f : B^* \to C^*$ تحقق الحاسبة $M = (S , I , F , \delta , \lambda)$ إذا كان:

1) $B \subseteq I$,

2) $\exists s \in S , \forall u \in B^* : [f (u) = \varphi_M (s , u)]$.

3.2.3. تعريف.

نقول إن الكلمة u مقطع سابق من الكلمة v ونرمز $u \wp v$ (أي يوجد كلمة w بحيث $uw = v$) .

تكون الدالة $f : I^* \to F^*$ دالة تعاقبية إذا كان : $(\forall u , v) \ [u \wp v \Rightarrow f (u) \wp f (v)]$ ،

وتكون حافظة للطول ، إذا كان $(\forall u) [| f (u) | = | u |]$.

4.2.3. نظرية.

كل دالة يمكن تحقيقها بآلة تعاقبية ، تكون تعاقبية وحافظة للطول.

ينتج برهان النظرية مباشرة من التعريف 2.2.3.

5.2.3. تعريف.

لتكن $f : I^* \to F^*$ دالة تعاقبية . نعرف لكل $u \in I^*$ الدالة f_u (وتسمى u– قرينة) كما يلي:

مهما تكن $v \in I^*$ ، فإن $f_u(v)$ تكون كلمة ، حيث $f(uv) = f(u)\, f_u(v)$.

6.2.3. مبرهنة.

لتكن $f : I^* \to F^*$ دالة تعاقبية ، عندئذ :

1. من أجل كل $u \in I^*$ ، تكون الدالة f_u دالة تعاقبية.

2. إذا كانت f حافظة للطول ، عندئذ تحافظ الدالة f_u على الطول من أجل كل $u \in I^*$.

3. من أجل كل $u , v \in I^*$ يكون $(f_u)_v = f_{uv}$.

4. من أجل كل $i_1 i_2 \ldots i_n \in I^*$ يكون

$$f(i_1 i_2 \ldots i_n) = f(\wedge)\ f_\wedge(i_1)\ f_{i_1}(i_2) \ldots f_{i_1 \ldots i_{n-1}} (i_n) .$$

البرهان.

1. $v \wp w \Rightarrow uv \wp uw \Rightarrow f(uv) \wp f(uw) \Rightarrow f(u)\, f_u(v) \wp f(u) f_u(v)$
$\Rightarrow f_u (v) \wp f_u(w)$

2. $| u | + | v | = | uv | = | f(uv) | = |f(u)\, f_u(v) | = | f(u) | + |f_u(v) |.$

وبما أن $| u | = | f(u) |$ ، يكون أيضا $| v | = | f_u (v) |$.

3. $f(uvw) = f(u)\, f_u(vw) = f(u)\, f_u(v)\, (f_u)_v\, (w)$,

$F(uvw) = f(uv)\, f_{uv}(w) = f(u)\, f_u(v)\, f_{uv}(v) \Rightarrow (f_u)_v(w) = f_{uv}(w)$.

ينتج بالتدريج من (3) ومن تعريف الدالة الحافظة .

7.2.3. تعريف.

إذا كانت F مجموعة دوال تعاقبية من I^* إلى F^* ، عندئذ غلق المجموعة F يعطى بالعلاقة :

$$F^{\sim} = F \cup \{ f_u ; f \in F \ \& \ u \in I^* \} .$$

8.2.3. تعريف .

نقول إنه يمكن تحقيق مجموعة الدوال F بالحاسبة M ، إذا أمكن تحقيق كل دالة $f \in F$ بالحاسبة M.

9.2.3. نظرية.

إذا أمكن تحقيق مجموعة الدوال F بالحاسبة M ، عندئذ يمكن بالحاسبة M تحقيق المجموعة $F^{\sim}$.

البرهان . بفرض أنه يمكن تحقيق المجموعة F بالحاسبة

$M = (S , I , F , \delta , \lambda)$.

وليكن $f \in F$, $s \in S$ حيث $(\forall w \in I^*) \ [f(w) = \varphi (s , w)$ عندئذ ،

لكل $u \in I^*$ يكون $(\forall w \in I^*) \ [f_u(w) = \varphi(\delta (s , u) , w)]$.

10.2.3. نظرية.

لتكن G مجموعة الدوال التعاقبية من I^* إلى F^* . إذا كانت جميع الدوال من G تحفظ الطول و $Card(G^{\sim}) = n$ ، عندئذ يمكن تحقيق G بحاسبة معينة على n حالة ولا يمكن تحقيقها بأية حاسبة على عدد أقل من الحالات .

البرهان .

ننشئ الحاسبة $M = (S, I, F, \delta, \lambda)$ ، حيث λ , δ معرفة كما يلي :

$$(\forall i \in I, g \in G^{\sim}) \ [\delta(g, i) = g_i \ , \lambda(g, i) = g(i)] .$$

بما أن $G^{\sim}$ مجموعة تعاقبية مغلقة ، فإن الدالة δ تكون معرفة تماما و M حاسبة ميلي .

ويكون واضحا أيضا ، من أجل كل $g \in G$ يكون $(\forall w \in I^*) \ [g(w) = \varphi_M(g , w)]$ وبالتالي M تحقق جميع الدوال G . ومن المعلوم أن كل حاسبة تحقق G تحقق كامل المجموعة $G^{\sim}$. وبما أن الدوال المختلفة في $G^{\sim}$ تتحقق بمساعدة حالات مختلفة للحاسبة ، يجب أن يكون لكل حاسبة عدد من الحالات يساوي $card\ G^{\sim}$ على الأقل

.

4.2.3. تمارين.

1.2.3. تحقق أياً من الدوال التالية تعاقبية وأياً منها حافظة للطول ، حيث جميع هذه الدوال معرفة من $\{0, 1\}^*$ إلى $\{0, 1\}^*$ ، $\forall w \in \{0, 1\}^*$.

a) $f(w) = 0\,w$.

b) $f(w) = w\,0$.

c) $f(w) = w^R$.

d) $f(w) = w\,w$.

e) $f(i_1 \ldots i_n) = j_1 \ldots j_k$, $j_k = 1 - i_k$, $\forall\ k \in \{1, \ldots, n\}$.

2.2.3. لتكن $f : \{0,1\}^* \rightarrow \{0,1\}^*$ دالة معرفة كما يلي :

$$f(i_1 \ldots i_n) = j_1 \ldots j_n , \quad j_k \equiv i_1 + i_2 + \ldots + i_k \ (\mathrm{mod}\ 2) \ , 1 \le k \le n .$$

برهن أن f دالة تعاقبية ، ثم وصَف الدالة القرينة f_{01011} .

3.2.3. نعمم مفهوم حاسبة ميلي كما يلي : بالنسبة للحاسبة $M = (S, I, F, \delta, \lambda)$ ستكون S, I, F, δ كما سبق ، ونعرف λ بالشكل $\lambda : S \times I \rightarrow F^*$. وصَف مجموعة الدوال المميزة بهذه الحاسبة .

3.3. تحليل الحاسبات (Analysis Machines).

عملنا في الأجزاء السابقة على عرض خصائص الحاسبات المنتهية والحاسبات التعاقبية . وسنتعرف في هذا الجزء على منحى أكثر خصوصية في نظرية الحوسبة يعالج مفاهيم تقنية .

سنعرض طريقة تمكننا من تشكيل بعض الحاسبات التعاقبية من عدد من الحاسبات البسيطة . وستعالج هذه الطريقة كيفية البحث عن أشكال معينة لتحليل فضاء الحالات .

لتكن $M = (S, I, F, \delta, \lambda)$ حاسبة منتهية ، و π تحليلاً للمجموعة S ، عندئذ نكتب $s \equiv s' \ (\pi)$ إذا كانت الحلتان s , s' تقعان في الصف نفسه من التحليل π . ومن

المعلوم بأن مجموعة جميع تحليلات المجموعة S تشكل اتحاد (union) بالنسبة لعمليات الوصل "+" والتقاطع " . " والمعرفة كما يلي :

$$s \equiv s' (\pi_1 . \pi_2) \Leftrightarrow_{df} s \equiv s' (\pi_1) \ \& \ s \equiv s' (\pi_2) .$$

$$s \equiv s' (\pi_1 + \pi_2) \Leftrightarrow_{df} (\exists s_1 , \dots , s_n \in S) [s_1 = s \ \& \ s_n = s' \ \& \ (s_i \equiv s_{i+1} (\pi_1) \text{ or } s_i \equiv s_{i+1} (\pi_2)) , \forall \ 1 \le i \le n]$$

1.3.3. تعريف .

نقول إن التحليل π لفضاء حالات الحاسبة التعاقبية $M = (S , I , F , \delta , \lambda)$ يملك خاصية إبدالية (substitution property) ، ونرمز لها sp - تحليل ، إذا كان

$$s \equiv s' (\pi) \Rightarrow [\forall i , \ \delta (s , i) \equiv \delta (s' , i) (\pi)] .$$

2.3.3. نظرية.

مجموعة جميع sp - التحليلات لفضاء حالات حاسبة تعاقبية تشكل اتحاداً جزئياً من اتحاد جميع تحليلات فضاء الحالات . وهذا الاتحاد الجزئي يحوي أيضا التحليلات البدائية I , O ، حيث O تحليل على مجموعة وحيدة العنصر ، I تحليل مكون من صف وحيد يضم فضاء الحالات كاملاً .

البرهان:

ليكن π , τ , sp – تحليلين لفضاء حالات الحاسبة التعاقبية $M = (S, I, F, \delta, \lambda)$.

1. إذا كان $s \equiv s'(\pi . \tau)$ ، عندئذ يكون $s \equiv s'(\pi)$ & $s \equiv s'(\tau)$
وبالتالي

$$(\forall i)[\delta(s,i) \equiv \delta(s',i)(\pi) \;\&\; \delta(s,i) \equiv \delta(s',i)(\tau)]$$

وهذا يعني أن $\pi . \tau$ يشكل sp - تحليل .

2. إذا كان $s \equiv s'(\pi + \tau)$ ، أي

$$(\exists s_1, \dots, s_n \in S)\ [s_1 = s \;\&\; s_n = s' \;\&\; (s_j \equiv s_{j+1}(\pi) \text{ or } s_j \equiv s_{j+1}(\tau)),\quad \forall\ 1 \le j \le n].$$

عندئذ يكون بالنسبة لمثل هكذا متوالية $s_1, \dots, s_n$ ،

$$(\forall i)\ [\delta(s_j, i) \equiv \delta(s_{j+1}, i)(\pi) \text{ or } \delta(s_j, i) \equiv \delta(s_{j+1}, i)(\tau),\quad \forall\ 1 \le j \le n].$$

وبالتالي يكون $(\forall i)[\delta(s,i) \equiv \delta(s',i)(\pi + \tau)]$.

ينتج مما سبق أن $\pi + \tau$ يشكل sp - تحليل . وأن كل من I , O يشكل sp - تحليل ، ينتج مباشرة من التعريف.

3.3.3. تعريف.

لتكن M' , M حاسبتي ميلي . نقول إن M' تحقق M (realization) ، إذا كان يوجد تشاكل تبولوجي متباين من M إلى M' .

4.3.3. تعريف .

لتكن $M_1 = (S_1 , I_1 , F_1 , \delta_1 , \lambda_1)$ و $M_2 = (S_2 , I_2 , F_2 , \delta_2 , \lambda_2)$

حاسبتي ميلي و $F_1 \subseteq I_2$، عندئذ يشكل الوصل التسلسلي (sequential connection)

للحاسبتين M_1 , M_2 حاسبة جديدة على النحو التالي

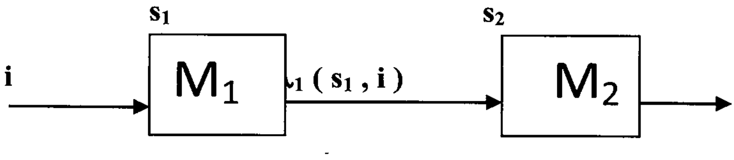
$M_1 \oplus M_2 = (S_1 \times S_2 , I_1 , F_2 , \delta , \lambda)$ ، حيث

$$\delta ((s_1 , s_2) , i) = (\delta_1 (s_1 , i) , \delta_2 (s_2 , \lambda_1 (s_1 , i))) ,$$

$$\lambda ((s_1 , s_2) , i) = (\lambda_2 (s_2 , \lambda_1 (s_1 , i)) .$$

وذلك مهما تكون $s_1 \in S_1 , \; s_2 \in S_2 , \; i \in I_1$.

يمكن تمثيل الوصل التسلسلي $M_1 \oplus M_2$ بالشكل 5.3.



(شكل 5.3)

5.3.3. تعريف.

إذا كانت الحاسبة $M_1 \oplus M_2$ تحقق الحاسبة M ، نقول إن $M_1 \oplus M_2$ تحليل تسلسلي للحاسبة M. ويكون هذا التحليل غير بدائي إذا كانت كل من الحاسبتين M_1 ، M_2 تحوي عدداً من الحالات أقل من الحاسبة M .

6.3.3. **نظرية.**

يوجد لحاسبة ميلي M تحليل تسلسلي غير بدائي ، إذا وفقط إذا كان يوجد sp– تحليل غير بدائي لمجموعة حالات M .

البرهان.

1) لتكن $M = (S , I , F , \delta , \lambda)$ حاسبة و $M_1 \oplus M_2$ تحليل تسلسلي غير بدائي لها ، حيث

$M_1 = (S_1 , I_1 , F_1 , \delta_1 , \lambda_1)$ و $M_2 = (S_2 , I_2 , F_2 , \delta_2 , \lambda_2)$ ، وليكن

$b = (b_S , b_I , b_F)$ تشاكلاً تبولوجياً متبايناً من M إلى $M_1 \oplus M_2$.

نعرف التحليل π للمجموعة S بالشكل التالي :

$s \equiv s' (\pi) \Leftrightarrow_{df} s_1 = s'$ ، حيث $b_S (s) = (s_1 , s_2)$

و $b_S (s') = (s_1' , s_2')$.

من السهل التحقق من أن التحليل π يملك خاصية إبدالية ، وأن π غير بدائي ، لأن $b_S : S \to S_1 \times S_2$ دالة متباينة

وأن $\max (card\ S_1 , card\ S_2) < card (S)$.

2) ليكن π ، sp – تحليل غير بدائي للمجموعة S . ننشئ التحليل τ للمجموعة S ، حيث يكون $\pi . \tau = O$ و τ يحوي عدداً من الصفوف يساوي عدد عناصر الصف الأكبر من التحليل π . نأخذ

$M_1 = (\pi , I , \pi \times \tau , \delta_\pi , id)$ و $M_2 = (\tau , \pi \times I , F , \delta_2 , \lambda_2)$ ، حيث :

a) . $\delta_\pi ([s]_\pi , i) =_{df} [\delta (s , i)]_\pi$ ، من أجل كل $s \in S , i \in I$ ، تكون الدالة δ_π معرفة بشكل ملائم ، لأن π يملك خاصية إبدالية .

b) . $\delta_2 (T_j , (P , i)) =_{df} T_k$ ، من أجل جميع $T_j , T_k \in \tau$ و $P \in \pi$ ، تحقق الخاصية

$\delta (T_j \cap P , i) \in T_k$. وفي بقية النقاط نعرف δ_2 بشكل اختياري .

c) . $\lambda_2 (T , (P , i)) =_{df} \lambda (T \cap P , i)$ ، من أجل $T \cap P \neq \varnothing$ ، وفي بقية النقاط تعرف λ_2 بشكل اختياري .

تعرف الدالتان δ_2 , λ_2 بشكل ملائم ، لأن $T \cap P$ تحوي مجموعة وحيدة العنصر على الأكثر حيث ، $P \in \pi , T \in \tau$ اختيارية .

ويكون للحاسبتين M_1 , M_2 عدد من الحالات أقل من الحاسبة M .

أخيرا يمكن احتواء M في $M_1 \oplus M_2$ بالتشاكل التبولوجي $b = (b_S , b_I , b_F)$ ، حيث

$$b_S (s) =_{df} (P , I) \quad , \quad b_F = id_F \ , \quad b_I = id_I \ ,$$

و $P \in \pi$, $T \in \tau$ تحقق $P \cap T = \{ s \}$.

من الواضح أن b دالة متباينة . ونترك للقارئ التحقق من أن b تشاكل تبولوجي .

7.3.3. تعريف.

ليكن $M_1 = (S_1 , I_1 , F_1 , \delta_1 , \lambda_1)$ و $M_2 = (S_2 , I_2 , F_2 , \delta_2 , \lambda_2)$ حاسبتي ميلي ، وليكن $b : F_1 \times F_2 \to F$ ، حيث F أبجدية اختيارية . عندئذ يشكل الوصل التفرعي (parallel connection) للحاسبتين M_2 , M_1 مع الرابطة b الحاسبة الآتية

$$M_1 | b | M_2 = (S_1 \times S_2 , I_1 \times I_2 , F , \delta , \lambda) ,$$

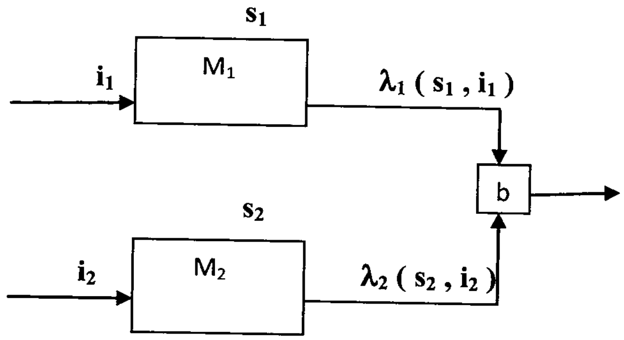
حيث :

$$\delta ((s_1 , s_2) , (i_1 , i_2)) = (\delta_1 (s_1 , i_1) , \delta_2 (s_2 , i_2))) ,$$

$$\lambda ((s_1 , s_2) , (i_1 , i_2)) = b (\lambda_1 (s_1 , i_1) , \lambda_2 (s_2 , i_2)) ,$$

من أجل كل $s_1 \in S_1$, $s_2 \in S_2$, $i_1 \in I_1$, $i_2 \in I_2$.

يمكن تمثيل الوصل التفرعي $M_1 | b | M_2$ بالشكل 6.3.



(شكل 6.3)

إذا كانت الحاسبة $M_1 | b | M_2$ تحقق الحاسبة M ، عندئذ نقول إن $M_1 | b$
$| M_2$ تشكل تحليلاً تفرعياً للحاسبة M . وإذا كان عدد حالات الحاسبتين M_1 , M_2
أصغر من عدد حالات M ، يكون هذا التحليل التفرعي غير بدائي .

9.3.3. نظرية.

يكون للحاسبة M تحليل تفرعي غير بدائي ، إذا وفقط إذا كان يوجد - sp
تحليلان غير بدائيين π ، τ لفضاء حالات M ، حيث $\pi . \tau = O$.

البرهان. لتكن $M = (S , I , F , \delta , \lambda)$ حاسبة معطاة .

1) نفرض أنه يوجد حاسبتان $M_1 = (S_1 , I_1 , F_1 , \delta_1 , \lambda_1)$

و $M_2 = (S_2 , I_2 , F_2 , \delta_2 , \lambda_2)$ ، ودالة b حيث يكون

$\max (card\ S_1 , card\ S_2) < card (S)$. كما نفرض وجود تشاكل تبولوجي متباين $g = (g_S , g_I , g_F)$ من الحاسبة M إلى الحاسبة $M_1 | b | M_2$.

نعرف على S التحليلين π , τ حيث :

$$\forall (g_S (s) = (s_1 , s_2) , g_S (s') = (s_1' , s_2')) : s \equiv s' (\pi) \Leftrightarrow_{df} s_1 = s_1'$$

$$\forall (g_S (s) = (s_1 , s_2) , g_S (s') = (s_1' , s_2')) : s \equiv s' (\tau) \Leftrightarrow_{df} s_2 = s_2'$$

التحليلان τ , π غير بدائيين ، لأن M_1 , M_2 تملكان عدداً من الحالات أقل من M .

ينتج مباشرة ، من تعريف τ , π وحقيقة أن g_S متباينة ، أن $\pi . \tau = O$. كلا التحليلين يملكان خاصية إبدالية .

2) نفرض وجود sp– تحليلين غير بدائيين π ، τ على الحاسبة M بحيث يكون

$$\pi . \tau = O$$

نعرف الحاسبتين

$$M_1 = (\pi , I , \pi \times I , \delta_\pi , id) \quad , M_2 = (\tau , I , \tau \times I , \delta_\tau , id)$$

على النحو الآتي :

$$\forall s \in S , i \in I \quad : \quad \delta_\pi ([s]_\pi , i) = [\delta (s , i)]_\pi \quad ,$$

$$\delta_\tau ([s]_\tau , i) = [\delta (s , i)]_\tau \quad .$$

كلا الحاسبتين معرفتان تماما ، لأن τ , π يمثلان sp– تحليلين . كما نعرف الدالة $b : (\pi \times \tau) \to F$ بالشكل الآتي :

$$\forall s \in S, i \in I \quad : \quad b(([s]_\pi, i), ([s]_\tau, i)) = \lambda(s, i).$$

تنتج صحة التعريف من الفرض بأن $\pi . \tau = O$. وفي بقية الحالات نعرف الدالة b اختياريا .

عندئذ يمكن احتواء الحاسبة M في $M_2 \mid b \mid M_1$ بالتشاكل التبولوجي $g = (g_S, g_I, g_F)$ ، حيث

$$\forall s \in S: \; g_S(s) = ([s]_\pi, [s]_\tau),$$

$$\forall i \in I: g_I(i)(i, i),$$

$$g_F = id_F.$$

لقد برهنا أنه يمكن الحصول على جميع التحليلات التسلسلية والتفرعية الممكنة للحاسبة ، عندما نعلم جميع sp – التحليلات لفضاء حالاته . ولإيجاد جميع -sp التحليلات نتدرج وفق الخوارزمية الآتية :

10.3.3. خوارزمية تشكيل -sp تحليلات الحاسبة M .

لتكن $M = (S, I, F, \delta, \lambda)$ حاسبة معطاة . نرمز بـ- $< s_1, s_2 >$ لأصغر (لأنعم) sp – تحليل π ، حيث $s_1 \equiv s_2 (\pi)$. عندئذ نحصل على جميع sp - تحليلات لفضاء الحالات S كما يلي :

(1) . لكل ثنائية $s_1, s_2 \in S$ ننشئ $< s_1, s_2 >$.

(2) . نشكل جميع المجاميع المنتهية الممكنة (أي الوصل) للتحليلات $< s_1 , s_2 >$.

من النظرية 2.3.3. ينتج بالطريقة المقدمة ،أنه بعد عدد منته من التكرارات نحصل على جميع sp- التحليلات . نوضح ذلك بعرض المثال البسيط الآتي :

11.3.3. مثال.

لتكن $M = (S , I , F , \delta , \lambda)$ حاسبة معطاة ، حيث $I = \{ a , b , c \}$ ، $S = \{ 1 , 2 , 3 , 4 , 5 , 6 \}$, ودالة النقل δ معطاة بالجدول الآتي :

δ	a	b	c
1	6	3	2
2	5	4	1
3	2	5	4
4	1	6	3
5	4	1	6
6	3	2	5

ننشئ جميع sp - التحليلات لفضاء الحالات S .

$$< 1 , 2 > = \{ 1 , 2 \} , \{ 3 , 4 \} , \{ 5 , 6 \} .$$

نحصل على التحليل $< 1 , 2 >$ على النحو الآتي :

إذا كان $1 \equiv 2 (\pi)$ بالنسبة لـ sp– تحليل ما π ، عندئذ يكون $3 \equiv 4 (\pi)$ و $5 \equiv 6 (\pi)$ ، لأن $3 = \delta (1 , b)$, $4 = \delta (2 , b)$.

وبشكل مشابه يكون $5 = \delta (2 , a)$, $6 = \delta (1 , a)$. وبذلك يكون التحليل $\{ 1 , 2 \} , \{ 3 , 4 \} , \{ 5 , 6 \}$ - sp تحليلاً للمجموعة S ، ويكون كذلك أصغر (أنعم) sp - تحليلاً يحوي الحالتين 1 و 2 في الصف نفسه . وبالطريقة نفسها نحصل على جميع sp - التحليلات بالشكل $< s , s' >$.

$$< 1 , 2 > = \{ 1 , 2 \} , \{ 3 , 4 \} , \{ 5 , 6 \} ,$$

$$< 1 , 3 > = \{ 1 , 3 , 5 \} , \{ 2 , 4 , 6 \} ,$$

$$< 1 , 4 > = \{ 1 , \cancel{2} , 3 , 4 , 5 , 6 \} = I ,$$

$$< 1 , 5 > = \{ 1 , 3 , 5 \} , \{ 2 , 4 , 6 \} = < 1 , 3 > ,$$

$$< 1 , 6 > = I ,$$

$$< 2 , 3 > = I ,$$

$$< 2 , 4 > = < 1 , 3 > ,$$

$$< 2 , 5 > = I ,$$

$$< 2 , 6 > = < 1 , 3 > ,$$

$$< 3 , 4 > = < 1 , 3 > ,$$

$$< 3 , 5 > = < 1 , 3 > ,$$

$$< 3 , 6 > = I ,$$

$$< 4 , 5 > = I ,$$

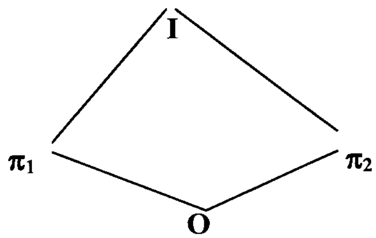
$< 4, 6 > = < 1, 3 >,$

$< 5, 6 > = < 1, 2 > .$

لقد حصلنا على أربع sp - تحليلات أساسية هي :

$\pi_1 = < 1, 2 >$, $\pi_2 = < 1, 3 >$, I , O .

بما أن $\pi_1 + \pi_2 = I$ ، فهذا يعني أننا نعلم جميع sp - التحليلات . ومخطط اتحاد sp - التحليلات على S يكون بالشكل :



(شكل 7.3)

بما أن $\pi_1 . \pi_2 = O$ ، سيكون لكل حاسبة بدالة النقل المعطاة تحليل تسلسلي وكذلك تحليل تفرعي غير بدائيين .

12 .3.3 . **مثال** .

اكتب جميع sp - التحليلات لفضاء حالات الحاسبة المعطاة بدالة النقل في الجدول الآتي:

δ	a	b	c	d	e
1	2	1	5	8	3
2	1	2	6	2	4

3	4	3	6	6	1
4	3	4	5	5	2
5	5	6	3	4	7
6	6	5	4	3	8
7	7	8	4	2	5
8	8	7	3	1	6

ننشئ أولا جميع sp– التحليلات الأساسية :

$\langle 1,2\rangle = \{1,2\},\{3,4\},\{5,6\},\{7,8\}$

$\langle 1,3\rangle = \{1,2,3,4\},\{5,6,7,8\}$

$\langle 1,4\rangle = \{1,4\},\{2,3\},\{5,8\},\{6,7\}$

$\langle 1,5\rangle = \langle 1,6\rangle = I$

$\langle 1,7\rangle = \langle 1,8\rangle = \{1,2,7,8\},\{3,4,5,6\}$

$\langle 2,3\rangle = \langle 1,4\rangle$

$\langle 2,4\rangle = \langle 1,3\rangle$

$\langle 2,5\rangle = \langle 2,6\rangle = I$ $\qquad$ $\langle 3,4\rangle = \langle 1,2\rangle$

$\langle 2,7\rangle = \langle 2,8\rangle = \langle 1,7\rangle$ $\qquad$ $\langle 3,5\rangle = \langle 3,6\rangle = \langle 1,7\rangle$

$\langle 4,5\rangle = \langle 4,6\rangle = \langle 1,7\rangle$ $\qquad$ $\langle 3,7\rangle = \langle 3,8\rangle = I$

$\langle 4,7\rangle = \langle 4,8\rangle = I$ $\qquad$ $\langle 5,6\rangle = \langle 1,2\rangle$

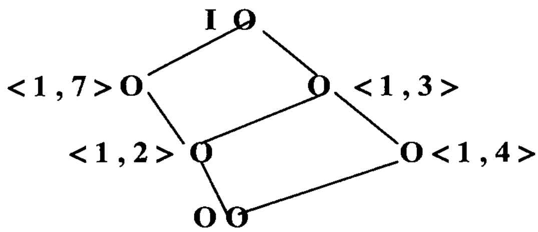
$\langle 5 , 8 \rangle = \langle 1 , 4 \rangle$ $\qquad$ $\langle 6 , 7 \rangle = \langle 1 , 4 \rangle$

$\langle 7 , 8 \rangle = \langle 1 , 2 \rangle$

حصلنا على جميع sp - تحليلات أساسية وعددها 6 هي :

$O , I , \quad \langle 1 , 2 \rangle , \langle 1 , 3 \rangle , \langle 1 , 4 \rangle , \quad \langle 1 , 7 \rangle .$

بمجاميع هذه التحاليل لا نحصل على أي sp – تحليل جديد . ويكون مخطط sp – التحليلات بالشكل الآتي :



(شكل 8.3)

13.3.3. تمارين .

1.3.3. نعرف الدالة $f : \{ 0 , 1 \}^* \rightarrow \{ 0 , 1 \}^*$ بالشكل

$$f (i_1 \dots i_n) = j_1 \dots j_n \quad ,$$

حيث :

$$j_k = \begin{cases} 1 \ , & \sum_{i=1}^{k} t_i \equiv 0 (\mathrm{mod}\, 6), \\ 0 \ , & otherwais \end{cases}$$

أنشئ حاسبة ميلي المختزلة الممثلة لهذه الدالة وعين التحليل المتسلسل والمتفرع لها .

4. الفصل الرابع

اللغات المنتظمة (regular languages).

محتوى الفصل :

1.4. العمليات على اللغات .

2.4. بعض خصائص اللغات الممثلة بحاسبة منتهية.

3.4. اللغات المنتظمة و التعابير المنتظمة .

4.4. التطابق اليميني .

5.4. الحاسبة المنتهية باتجاهين .

6.4. المعادلات المنتظمة .

7.4. متوالية ترتيب (تزامن) الأحداث.

أهداف الفصل :

يهدف هذا الفصل إلى اكتساب المعارف التالية :

العمليات على اللغات المنتظمة . خصائص اللغات الممثلة بحاسبة منتهية . اللغات المنتظمة و التعابير المنتظمة . التطابق اليميني . الحاسبة المنتهية باتجاهين . المعادلات المنتظمة . متوالية ترتيب (تزامن) الأحداث .

1.4. **العمليات على اللغات** .

عرضنا في الفصل الثاني صف τ من اللغات الممثلة بحاسبة منتهية . وسنعالج في هذا الفصل خصائص اللغات. سنتعرف أولا على بعض العمليات الهامة على اللغات . وفي الفقرات التالية سنفرض أن L_1 , L_2 , L لغات اختيارية و w كلمة اختيارية .

تتكون اللغات من مجموعات من الكلمات ، لذا يمكن عرض العمليات الأساسية على المجموعات الآتية :

عملية الاتحاد : $L_1 \cup L_2$.

عملية التقاطع : $L_1 \cap L_2$.

عملية المتمم : $I^* - L_1$ ، حيث $L_1 \subseteq I^*$.

إذا كنا نتحدث عن المتمم ، فيجب أن يكون واضحا عن أية مجموعة I نتحدث .

أما بقية العمليات فستكون خاصة بالنسبة للغات .

1.1.4. **تعريف** .

1) **تسلسل** (ضرب) لغتين L_1 ، L_2 يشكل لغة ويعرف على النحو الآتي :

$$L_1 . L_2 = \{ u v , u \in L_1 \ \& \ v \in L_2 \} .$$

(2

$$L^0 =_{df} \{ \varepsilon \} ,$$

$$L^{n+1} =_{df} L^n . L \quad , \forall \ n \geq 0 .$$

تسمى L^n ، **n - قوة** للغة L .

(3

$$L^+ =_{df} \cup_{i=1}^{\infty} L^i ,$$

$$L^* =_{df} \cup_{i=0}^{\infty} L^i .$$

نسمي L^* تكرار اللغة L ، ونسمي L^+ التكرار الموجب للغة L .
نلاحظ أن التعريف 4.1.2. للمجموعات I^+ , I^* يتوافق مع العمليات المعرفة أعلاه + ، * .

من الواضح أن $\{ \varepsilon \} . L = L . \{ \varepsilon \} = L$, $L . \varnothing = \varnothing . L = \varnothing$.

وبدلا من $\{ w \} . L$ سنكتب أحيانا للسهولة $w . L$ ، وبشكل خاص يكون

$$\varepsilon L = L \varepsilon = L .$$

كذلك من الواضح أن

$$L^+ = L^* - \{ \varepsilon \} .$$

2.1.4. تعريف .

1. **خرج قسمة يساري** للغة L_1 وفق L_2 يعطى باللغة :

$$L_2 \backslash L_1 = \{ u ; wu \in L_1 , w \in L_2 \} .$$

2. **خرج قسمة يميني** للغة L_1 وفق L_2 يعطى باللغة :

$$L_1 / L_2 = \{ u ; uw \in L_1 , w \in L_2 \} .$$

3. **التفاضل اليساري** للغة L وفق w يعطى باللغة :

$$\partial_w L = w \backslash L .$$

4. التفاضل اليميني للغة L وفق w يعطى باللغة :

$$\partial^r_w L = w / L .$$

3.1.4. تعريف .

نعرف صورة الكلمة $w = i_1 i_2 \dots i_n$ بالكلمة $w^R = i_n \dots i_2 i_1$. وبشكل خاص $\varepsilon^R = \varepsilon$.

ونعرف صورة اللغة L بالعلاقة $L^R = \{ w^R , w \in L \}$.

4.1.4. تعريف.

لتكن I أبجدية منتهية . ومن أجل كل $i \in I$ ، لتكن $\sigma(i)$ لغة على أبجدية ما J_i .

وليكن أيضا $\sigma(uv) = \sigma(u) . \sigma(v)$ ، $\sigma(\varepsilon) = \{\varepsilon\}$ ، من أجل كل $u , v \in I^*$.

عندئذ تسمى الدالة $\sigma : I^* \to P(J^*)$ **تبديلاً** ، حيث $J = \cup_{i\in I} J_i$. بالنسبة إلى اللغة $L \subseteq I^*$ نعرف $\sigma(L) = \cup_{w\in L} \sigma(w)$ ونقول إن $\sigma(L)$ تنتج من L بعملية التبديل σ .

يكون التبديل σ غير مطلق ، إذا كان لا يحوي أيا من اللغات $\sigma(i)$ كلمة فارغة . كما نسمي التبديل ، حيث كل من اللغات $\sigma(i)$ يحوي كلمة w_i ، تشاكلاً تبولوجياً .

إذا يمكن التعبير عن التشاكل التبولوجي بالدالة $\sigma : I^* \to J^*$.

إذا كان $\sigma(i) \neq \{\varepsilon\}$ من أجل كل $i \in I$ ، فإننا ندعو σ تشاكلاً تبولوجياً غير مطلق .

وسنتعرف لاحقا على بعض عمليات أخرى على اللغات .

5.1.4. تمارين.

1.1.4. برهن صحة العلاقات التالية ، حيث : L_1 , L_2 , L_3 , L لغات اختيارية و h تشاكل تبولوجي اختياري ، w كلمة ، σ تبديل .

$L_1 (L_2 \cup L_3) = L_1 L_2 \cup L_1 L_3 .$

$(L^*)^* = L^* .$

$L_1 (L_2 \cap L_3) = L_1 L_2 \cap L_1 L_3 .$

$(L_1 \cup L_2)^* = L_1^* (L_2 L_1^*)^* .$

$h(L_1 \cup L_2) = h (L_1) \cup h (L_2) .$

$h(L_1 . L_2) = h (L_1) . h (L_2)$

$\partial_w (L_1 \cup L_2) = \partial_w (L_1) \cup \partial_w (L_2) .$

$\partial_w (I^* - L) = I^* - \partial_w L .$

$(L_1 . L_2)^R = (L_2)^R . (L_1)^R .$

$\sigma (L_1 . L_2) = \sigma (L_1) . \sigma (L_2) .$

2.1.4. برهن أن العلاقات التالية عموما لا تتحقق بالضرورة.

$L_1 . L_2 = L_2 . L_1 .$

$(L_1 \cap L_2)^* = L_1^* \cap L_2^* .$

$(L_1 \cup L_2)^* = L_1^* \cup L_2^* .$

$h(L_1 \cap L_2) = h (L_1) \cap h (L_2) .$

$h (h (L)) = h (L) .$

$(L_1 L_2)^* L_1 = L^*_1 (L_2 L_1)^* .$

$L_2 (L_2 \setminus L_1) = L_1 .$

2.4. بعض خصائص اللغات الممثلة بحاسبة منتهية.

سنبرهن في هذه الفقرة أن مجموعة اللغات الممثلة بحاسبات منتهية ستكون مغلقة بالنسبة لبعض العمليات . وهذا يعني أن كل لغة يمكن تشكيلها بوساطة هذه العمليات انطلاقا من لغة ممثلة بحاسبة منتهية . وسنقدم براهين هذه التأكيدات حيث نتحقق من صحتها وبالوقت نفسه سنقدم طريقة لإنشاء الحاسبات التي تمثل تلك اللغات.

1.2.4. نظرية.

لتكن I أبجدية منتهية ، عندئذ مجموعة اللغات τ على الأبجدية I الممثلة بحاسبات منتهية تشكل جبر بول بالنسبة للعمليات $- \ , \ \cap \ , \ \cup$. أي إذا كانت $L_1 , L_2 \subseteq I^*$ لغات اختيارية فإن :

1) $L_1 \in \tau \Rightarrow I^* - L_1 \in \tau$.

2) $L_1 , L_2 \in \tau \Rightarrow L_1 \cup L_2 \in \tau$.

3) $L_1 , L_2 \in \tau \Rightarrow L_1 \cap L_2 \in \tau$.

البرهان (خوارزمية) .

لتكن L_1 لغة ممثلة بحاسبة منتهية $M_1 = (S_1 , I , \delta_1 , s_{01} , F_1)$ و L_2 لغة ممثلة بحاسبة منتهية $M_2 = (S_2 , I , \delta_2 , s_{02} , F_2)$.

يكون من الواضح :

1) إن اللغة $I^* - L_1$ تكون ممثلة بالحاسبة $(S_1 , I , \delta_1 , s_{01} , S_1 - F_1)$.

2) إن اللغة $L_1 \cup L_2$ تكون ممثلة بالحاسبة المنتهية $M = (S , I , \delta , s_0 , F)$ ، حيث :

$$S =_{df} S_1 \times S_2 \quad , \; s_0 =_{df} (s_{01} \times s_{02}) \; , \; F =_{df} (F_1 \times S_2) \cup (S_1 \times F_2) ,$$

والدالة δ معرفة على النحو الآتي :

$$\forall ((s_1 , s_2) \in S , i \in I) :$$

$$\delta ((s_1 , s_2) , I) =_{df} (\delta_1 (s_1 , i) , \delta_2 (s_2 , i)) .$$

من الواضح أن M حاسبة منتهية ، ولنبرهن أنها تمثل تماما $L_1 \cup L_2$.

من أجل $w \in I^*$ كلمة اختيارية ، يكون :

$$\delta ((s_{01} , s_{02}) , w) = (\delta_1 (s_{01} , w) , \delta_2 (s_{02} , w)) ,$$

ومن السهل التحقق من أن :

$$w \in L_1 \cup L_2 \Leftrightarrow w \in L_1 \text{ or } w \in L_2 \Leftrightarrow$$

$$\delta_1 (s_{01} , w) \in F_1 \text{ or } \delta_2 (s_{02} , w) \in F_2 \Leftrightarrow$$

$$(\delta_1 (s_{01} , w) , \delta_2 (s_{02} , w)) \in (F_1 \times S_2) \cup (S_1 \times F_2) \Leftrightarrow$$

$$\delta (s_0 , w) \in F .$$

ينتج فعلا أن $L (M) = L (M_1) \cup L (M_2)$.

3) بشكل مشابه نبين أن اللغة $L_1 \cap L_2$ ممثلة بحاسبة منتهية $M' = (S , I , \delta , s_0 , F')$ ، حيث S , I , δ , s_0 تكون كما في الفقرة (2) و $F' =_{df} F_1 \times F_2$.

تطبق النظرية السابقة والنظرية الآتية غالبا ، أثناء تشكيل الحاسبات المنتهية والتي تمثل لغة معطاة. يكون للغة غالبا ميزات خاصة تحقق كلماتها . إذا كانت هذه الميزات معطاة مثلا بوساطة الاتحاد أو التقاطع أو النفي لخواص بسيطة ، عندئذ تكون تطبيقات النظرية واضحة .

مثال .

إذا أردنا إنشاء حاسبة منتهية تقبل الكلمات من الأبجدية { 1 , 0 } ، والتي :

a) تحوي عدداً زوجياً من الرموز 1 ،

b) أطوالها تقبل القسمة على 7 ،

عندئذ تقدم لنا النظرية السابقة إمكانية إنشاء حاسبتين ، إحداها تقبل الكلمات التي تحقق الشرط (a) ، والثانية تقبل الكلمات التي تحقق (b) . وبلتباع خطوات الإجراء الوارد في برهان النظرية 1.2.4 ننشئ الحاسبة المطلوبة .

2.2.4. نظرية .

لتكن I أبجدية منتهية ، عندئذ من أجل $L_1 , L_2 \subseteq I^*$ لغتين اختياريتين يكون :

$$1)\ L_1 , L_2 \in \tau \Rightarrow L_1 . L_2 \in \tau ,$$

$$2)\ L_1 \in \tau \Rightarrow L^*_1 \in \tau .$$

البرهان (خوارزمية).

لتكن $M_1 = (S_1 , I , \delta_1 , s_{01} , F_1)$ و $M_2 = (S_2 , I , \delta_2 , s_{02} , F_2)$ حاسبتين منتهيتين ، حيث :

$L (M_1) = L_1 ,\ L (M_2) = L_2$ ، كما نفرض أن $S_1 \cap S_2 = \varnothing$.

1) ننشئ الحاسبة المنتهية غير المحدودة $M = (S , I , \delta , B , F)$ ، وفق الإجراء الآتي :

$L (M) = L_1 . L_2$,

$S =_{df} S_1 \cup S_2$,

$F =_{df} F_2$,

$$B =_{df} \begin{cases} \{S_{01}\}, & S_{01} \notin F_1 \\ \{S_{01}, S_{02}\}, & S_{01} \in F_1 \end{cases}$$

$$\delta(S,\ i) = \begin{cases} \{\delta_1(S,\ i)\}, & S \in S_1,\ \delta_1(S,\ i) \notin F_1\ , \\ \{\delta_1(S,\ i),\ S_{02}\}, & S \in S_1,\ \delta_1(S,\ i) \in F_1\ , \\ \{\delta_2(S,\ i)\}, & S \in S_2 \quad . \end{cases}$$

نترك برهان صحة العلاقة $L (M) = L (M_1) . L (M_2)$ للقراء كتمرين ، وحسب النظرية 4.4.2 يكون $L_1 . L_2 \in \tau$.

2) ننشئ حاسبة منتهية غير محدودة $M^{\sim} = (S^{\sim} , I , \delta^{\sim} , B^{\sim} , F^{\sim})$ ، حيث يكون $L (M^{\sim}) = L^*_1$ وفق الإجراء الآتي :

$S^{\sim} =_{df} S_1 \cup \{ q , r \} \quad , q , r \notin S_1$,

$B^{\sim} =_{df} \{ s_{01} , q \}$,

$\widetilde{F} =_{df} F_1 \cup \{ q \}$,

$$\widetilde{\delta}(S,\ i) = \begin{cases} \{\delta_1(S,\ i)\}, & S \in S_1,\ \delta_1(S,\ i) \notin F_1, \\ \{\delta_1(S,\ i),\ S_{01}\}, & S \in S_1,\ \delta_1(S,\ i) \in F_1, \end{cases}$$

$$\tilde{\delta}(q,i) = \{r\},\ \tilde{\delta}(r,i)\} = \{r\} \quad , \ \forall i \in I .$$

نترك برهان صحة العلاقة $L(\tilde{M}) = L^*_1$ للقارئ . والحالات q , r تضمن احتواء اللغة الممثلة بالحاسبة M للكلمة الفارغة .

3.2.4. نظرية.

لكل لغة L تتحقق العلاقة $L \in \tau \Leftrightarrow L^R \in \tau$.

البرهان (خوارزمية).

نعلم أنه لكل لغة L تتحقق العلاقة $(L^R)^R = L$.

إذا يكفي برهان أن $L \in \tau \Rightarrow L^R \in \tau$ بالنسبة لكل L . عندئذ يكون:

$L^R \in \tau \Rightarrow (L^R)^R \in \tau$. ولذلك نتبع الإجراء التالي :

ليكن $L \in \tau$. عندئذ تكون L ممثلة بحاسبة ما $M = (S, I, \delta, s_0, F)$.

على أساس M ننشئ حاسبة غير محدودة $N = (S, I, \delta', B, F')$ مميزة للغة L^R . بمعنى أوضح، نحصل على مخطط الحالات للحاسبة غير المحدودة N انطلاقا من مخطط الحاسبة M ، وذلك بأن " نعكس أسهم " الحالات النهائية في الحاسبة M ونعدّها حالات أولية في الحاسبة N . أي

$$B =_{df} F \quad , \quad F' =_{df} \{s_0\} ,$$

$$s^- \in \delta'(s, i) \Leftrightarrow \delta(s^-, ii) = s .$$

عندئذ يكون

a) $\varepsilon \in L(M) \Leftrightarrow s_0 \in F \Leftrightarrow B \cap F' \neq \varnothing \Leftrightarrow \varepsilon \in L(N)$.

b) $i_1 \ldots i_n \in L(M) \Leftrightarrow \exists\, s_0, \ldots, s_n \in S :$

$$s_n \in B \ \&\ s_{i+1} = \delta(s_i, s_{i+1} , \quad 0 \leq i \leq n-1 , \Leftrightarrow$$

$$\Leftrightarrow \exists\, s_n, \ldots, s_1, s_0 \in S : \ s_n \in B \ \&\ s_i \in \delta'(s_{j+1}, i_{j+1}) ,$$

حيث $0 \le j \le n-1$ ، $s_0 \in F' \Leftrightarrow i_n \dots i_1 \in L(N)$.

ينتج أن $L(M) = [L(M)]^R$. *

تكون في بعض الحالات الحاسبة الممثلة للغة L^R أبسط من الحاسبة الممثلة للغة L . ولهذا السبب ، إذا أردنا إنشاء الحاسبة الممثلة للغة L ، يكون من المفيد إنشاء الحاسبة الممثلة للغة L^R أولا ، ثم ننتقل إلى الحاسبة الممثلة لـ L . وتتم الطريقة الميكانيكية لهذا الانتقال وفق الإجراء الوارد في برهان النظريتين 3.2.4 ، 4.4.2.

عندما سنورد التعابير المنتظمة لاحقا لوصف لغات ممثلة بحاسبات منتهية ، سنرى أن النظرية 3.2.4. واضحة من النظرة الأولى .

4.2.4. تمارين.

1.2.4. برهن على تحقق العلاقات التالية ، من أجل أي لغتين L_1 , L_2 :

a) $L_1 , L_2 \in \tau \Rightarrow L_1 - L_2 \in \tau$.

b) $L \in \tau \Rightarrow L^+ \in \tau$.

c) $L \in \tau \Rightarrow \partial_w (L) \in \tau , \quad \forall w$.

d) $L_1 , L_2 \in \tau \Rightarrow L_1 \setminus L_2 \in \tau$.

e) $L \in \tau \Rightarrow \partial_w^r (L) \in \tau , \quad \forall w$.

f) $L_1 , L_2 \in \tau \Rightarrow L_1 / L_2 \in \tau$.

g) $L \in \tau \Rightarrow h (L) \in \tau$,

حيث h تشاكل تبولوجي اختياري .

2.2.4. يكون الصف τ مغلقا على التبديل . أي إذا كانت الدالة $\sigma : I^* \to P (Y^*)$ تبديل ، حيث $\sigma (i) \in \tau$ من أجل كل $i \in I$ ، عندئذ يكون $L \in \tau \Rightarrow \sigma (L) \in \tau$. برهن ذلك.

3.2.4. إذا كان $h : I^* \to Y^*$ تشاكلاً تبولوجياً ، عندئذ يكون $L \in \tau \Rightarrow h^{-1} (L) \in \tau$، حيث $h^{-1} (L) = \{ w \in I^* ; h (w) \in L \}$

3.4. اللغات المنتظمة و التعابير المنتظمة .

سنتعرف في هذه الفقرة على خصائص هامة أخرى للغات الممثلة بحاسبات منتهية .
وسنبرهن أن اللغات الممثلة بحاسبات منتهية هي ما تسمى اللغات المنتظمة .

1.3.4. تعريف.

يكون صف جميع اللغات المنتظمة $RL(I)$ على الأبجدية I أصغر صف لغات على I يحقق الشروط التالية :

1) $\varnothing \in RL(I)$, $\{i\} \in RL(I)$, $\forall i \in I$.

2) $M, N \in RL(I) \Rightarrow M \cup N \in RL(I)$.

3) $M, N \in RL(I) \Rightarrow M . N \in RL(I)$.

4) $M \in RL(I) \Rightarrow M^* \in RL(I)$.

تسمى أحيانا عمليات " الاتحاد ، الضرب ، المتتم " عمليات منتظمة على اللغات . ويمكن القول فعلا ، إن اللغات المنتظمة هي اللغات التي يمكن الحصول عليها من اللغات المنتهية بتطبيق العمليات المنتظمة عليها .

نلاحظ من التعريف ، أن اللغة التي تحوي الكلمة الخالية فقط تكون أيضا منتظمة ، لأن $\{\varepsilon\} = \varnothing^*$.

2.3.4. تعريف .

نعرف مجموعة التعابير المنتظمة $RV(I)$ على الأبجدية $I = \{ i_1, \dots, i_n \}$ كأصغر مجموعة كلمات في الأبجدية $\{ i_1, \dots, i_n, \varepsilon, \varnothing, +, ., *, (,) \}$ ، حيث $\varepsilon, \varnothing,$ $+, ., *, (,)$ رموز لا تنتمي إلى I ، تحقق الشروط التالية :

1) $\varnothing \in RV(I)$, $\varepsilon \in RV(I)$, $i \in RV(I)$, $\forall i \in I$.

2) $\alpha, \beta \in LV(I) \Rightarrow (\alpha + \beta) \in LV(I)$, $(\alpha . \beta) \in LV(I)$, $\alpha^* \in LV(I)$.

3.3.4. تعريف .

نرمز لقيمة كل تعبير $\alpha \in LV(I)$ بالرمز $[\alpha]$ ونعرّفها بالشكل التالي :

1) $[\varnothing] = \varnothing$; $[\varepsilon] = \{\varepsilon\}$; $[i] = \{i\}$.

2) $[(\alpha + \beta)] = [\alpha] \cup [\beta]$; $[(\alpha . \beta)] = [\alpha] . [\beta]$;

$[\alpha^*] = [\alpha]^*$.

من السهل التحقق ، أن قيمة أي تعبير منتظم تشكل لغة منتظمة ، وبالعكس كل لغة منتظمة يمكن تمثيلها بتعبير منتظم .

4.3.4. نظرية .

كل لغة منتظمة تكون ممثلة بحاسبة منتهية .

البرهان . من السهل بالنسبة لأبجدية منتهية اختيارية I ، إنشاء حاسبة ممثلة للغات $\{ i \}$ $\varnothing$, ε , ، من أجل كل $i \in I$. وبالتالي تكون النظرية نتيجة مباشرة للنظريتين 2.2.4. و 1.2.4.

5.3.4. نظرية .

كل لغة تكون ممثلة بحاسبة منتهية تكون منتظمة .

البرهان . لتكن $M = (S , I , \delta , s_1, F)$ حاسبة منتهية ممثلة للغة L . ولتكن $S = \{ s_1 , \dots , s_n \}$.

نعرّف المجموعة $R_{ij} = \{ w \in I^* ; \delta (s_i , w) = s_j \}$.

من الواضح أن $L (M) = L = \cup \{ R_{1i} ; s_i \in F \}$. $(*)$

يكفي برهان أن كل R_{ij} تشكل لغة منتظمة ، وعندئذ تكون $L (M)$ منتظمة حسب $(*)$.

نعرف R_{ij}^k كمجموعة الكلمات التي تنقل الحاسبة من الحالة s_i إلى الحالة s_j بدون المرور بأية حالة s_m ، حيث $m > k$. من الواضح أن $R_{ij}^0 \subseteq I \cup \{ \varepsilon\}$ ، ولهذا تكون R_{ij}^0 مجموعة منتظمة من أجل $i , j \in \{ 1 , \dots , n \}$.

وينتج هذا من أن $R_{ij}^{k+1} = R_{ij}^k \cup R^k_{i,k+1} . (R^k_{k+1,k+1})^* . R_{k+1,j}$.

وبالتالي نكون قد برهنا بالتدريج أن ، R_{ij}^k تكون منتظمة من أجل كل $i , j , k \in \{ 1 , \dots , n \}$. وبشكل خاص تكون R^n_{ij} منتظمة . وبما أن $R^k_{ij} = R_{ij}$ ، تكون النظرية صحيحة . *

بدمج النظريتين السابقتين نحصل على النتيجة التالية ، التي تسمى نظرية كلين (Kleen) .

6.3.4. نظرية .

تكون اللغة على أبجدية منتهية منتظمة إذا وفقط إذا كانت ممثلة بحاسبة منتهية .

سنوضح معنى هذه النظرية بمثالين ، وسنعتمد فيهما وفي الأمثلة اللاحق ة أفضلية العمليات المنتظمة وحذف الأقواس الزائدة . وتكون الأفضلية الأقوى للعملية * والعملية الأضعف + . وبالتالي يمكن كتابة التعبير

$a + (bc + d)^* a$ بدلا من التعبير $(a +(((bc) + d)^* a))$.

7.3.4. مثال . (التعابير المنتظمة الموسعة).

حسب النظرية a.1.2.4 والنظرية 6.3.4 ، يكون تقاطع لغتين منتظمتين ومتمم لغة منتظمة لغات منتظمة .

ولتشكيل اللغات المنتظمة يمكن استخدام ، ما عدا عمليات " الاتحاد ، التسلسل ، التكرار (عملية نجمة) " ، عملية التقاطع وعملية تعويض اللغة بمتممتها . وبهذا يمكن إغناء مجموعة التعابير المنتظمة بتعابير من الشكل $\alpha \ \& \ \beta$ ، والتي ترمز لتقاطع اللغات الممثلة بالتعابير $\alpha \ , \ \beta$ ، وتعابير بالشكل α^- حيث ترمز لمتمم اللغة الممثلة بالتعبير α . ولهذا نتحدث عن التعابير المنتظمة الموسعة . تساعد مثل هذه التوسيعات في تبسيط وصف اللغة .

مثلاً ، اللغة في الأبجدية $\{ 0 , 1 \}^*$ والمشكلة من جميع الكلمات التي تحوي 1 مرة واحدة على الأقل يمكن وصفها بالعبير $(0^* 1)^*$. ويمكن وصف اللغة على هذه الأبجدية والمشكلة من جميع الكلمات التي تحوي عدداً زوجياً من الرموز 1 وعدداً زوجياً من الرموز 0 بالتعبير $(1^*01^*01^*)^* \ \& \ (0^*1 \ 0^*10^*)^*$.

وبشكل مشابه يمكننا توسيع التعابير المنتظمة على أساس بقية العمليات ، التي يكون عليها الصف τ مغلقا .

8.3.4. مثال . (تكافؤ التعابير المنتظمة) .

نعرّف تكافؤ التعابير المنتظمة ونرمز له $\equiv$ بالشكل

$$\alpha \equiv \beta \Leftrightarrow_{df} [\, \alpha \,] = [\, \beta \,] \ .$$

ونقول عن تعبيرين منتظمين إنهما متكافئان إذا وفقط إذا كانا يرمزان للغة نفسها.

يمكن التحقق من تكافؤ تعبيرين أحيانا بسهولة ، مثلا:

$$(\, 0^{*}1 \,)^{*} \equiv \varepsilon + (\, 0 + 1 \,)^{*}1 \ .$$

ولكن عند التعابير الطويلة تكون المسألة أكثر تعقيدا ، مثلاً

$$(\ 10\ )^{*}\ (\ 00\ )\ (\ 11\ +\ 01(10)^{*}00)^{*}\ \equiv\ (\ 10\ +\ 00(11)^{*}01\ )^{*}00(11\ )^{*}\ \ .$$

ولهذا من الضروري البحث عن خوارزمية تمكننا التحقق من تكافؤ تعبيرين منتظمين أم لا . مثل هذه الخوارزمية تقدمها لنا النظرية 6.3.4. التي يكون وفقها α , β تعبيرين منتظمين اختياريين ممثلين للغتين مميزتين بحاسبتين منتهيتين . وباستخدام العرض الوارد في برهاني النظريتين 2.2.4 و **1.2.4** ، **نستطيع إنشاء الحاسبة المناسبة .**

إذن يمكننا تحويل السؤال عن تحقق علاقة التكافؤ $\alpha \equiv \beta$ ، إلى التحقق من صحة المساواة $L\,(\,M_1\,) = L\,(\,M_2\,)$ بالنسبة للحاسبتين M_1 , M_2 ، حيث يكون من أجلهما:

$L\,(\,M_1\,) = [\,\alpha\,]$, $L\,(\,M_2\,) = [\,\beta\,]$.

وهذه المسألة نستطيع حلها (انظر المثال 18.3.2.) . ولهذا نقدم الخوارزمية الآتية :

9.3.4. خوارزمية .

تصف هذه الخوارزمية طريقة يمكن بموجبها ، لكل مجموعة منتهية من التعابير المنتظمة $\alpha_1 , \ldots, \alpha_m$ على أبجدية منتهية I ، إنشاء حاسبة منتهية بفضاء حالات S وأبجدية دخل I ودالة نقل δ وحالة أولية s_0 ، حيث لكل تعبير α_j يوجد مجموعة $F_j \subseteq S$ يكون من أجلها $M_j = (S , I , \delta , s_0 , F_j)$ حاسبة مميزة لـ $[\alpha_j]$.

سنعرض خطوات هذه الخوارزمية موضحة بمثال .

الخطوة 1. نرمز بـ $L_j = [\alpha_j]$. نعدل التعابير α_j بالطريقة الآتية : لتكن $\alpha_1 , \ldots, \alpha_m$ تحوي عموما رموزاً من I مكررة n مرة . نرقم بالأعداد $1 , 2 , \ldots , n$ جميع تكرارات الرموز من I في التعابير $\alpha_1 , \ldots, \alpha_m$ بشكل اختياري و بحيث تأخذ الرموز المختلفة والتكرارات المختلفة لهذه الرموز أرقام مختلفة .

نحصل بذلك على التعابير $\alpha_1' , \ldots, \alpha_m'$.

عندئذ نقول عن كل تكرار للرمز i_k (حيث $\{ 1 , \ldots , n \}$ $(k \in$ إنه مشتق من i ، ونقول عن التعبير α_j' إنه مشتق من التعبير α_j .

مثلا : إذا كان لدينا التعبيران $\alpha_2 = a c + b^* a b^*$, $\alpha_1 = a b^* a$.

نرقمهما بالشكل $\alpha_2' = a_4 c_5 + b_6^* a_7 b_8^*$, $\alpha_1' = a_1 b_2^* a_3$ ،

(حيث a_1 , a_3 , a_4 , a_7 مشتقة من a و b_2 , b_6 , b_8 مشتقة من b و c_5 مشتقة من c).

تمثل التعابير $\alpha_1' , \ldots, \alpha_m'$ لغات منتظمة في أبجدية جديدة ما I' .

(في مثالنا يكون $I' = \{ a_1 , b_2 , a_3 , a_4 , c_5 , b_6 , a_7 , b_8$) .

الخطوة 2 . نرمز بـ $L_j' = [\alpha_j']$ ، وننشئ المجموعات الآتية :

$C_I =_{df} \{ i^{'} i^{''} \in (I^{'})^2$ & $L_j^{'}$ كلمة جزئية من إحدى كلمات $i^{'} i^{''}$, $1 \leq j \leq m \}$

($i^{'} i^{''}$ كلمة جزئية من w تكافئ وجود كلمتين u , v ، حيث $w = u i^{'} i^{''} v$) .

$C_{II} =_{df} \{ i^{'} ; i^{'} \in I^{'}$ & $L_j^{'}$ سابقة لكلمة ما من $i^{'}$, $1 \leq j \leq m \}$.

$C_{III} =_{df} \{ i^{'} ;$ & $L_j^{'}$ لاحقة لكلمة ما من $i^{'}$, $1 \leq j \leq m \}$.

في مثالنا يكون :

$C_I = \{ a_1b_2 , b_2b_2 , b_2a_3 , a_1a_3 , a_4 c_5 , b_6b_6 , b_6a_7 , a_7b_8 , b_8b_8 \}$.

$C_{II} = \{ a_1 , a_4 , b_6 , a_7 \}$.

$C_{III}^{1} = \{ a_3 \}$.

$C_{III}^{2} = \{ c_5 , a_7 , b_8 \}$.

بوساطة المجموعتين C_I , C_{II} ننشئ حاسبة بالخصائص المطلوبة على النحو الآتي :

$S^{'} = \{ s_0 \} \cup P (I^{'}) \; ; \; s_0 \notin P (I^{'})$.

$\delta^{'} = : S^{'} \times I \rightarrow S^{'}$:

$\delta^{'} (s_0 , i) =_{df} \{ i^{'} \in I^{'} ; i^{'} \in C_{II}$ & i مشتقة من $\} , i \in P(S^{'})$,

$\delta^{'} (s , i) =_{df} \{ i^{''} \in I^{'} ; i i^{''} \in C_I$ بحيث $i^{'} \in s$ ويوجد ، i مشتقة من $i^{'} i^{''} \}$,

عندئذ يكون

$S =_{df} \{ s_0 \} \cup \{ s^{'} ; s^{'} \in S^{'}$ & $\delta^{'}$ قابلة للبلوغ من s_0 وفق الدالة $s^{'} \}$.

$\delta =_{df} \delta^{'} / S \times I$.

في مثالنا يكون :

$S = \{ s_0 , \{ a_1 , a_4 , a_7 \} , \{ b_6 \} , \varnothing , \{ a_3 \} , \{ b_2 , b_8 \} , \{ c_5 \} , \{ a_7 \} , \{ b_8 \} \}$.

و δ معرفة بالجدول التالي :

δ	a	b	c
s_0	$\{ a_1 , a_4 , a_7 \}$	$\{ b_6 \}$	$\varnothing$
$\{ a_1 , a_4 , a_7 \}$	$\{ a_3 \}$	$\{ b_2 , b_8 \}$	$\{ c_5 \}$
$\{ b_6 \}$	$\{ a_7 \}$	$\{ b_6 \}$	$\varnothing$
$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
$\{ a_3 \}$	$\varnothing$	$\varnothing$	$\varnothing$
$\{ b_2 , b_8 \}$	$\{ a_3 \}$	$\{ b_2 , b_8 \}$	$\varnothing$
$\{ c_5 \}$	$\varnothing$	$\varnothing$	$\varnothing$
$\{ a_7 \}$	$\varnothing$	$\{ b_8 \}$	$\varnothing$
$\{ b_8 \}$	$\varnothing$	$\{ b_8 \}$	$\varnothing$

إذا وضعنا :

$$O_j =_{df} \{s \in S; S \cap C_{III}^{j} \neq \phi\} \cup \begin{cases} \{S_0\}, & if \ \varepsilon \in L_j, \\ \phi, & otherwais \end{cases}$$

$M_j (S , I , \delta , s_0 , F_i)$, $L_j = T (M_j)$.

في مثالنا يكون :

$O_1 = \{\{ a_3 \}\}$

$O_2 = \{ \{ a_1 , a_4 , a_7 \} , \{ b_2 , b_8 \} , \{ c_5 \} , \{ a_7 \} , \{ b_8 \} \}$.

10.3.4. تمارين.

1.3.4. ميز اللغات المعطاة في التمرين 3.1.2. بتعابير منتظمة.

2.3.4. تحقق فيما إذا كانت العلاقات التالية محققة :

a) $(011 + (10)^* 1 + 0)^* \equiv 011\,(011 + (10)^* 1 + 0)^*$.

b) $((1 + 0)^* 100\,(1 + 0)^*)^* \equiv ((1 + 0)\,100\,(1 + 0)^* 100)^*$.

c) $(1 + 0)^* 001\,\overline{(1 + 0)^* \equiv ((1 + 0)\,000\,(}\,1 + 0)^*$.

3.3.4. برهن صحة النظرية 3.2.4. مباشرة من النظرية 6.3.4.

(توجيه : برهن أن $(L_1 . L_2)^R = L2^R . L_1^R$ الخ ..)

4.3.4. استخدم النظرية 3.2.4. لبرهان التأكيدين في التمرين 1.2.4. (g , h) .

5.3.4. أنشئ التعبير المنتظم ، الذي يمثل لغة مميزة بحاسبة غير محدودة M ، مثال 3.4.2.

6.3.4. برهن أن التعابير التالية محققة بالنسبة لتعابير اختيارية :

a) $\alpha(\beta + \Upsilon) \equiv \alpha\beta + \alpha\Upsilon$

b) $\alpha + (\beta + \Upsilon) \equiv (\alpha + \beta) + \Upsilon$

c) $\alpha(\beta\ \Upsilon) \equiv (\alpha\beta)\Upsilon$

d) $\alpha\varepsilon \equiv \varepsilon\alpha \equiv \alpha$

e) $\varnothing.\alpha \equiv \alpha.\varnothing \equiv \varnothing$

f) $\alpha + \alpha \equiv \alpha$

g) $(\alpha + \beta)\Upsilon \equiv \alpha\Upsilon + \alpha\Upsilon$

h) $\varnothing^* \equiv \varepsilon$

i) $\alpha^* + \alpha \equiv \alpha^*$

j) $(\alpha^*)^* \equiv \alpha^*$

k) $(\alpha + \beta)^* \equiv (\alpha^* \beta^*)^*$

l) $\alpha + \varnothing \equiv \alpha$

4.4. التطابق اليميني .

لقد بينا حتى الآن ، أن اللغات المنتظمة تميَّز بثلاث طرق هي : التعابير المنتظمة ، الحاسبات المنتهية ، والحاسبات المنتهية غير المحددة . وسنتعرف في الفصل الخامس على كيفية تمييز هذه اللغات بمساعدة صف معين من القواعد . وسنعرض فيما يلي طريقة مفيدة لتمييز اللغات:

1.4.4. تعريف.

لتكن I أبجدية منتهية و $\sim$ علاقة تكافؤ على I^* . عندئذ تسمى العلاقة $\sim$ تطابقاً يمينياً إذا تحقق الشرط

$$(\forall u , v \in I^*) \ u \sim v \Rightarrow u w \sim v w .$$

نبين العلاقة بين اللغات المنتظمة والتطابق اليميني بالنظرية التالية. كما نقول إن التكافؤ على المجموعة A يمثل دليلاً منتهياً ، إذا كان تحليل المجموعة A وفق هذا التكافؤ يحوي عدداً منتهياً من الصفوف .

2.4.4. نظرية .

لتكن L لغة على أبجدية منتهية I ، عندئذ تكون L مميزة بحاسبة منتهية إذا وفقط إذا وجد تطابق يميني بدليل منته ، حيث L تمثل اتحاد صفوف معينة لتحليل I^* وفق $\sim$.

البرهان.

1) لتكن L مميزة بحاسبة منتهية $M = (S , I , \delta , s_0 , F)$. نعرف علاقة التكافؤ $\sim$ بالشكل

$$u \sim v \Leftrightarrow \delta (s_0 , u) = \delta (s_0 , v) .$$

من الواضح أن $\sim$ تشكل تكافؤا ً، وأن الدليل المنته أو عدد صفوف التكافؤ يساوي عدد حالات الحاسبة M

(وبصورة أدق ، عدد الحالات القابلة للبلوغ) ، كما أن $\sim$ تطابق يميني . لأن

$$\delta (s_0 , u) = \delta (s_0 , v) \Rightarrow \delta (s_0 , uw) = \delta (s_0 , vw) \ ,$$

$$(\forall u , v , w \in I^*) .$$

وبما أن

$$L = \{ w \in I^* ; \delta (s_0 , w) \in F \} = \cup_{s \in F} \{ w ; \delta (s_0 , w) = s \} \ ,$$

تشكل L اتحاد صفوف معينة وفق $\sim$ ، وبذلك ينتهي برهان الجزء الأول.

2) ليكن $\sim$ تطابقاً يمينياً بدليل منته ~~على~~ I^* ، وبفرض أنه يوجد $c_1 , c_2 , \dots , c_n \in$ $I^* / \sim$ ،

حيث $L = \cup_{j=1}^{n} c_j$.

نعرف الحاسبة $M' = (S' , I , \delta' , s_0' , F')$ كما يلي :

$$S' = I^* / \sim \quad , \quad s_0' = [\varepsilon] \quad , \quad F' = \{ c_1 , c_2 , \dots , c_n \} \quad ,$$

$$\delta' ([w] , i) = [w i] .$$

وللبرهان أن M' تشكل حاسبة يكفي أن نتحقق أن تعريف δ' صحيح .

إذا كان $[w_1] = [w_2]$ ، فإن $w_1 \sim w_2$. وبالتالي يكون $w_1 i \sim w_2 i$ ، من أجل كل $i \in I$.

تكون M' منتهية ، لأن $I^*/\sim$ تكون حسب الفرض مجموعة منتهية . وتكون اللغة L بذلك مميزة بالحاسبة M' ، لأن

$w \in L \Leftrightarrow w \in c_1 \vee \dots \vee w \in c_n \Leftrightarrow$

$[\, w \,] = c_1 \vee \dots \vee [w] = c_n \Leftrightarrow$

$\delta' (\, [\, \varepsilon \,] , w \,) = c_1 \vee \dots \vee \delta' (\, [\, \varepsilon \,] , w \,) = c_n \Leftrightarrow$

$\delta' (\, s_0' , w \,) \in F' .$ *

يمكن استخدام النظرية 2.4.4. لبرهان أن عدداً من اللغات منتظمة ، ويحتاج ذلك إلى إنشاء حاسبة مناسبة أو تعبير منتظم . ولذلك سنعرض مثالا في الفقرة التالية ،حيث سنبرهن أن الحاسبة المنتهية باتجاهين تتميز بلغات منتظمة .

أما الآن فسنتعرف على شكل هام لاستخدام النظرية 2.4.4، حيث نستطيع بطرية بسيطة وأنيقة البرهان أن لغة معينة غير منتظمة . أثناء استخدام هذه النظرية ، نستخدم غالبا الرمز w^n والذي يرمز لكلمة مكونة من الكلمة w مكررة n مرة متتالية . أي 0^n يرمز لكلمة مكونة من الصفر n مرة .

3.4.4. مثال .

برهن أن اللغة $L = \{\, 0^n \, 1^n \, ; \, n \geq 1 \,\}$ غير مميزة بحاسبة منتهية .

البرهان . نفرض مؤقتا أن اللغة L مميزة بحاسبة منتهية ، عندئذ يوجد تكافؤ يميني ~ على $\{\, 0 \, , 1 \,\}^n$ بدليل منته ، بحيث تشكل L اتحاد صفوف معينة من $\{\, 0 \, , 1 \,\}^n /\!\sim$.

لتكن $\{\, 0 \, , 1 \,\}^n /\!\sim$ تحوي n_0 صفا . عندئذ اثنين على الأقل من الكلمات $0^{n_0+1} \, , \, 0^{n_0+1} \, 1 \, , 0^{n_0+1} \, 1^2 , \dots , 0^{n_0+1} \, 1^{n_0}$ تقع على الصف نفسه.

إذن $0^{n_0+1}1^i \sim 0^{n_0+1}1^j$ بالنسبة لـ i , j ما ، حيث $0 \le i < j \le n_0$. وبالتالي يكون

$0^{n_0+1}1^i\ 1^{n_0-j+1} \sim 0^{n_0+1}1^j\ 1^{n_0-j+1} = 0^{n_0+1}\ 1^{n_0+1}$.

وبما أن $0^{n_0+1}\ 1^{n_0+1} \in L$ ، فيجب أن يكون $0^{n_0+1}\ 1^{n_0+i-j+1} \in L$ ، وهذا يناقض الفرض المؤقت .*

4.4.4. مثال.

برهن أن اللغة $L = \{ a^{n2}\ ;\ n \ge 0 \}$ غير منتظمة .

البرهان . نفرض مؤقتا أنه يوجد تطابق يميني مناسب $\sim$ يحوي n_0 صف تكافؤ . عندئذ اثنتين على الأقل من الكلمات $a^{n_0^2}, a^{n_0^2+1}, \ldots , a^{n_0^2+n_0}$ تقع حتما في صف واحد ، أي أن

$a^{n_0^2+i} \sim a^{n_0^2+j}$ ، من أجل $0 \le i < j \le n_0$ ما . إذن

$a^{n_0^2+i}\ .\ a^{2n_0-j+1} \sim a^{n_0^2+j}\ .\ a^{2n_0-j+1}$. ولكن أيضا

$a^{n_0^2+j}\ .\ a^{2n_0-j+1} = a^{n_0^2+2n_0+1} = a^{(n_0+1)2} \in L$ ، وبالتالي يكون

$n_0^2 < n_0^2 + i + 2n_0 - j + 1 < (n_0 + 1)^2$. ينتج أن $a^{n_0^2+i+2n_0-j+1} \notin L$. *

من الواضح ، أنه يمكن استخدام الخصائص الأخيرة للغات المنتظمة أثناء براهين كون بعض اللغات غير منتظمة .

5.4.4. مثال .

برهن أن اللغة $L = \{ w \in \{ 0 , 1 \}^* \}$ ، حيث تحوي الكلمات w العدد نفسه من الصفر ومن الواحد ، غير منتظمة .

البرهان . نفرض أن L لغة منتظمة . بما أن $L_1 = [0^* 1^*]$ منتظمة ، تكون أيضا حسب النظرية 1.2.4. اللغة $L \cap L_1$ منتظمة . ولكن $L \cap L_1 = \{ 0^n 1^n ; n \geq 0 \}$ ، وهذا يناقض التأكيد في المثال 3.4.4. وبالتالي ينتج أن L غير منتظمة .

6.4.4. تمارين.

1.4.4. برهن أن اللغات التالية غير منتظمة :

a) $\{ 0^m 1^m \; ; \; 0 \leq m \leq n \}$.

b) $\{ 0^p$; عدد أولي $p \}$

c) $\{ 0^m 1^m 0^m \; ; \; 0 \leq m , n \}$.

d) $\{ ww \; ; w \in \{ 0 , 1 \}^* \}$.

e) $\{ w (w)^R \; ; w \in \{ 0 , 1 \}^* \}$.

I . أبجدية اختيارية ; Rs (I)

2.4.4. يسمى التكافؤ ~ على I^* تطابقاً يسارياً إذا تحقق الشرط التالي :

$$(\forall u , v , w \in I^*) [u \sim v = w u \sim w v].$$

برهن أن النظرية 2.4.4. تكون صحيحة حتى عندما نبدل عبارة " تطابق يميني " بعبارة " تطابق يساري" .

3.4.4. يسمى التكافؤ $\sim$ على I^* تطابقاً ، إذا كان بالوقت نفسه يمثل تطابقاً يمينياً ويسارياً .

برهن أن النظرية 2.4.4. تكون محققة بالنسبة للتطابق أيضا.

4.4.4. يمكننا لكل لغة $L \subseteq I^*$ تعريف أثمن تطابق يميني $\sim$ بالشكل :

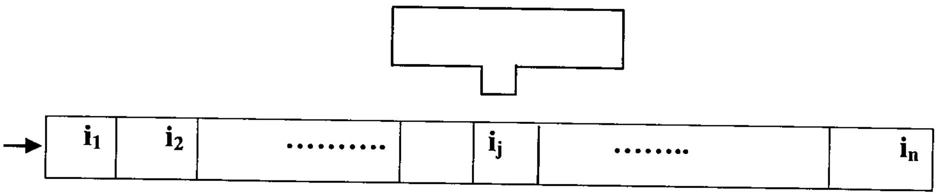
$$u \sim v \Leftrightarrow (\forall w \in I^*)\ [uv \in L \Leftrightarrow vw \in L].$$

برهن أن أثمن تطابق يميني للغة L تكون بدليل منته إذا وفقط إذا كانت L منتظمة .

B

5.4. الحاسبة المنتهية باتجاهين .

يمكن تصور الحاسبة المنتهية كوحدة قيادة منتهية متصلة بشريط دخل . والشريط مقسم إلى حقول ، كل منها يحوي رمزاً واحداً فقط من أبجدية الدخل . في كل مرة يصل إلى المدخل الرمز التالي من كلمة الدخل ، أي أن شريط الدخل يزاح باتجاه واحد بمعدل حقل واحد (أنظر الشكل 1.4).



(شكل 1.4)

إذا كان للحاسبة المنتهية القدرة على إزاحة الشريط في كلا الاتجاهين فتسمى حاسبة منتهية باتجاهين .

1.5.4. تعريف.

تعرف الحاسبة المنتهية باتجاهين بالخماسية التالية : $M = (S , I , \delta , s_0 , F)$ ،

حيث S مجموعة منتهية (فضاء الحالات) ، I مجموعة منتهية (أبجدية الدخل) ، $s_0 \in S$ الحالة الأولية ، $F \subseteq S$ مجموعة الحالات النهائية ، ودالة النقل

$$\delta : S \times I \to \{ -1 , 0 , +1 \} .$$

تعمل الحاسبة M على الكلمة $w \in I^*$ بالطريقة التالية :

إذا كان المطلوب كتابة الكلمة w ، تبدأ العمل في الحالة الأولى على الرمز الأيسر للكلمة w . إذا قرأت في لحظة ما في الحالة s الرمز i وكان $\delta(s, i) = (s', p)$ ، عندئذ تنتقل إلى الحالة s' وتزاح بمقدار P حقل إلى اليمين .

تكون الكلمة w مقبولة إذا وفقط إذا غادرت الحاسبة النهاية اليمنى للكلمة w في إحدى الحالات النهائية .

ترمز $L(M)$ لمجموعة الكلمات المقبولة بالحاسبة M ، أي اللغة المميزة بهذه الحاسبة.

نعرض من النتائج الهامة والمفيدة للحاسبة المنتهية بطرفين النظرية التالية ، التي تبرهن على أن تغيير شريط الدخل من اتجاه واحد إلى اتجاهين لا يوسع صف اللغات المميزة بحاسبة منتهية ، استنادا إلى النظرية 2.4.4.

2.5.4. نظرية.

لكل حاسبة منتهية باتجاهين M يوجد لغة منتظمة $L(M)$.

البرهان . نعرف لكل كلمة $w \in I^*$ الدالة $f_w : \{s_0^-\} \cup S \rightarrow S \cup \{0\}$ ، حيث $s_0^-, 0 \notin S$ ، كما يلي : إذا كانت الحاسبة M تبدأ العمل في الحالة $s \in S$ (s_0 ، على التوالي) برأس قراءة مركز على الطرف الأيمن (الأيسر ، على التوالي) للكلمة w ، وتغادر النهاية اليمنى في الحالة s' ، عندئذ نضع

$f_w(s) = s'$ ، ($f_w(s_0^-) = s'$ ، على التوالي).

في الأوضاع الأخرى جميعها (أي ، M تغادر الطرف الأيسر للكلمة w أو لا تغادرها مطلقا) نضع

$f_0(s) = 0$ ، ($f_w(s_0^-) = 0$ ، على التوالي).

نعرّف الآن على I^* العلاقة ~ كما يلي :

$$w_1 \sim w_2 \Leftrightarrow_{df} (\forall w \in I^*) : (w_1 w \in L(M) \Leftrightarrow$$
$$w_2 w \in L(M)). \qquad (*)$$

ويكفي أن نبرهن أن

$$f_{w1} = f_{w2} \Rightarrow w_1 \sim w_2 \qquad (**).$$

نترك برهان هذه العلاقة للقارئ (انظر التمرين 8.3.4).

بما أن مجموعة الحالات S منتهية ، تكون أيضا المجموعة $\{ f_w \; ; w \in I^* \}$ منتهية .

وإذا كان أيضا $card\ S = n$ ، عندئذ يكون $card \{ f_w \; ; w \in I^* \} \leq (n+1)^{n+1}$.

تكون العلاقة ~ إذن وفق (**) بدليل منته . وتكون L اتحاد صفوف معينة من $I^* / \sim$ ، لأن

$$w_1 \sim w_2 \Leftrightarrow (w_1 \varepsilon \in L(M) \Leftrightarrow w_2 \varepsilon \in L(M)).$$

وذلك وفقا لـ (*) . وبذلك يتم البرهان .

3.5.4. تمارين .

1.5.4. تحقق من صحة العلاقة (**) $f_{w1} = f_{w2} \Rightarrow w_1 \sim w_2$ ، بوساطة برهان النظرية 2.5.4 .

توجيه : نفرض أن $f_{w1} = f_{w2}$. من أجل w كلمة اختيارية ، نرمز بـ $s_{i1}, \ldots, s_{im}$ لمتوالية الحالات التي يعبر فيها رأس القراءة للحاسبة M خلال شغل الكلمة $w_1 w$)

بكلا الاتجاهين) الحدود بين الرمز الأخير من الكلمة w_1 والرمز الأول للكلمة w .

ولتكن $s_{j1}, \ldots, s_{jn}$ ترمز لمتوالية مشابهة للكلمة w_2w .

ونبرهن بالتدريج أن $s_{j1}, \ldots, s_{jn} = s_{i1}, \ldots, s_{im}$. بوساطة ذلك استنتج أن

$$(w_1 w \in L(M) \Leftrightarrow w_2 w \in L(M)).$$

4.5.2. حسب التعريف السابق للكلمات المقبولة ، لا تستطيع الحاسبة المنتهية باتجاهين التمييز متى تكون على نهاية الكلمة . لذا نوسع إمكانية الحاسبة ، حيث تكون نهايتا الشريط مرمزتين برمز النهاية # ، والذي لا يظهر في أبجدية الدخل .

نعرّف $T(M) = \{ w ; \# w \# \in L(M) \}$. استخدم ما تعرفه عن اللغات المنتظمة لبرهان أن $T(M)$ تكون دائما منتظمة .

4.5.3. نوسع التعريف 4.5.1. إلى تعريف " حاسبة باتجاهين بلصاقة واحدة " ، أي رمز يمكن وضعه على حقل شريط . تستطيع الحاسبة التمييز فيما إذا يقع على الحقل لصاقة . في حال نعم ، تستطيع رفع اللصاقة ووضعها في مكان آخر . إذن تستطيع أن ترمز لحقل معين ، ومغادرته وبعد زمن معين يمكن البحث ثانية عن الحقل وفق اللصاقة . من المهم أنه لا يمكن تحريك اللصاقة بعيدا ، ولكن يمكن العودة إليها دوما .

أ) برهن أن هذا النوع من الحاسبات يميز أيضا لغات منتظمة .

ب) برهن أن " الحاسبات المنتهية باتجاهين مع لصاقتين تميز لغات غير منتظمة .

6.4. المعادلات المنتظمة .

نضيف إلى الإمكانيات السابقة المعطاة باللغات المنتظمة إمكانية جديدة لتعيين اللغة بحل مجموعة معادلات منتظمة . نوضح ذلك بمثال بسيط.

1.6.4. مثال .

لتكن L لغة ، كل كلمة منها تتكون من عدد زوجي من الرمز 1 متبوعة بالسلسلة 01110 . نستطيع أن نعبر عن هذه الخاصية بالمعادلة

$$X = 11X + 01110 \qquad (*)$$

بشكل عام ، يكون حل هذه المعادلة تعبيراً منتظماً α ، حيث بعد تعويض X به ، نحصل على تعبير منتظم مكافئ $\alpha = 11\alpha + 01110$.

من السهل التحقق أن مثل هذا الحل يكون التعبير $X = (11)^* + 01110$.

وعموما ، تكون المعادلة $(**)$ $X = \alpha X + \beta$ ، حيث α , β تعابير منتظمة ، قابلة للحل .

سنهتم غالبا بالحل الأصغري ، أي التعبير الذي يشكل حلاً وبالوقت نفسه يصف اللغة الأصغرية من بين جميع حلول المعادلة (**) .

يعطى الحل الأصغري لـ (**) بشكل وحيد التعيين ويساوي $X = \alpha^* \beta$. تستعمل هذه النتيجة عند حل جملة المعادلات النظامية القياسية .

2.6.4. **تعريف**.

تعرف جملة المعادلات النظامية القياسية بالمتغيرات $\chi = \{ X_1 , X_2 , \ldots , X_n \}$ بالشكل التالي:

$$X_i = \alpha_{i0} + \alpha_{i1} X_1 + \alpha_{i2} X_2 + \ldots + \alpha_{in} X_n \quad , \quad 1 \leq i \leq n ,$$

حيث α_{ij} تعبير منتظم على أبجدية ما منفصلة عن χ .

يمكن حل جملة المعادلات المنتظمة القياسية بطريقة مشابه لحل جملة المعادلات الخطية (مثلا طريقة غاوص) . وخلال ذلك نستخدم معارفنا لحل المعادلة (**) . وتكون النتيجة ثانية الحل الأصغري للجملة المعطاة .

نوضح طريقة الحل بالمثال التالي:

3.6.4. **مثال** .

حل جملة المعادلات التالية :

(1) $X_1 = (01^* + 1) X_1 + X_2$

(2) $X_2 = 11 + 1 X_1 + 00 X_3$

(3) $X_3 = \varepsilon + X_1 + X_2$

من المعادلة (1) نجد أن $X_1 = (01^* + 1)^* X_2$.

نعوض في المعادلتين (2) ، (3) فنجد

(4) $X_2 = 11 + 1 (01^* + 1) X_2 + 00 X_3$

(5) $X_3 = \varepsilon + (01^* + 1)^* X_2 + X_2$

نصلح المعادلتين (4) ، (5) فنحصل على

$$(6) \quad X_2 = 1\,(0+1)^*\, X_2 + (11 + 00X_3)$$

$$(7) \quad X_3 = \varepsilon + (0+1)^* X_2$$

من (6) نحصل على مكافئ X_2 وهو $[\,1\,(0+1)^*\,]^*\,(11 + 00X_3)$.

نبسط هذا التعبير إلى الشكل $(0+1)^*\,(11 + 00X_3)$ ، ونبدل X_2 في المعادلة (7) فنجد :

$$X_3 = \varepsilon + (0+1)^*\,(0+1)^*\,(11 + 00\,X_3)\,.$$

بالإصلاح نجد :

$$X_3 = \varepsilon + (0+1)^*\,(11 + 00\,X_3) = \ \varepsilon + (0+1)^*\,11 + (0+1)^*\,00\,X_3).$$

ومنه نجد : $X_3 = [(0+1)^*\,00\,]^*(\varepsilon + (0+1)^*\,11)$

بالإصلاح نجد :

$$(8) \quad X_3 = [(0+1)^*\,00 + \varepsilon\,]\,(\ \varepsilon + (0+1)^*\,11\,) =$$

$$= (0+1)^* 00 + \varepsilon + (0+1)^* / 00\,(0+1)^*\,11 + (0+1)^* 11 =$$

$$= (0+1)^*[\,00 + 11\,] + \varepsilon\,.$$

نعوض في (6) ونصلح

$$X_2 = 1\,(0+1)^*\,X_2 + (11 + 00\,[(0+1)^*\,[\,00 + 11\,] + \varepsilon\,]) =$$

$$= \ 1\,(0+1)^*\,X_2 + (11 + 00 + 00\,(0+1)^*\,[\,00 + 11\,])\,.$$

ينتج أن

$$(9) \quad X_2 = [\,1\,(0+1)^*]^*\ [\,11 + 00 + 00\,(0+1)^*\,[\,00 + 11\,]\,] =$$

$$= \ (0+1)^*[\,11 + 00\,].$$

نعوض في (1) فنجد :

$$X_1 = (01^* + 1)X_1 + (0+1)^*[11 + 00].$$

ومنه ينتج

$$X_1 = (01^* + 1)(0+1)^*[11 + 00] = \tag{10}$$
$$= (0+1)^*[11 + 00].$$

ويكون حل الجملة هو التعابير (8) , (9) , (10) .

4.6.4. تمارين.

1.6.4. اكتب خوارزمية لحل جملة المعادلات المنتظمة القياسية بطريقة غاوص ، وبرهن أنها تعطي الحل الأصغري.

2.6.4. بين كيف يمكن إعطاء بجملة معادلات منتظمة قياسية لغة ممثلة :

1) بحاسبة منتهية معطاة .

2) بقواعد خطية يمينية معطاة (انظر الفصل الخامس).

7.4. متوالية ترتيب (تزامن) الأحداث.

يوجد للغات المنتظمة أهمية كبيرة في حل جملة من المشاكل المرتبطة بالحاسبات المنتهية . ولتوضيح ذلك نعرض إحدى المسائل التي يمكن نقلها بطريقة طبيعية إلى اختبار للغة معينة مميزة بحاسبة منتهية.

بعض الحاسبات تملك الخاصية التالية : يوجد متوالية دخل تنقل الحاسبة من حالة اختيارية إلى حالة معينة ثابتة. تسمى مثل هذه المتوالية متوالية ترتيب الأحداث ، وتلعب دورا هاما عند إنشاء بعض التجهيزات الإلكترونية . تمكننا من عبور أكبر عدد من التجهيزات المتطابقة الموجودة في حالات مختلفة (غير معروفة لنا) مرة واحدة إلى الحالة نفسها، أو العودة إلى الوظيفة العادية للتجهيزات. ولهذا تظهر الحاجة أحيانا للتأكيد، فيما إذا تملك حاسبة معطاة متوالية ترتيب (متوالية تقود إلى حالة معينة) وتعيين مثل هذه المتوالية.

يمكن صياغة مثل هذه المشكلة باستخدام مفاهيم نظرية الحوسبة كما يلي :

لتكن $M = (S , I , F , \delta , \mu)$ آلة معطاة . والمطلوب التحقق فيما إذا يوجد $w \in I^*$ وحالة ما $s_F \in S$ ، حيث يكون $\forall s \in S ,\ \delta (s , w) = s_F$.

يمكن في لحظة ما، أن تكون الآلة M في حالة اختيارية $s \in S$ ، وبعدئذ تنقلها متوالية الدخل w إلى حالة ما من المجموعة

$$S_w = \{ s ; (\exists s' \in S) [\delta (s' , w) = s] \} .$$

إذن توجد متوالية ترتيب الأحداث فقط عندما توجد w حيث يكون $card\ S_w = 1$. وتكون متوالية ترتيب الأحداث للآلة M مساوية للغة الممثلة بالحاسبة

$N = (S' , I , \delta' , s_0' , F')$ ، حيث

$$S' = P(S), \quad s_0' = S, \quad F' = \{ \{ s \} ; s \in S \},$$

$$\delta'(Q, i) = \{ s ; \ (\exists s' \in Q)[\delta(s', i) = s] \}, \quad \forall (Q \subseteq S, \quad i \in I).$$

إذا أردنا تعيين متوالية تسلسل الأحداث التي تنقل الآلة إلى واحدة من الحالات المعينة s_F، عندئذ تحوي المجموعة الهدفية F' للحاسبة N مجموعة جزئية وحيدة من فضاء الحالات S وهي $\{ s_F \}$.

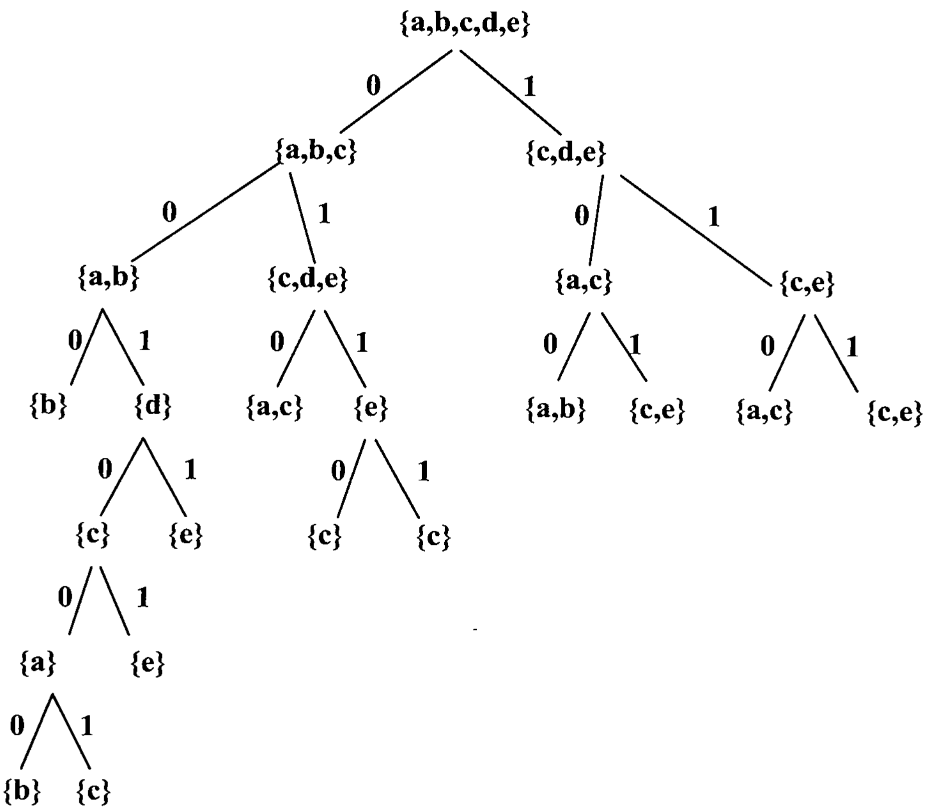
1.7.4. مثال.

عين متوالية تسلسل الأحداث للآلة M المعطاة بالجدول التالي :

	0	1	
a	b	c	0
b	b	d	1
c	a	e	0
d	c	e	0
e	c	c	1

لحل المسألة يكفي أن نضم إلى فضاء حالات الحاسبة N الممثلة بمتوالية تسلسل الأحداث فقط الحالات القابلة للبلوغ من الحالة الأولية a , b , c , d , e . ويتم وصف فضاء الحالات ودالة النقل للحاسبة N عندئذ بشجرة (انظر الشكل 2.4.) . ويكون للحاسبة ، التي تقبل جميع متواليات تسلسل الأحداث ، المجموعة الهدفية

$F' = \{\{a\},\{b\},\{c\},\{d\},\{e\}\}$. والحاسبة التي تقبل جميع متواليات تسلسل الأحداث التي تنقل إلى الحالة b تملك المجموعة الهدفية $F' = \{\{b\}\}$. الخ .من الواضح ، أن للآلة M عدداً لا نهائياً من متواليات تسلسل الأحداث التي تقود إلى حالة اختيارية معينة مسبقا . مثلا إلى الحالة b تقود المتواليات 0001000 , 0000 , 000، الخ.



(شكل 2.4)

2.7.4. تمارين.

1.7.4. لتكن $M_1 , \ldots , M_n$ حاسبات تعاقبية . هل تكون مجموعة جميع المتواليات ، التي تشكل متواليات تسلسل الأحداث لكل M_i ، لغات منتظمة ؟

2.7.4. لتكن $M = (S , I , F , \delta , \lambda)$ حاسبة تعاقبية ، $s_0 \in S$. نرمز بـ

$$\Delta (s_0 , L) = \{ \varphi_M (s_0 , w) ; w \in L \}$$

، حيث φ_M معرفة كما في 1.2.3.) .

برهن أنه لكل لغة منتظمة L تكون $\Delta (s_0 , L)$ منتظمة .

5. الفصل الخامس

القواعد الشكلية (Formal Grammars).

محتوى الفصل :

1.5. النظم الناسخة و القواعد .

2.5. هيكلية تشومسكي (Chomsky Hierarchy).

3.5. القواعد شكل 3 واللغات المنتظمة .

أهداف الفصل :

يهدف هذا الفصل إلى اكتساب المعارف التالية :

مفهوم النظم الناسخة والقواعد عليها .

هيكلية تشومسكي ، مكوناتها وخصائصها .

القواعد شكل 3 واللغات المنتظمة مع توضيح خصائصها بوساطة القواعد الشكلية .

1.5. النظم الناسخة و القواعد .

تعرفنا في الفصلين الثالث والرابع على طريقتين مهمتين لتوصيف اللغات المنتهية . تحدد الطريقة الأولى اللغة بتجهيزات معينة تميز الكلمات المناسبة للغة (الحاسبات) . وتبين الطريقة الثانية خصائص الكلمات التي تنتمي إلى اللغة (التعابير المنتظمة). وسنتعرف في هذا الفصل على شكل لا يقل أهمية لوصف اللغات الشكلية مع توضيح خصائصها بوساطة القواعد الشكلية . وقد اقترحت هذه الطريقة أصلا لفحص اللغات الطبيعية . وتلعب اليوم الدور الأول لدراسة اللغات البرمجية .

1.1.5. تعريف.

يعرّف **النظام الناسخ** (النظام الإنتاجي) بأنه ثنائية مرتبة $R = (V , P)$ ، حيث V أبجدية الدخل و P مجموعة منتهية من الشروط الناسخة (الإنتاجية). كل شرط ناسخ يشكل ثنائية مرتبة (u , v) ، حيث $u , v \in V^*$. ويكتب الشرط الناسخ (u , v) بالشكل $u \to v$.

2.1.5. تعريف .

ليكن $R = (V , P)$ نظاماً ناسخاً ، $w , z \in V^*$.

1) نقول إن w **تود مباشرة** z ، ونرمز له $w \Rightarrow_R z$ (أو $w \Rightarrow z$ ، في حالة عدم وجود التباس) ، إذا كان يوجد كلمات $x_1 , x_2 , u , v \in V^*$ ، حيث ، $z = x_1 v x_2$, $w = x_1 u x_2$ والشرط $u \to v$ ينتمي إلى المجموعة P .

2) نقول إن الكلمة w **تولد** (تنتج) z ، ونرمز لها $w \Rightarrow_R^* z$ (أو $w \Rightarrow^* z$) إذا كان يوجد متوالية كلمات

$$w_1 , w_2 , \ldots , w_n , \quad n \geq 1 \qquad (*)$$

من V^* ، حيث $w \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \ldots \Rightarrow w_n = z$.

تدعى المتوالية (*) **مشتقاً** بطول n للكلمة z من الكلمة w . ويكون المشتق (*) أصغرياً ، إذا كان $w_i \neq w_j$ ، من أجل جميع $i \neq j$.

3.1.5. مبرهنة .

إذا كان R نظاماً ناسخاً و $w \Rightarrow_R^* z$ ، عندئذ يوجد مشتق أصغري للكلمة z من الكلمة w .

البرهان . ينتج مباشرة من التعريف .

4.1.5. تعريف.

تعرّف **القواعد المولدة** بأنها رباعية مرتبة $G = (V_N , V_T , s , P)$ ، حيث V_N , V_T مجموعتين أبجديتين منفصلتين , $s \in V_N$, P مجموعة منتهية من الشروط الناسخة بالشكل $u \to v$ ،

حيث u , v كلمات من الأبجدية $V_N \cup V_T$ ، وأن u تحوي رمزاً واحداً على الأقل من V_N .

تمثل V_N مجموعة من المتغيرات (غير الطرفية ، Nonterminal) ، V_T مجموعة من الطرفيات (Terminals) ، s الرمز الأولي .

تعين القواعد المولدة G النظام الناسخ $(V_N \cup V_T , P)$. ويكون للتعابير " يولد (مباشرة) " ، " مشتق " ، الخ. بالنسبة إلى القواعد G ، وكذلك الرموز $\Rightarrow$ ، $\Rightarrow^*$ المعنى نفسه بالنسبة إلى النظام الناسخ .

تعرف اللغة $L (G)$ المولدة بالقواعد المولدة G بالشكل التالي :

$$L (G) = \{ w ; w \in V_T^* , s \Rightarrow^* w \},$$

وتتكون من جميع الكلمات في الأبجدية الطرفية ، التي يمكن اشتقاقها من الرمز الأولي .

5.1.5. مثال.

1). ليكن $G = (V_N , V_T , s , P)$ ، $V_N = \{ s \}$ ، $V_T = \{ 0 , 1 \}$ ،
$P = \{ s \to 0s1 , s \to 01 \}$.

من الواضح أن $L (G) = \{ 0^n 1^n ; n \geq 1 \}$.

2). لتكن $G = (V_N , V_T , s , P)$ ، $V_N = \{ s , B , C \}$ ، $V_T = \{ a , b , c \}$,
ولتكن P تحوي الشروط التالية : .

(1)	$s \to saBC$	(6)	$Ca \to aC$
(2)	$s \to aBC$	(7)	$BC \to BC$
(3)	$aB \to Ba$	(8)	$CB \to BC$
(4)	$Ba \to aB$	(9)	$B \to b$
(5)	$aC \to Ca$	(10)	$C \to c$

نلاحظ في هذه الحالة ، أن $L(G)$ تحوي تماما جميع الكلمات غير الخالية من V_T ، والتي تحوي العدد نفسه من الرموز a , b , c .

لنحاول مثلا اشتقاق الكلمة caabbcbac .

$s \Rightarrow_{(1)} saBC \Rightarrow_{(1)} saBCaBC \Rightarrow_{(2)} aBBaBCaBC \Rightarrow_{(7)}$

$aCBaBCaBC \Rightarrow_{(5)} CaBaBCaBC \Rightarrow_{(10)} caBaBCaBC \Rightarrow_{(4)}$

$caaBBCaBC \Rightarrow_{(9)} caabBCaBC \Rightarrow_{(3)}$

$caabbcBaC \Rightarrow_{(9)} caabbcbaC \Rightarrow_{(10)} caabbcbac$.

(1). لنبرهن أولا ، أن كل كلمة من $L(G)$ تحوي العدد نفسه من الرموز a , b , c .

ليكن $w \in L(G)$ ، عندئذ $s \Rightarrow^* w$ ، أي يوجد مشتق

$s \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \ldots \Rightarrow w_n = w$.

نلاحظ من شكل القواعد المكتوبة أعلاه ، أن جميع الكلمات في المشتق تملك الخواص التالية :

إذا كان $u(x)$ يرمز لعدد تكرارات الرمز x في الكلمة u ، عندئذ

$w_i(a) = w_i(b) + w_i(B) = w_i(c) + w_i(C) , \ 1 \leq i \leq n$.

وبما أن $w_n = w$ يحوي فقط طرفيات ، يكون $w_n(B) = w_n(C) = 0$.

وبالتالي $w(a) = w(b) = w(c)$.

2) لنبرهن الآن أن كل كلمة غير فارغة مشكلة من العدد نفسه من الرموز a , b , c تنتمي إلى $L(G)$.

إذا كانت $w \in V_T^*$ كلمة ما تحوي كل من الرموز a , b , c ، ن مرة ، مشكلة على الشكل التالي:

a) تستخدم الشرط (1) عدد من المرات يساوي (n-1) ، ثم تستخدم ال شرط (2) مرة واحدة فقط فتنتج الكلمة $(a B C)^n$.

b) بمساعدة الشروط (8) – (2) نجمع عدد الرموز في الكلمة $(a B C)^n$ ، والتي تتميز عن الكلمة w فقط بوجود B بدلا من b والرمز C بدلا من c .

c) بمساعدة القاعدتين (9) ، (10) نبدل جميع B بالرمز b وجميع C بالرمز c . فيتم بذلك البرهان.*

ينطلق تعريف القواعد المولدة التي استخدمناها من فكرة أن ملاءمة كلمة ما إلى لغة معطاة بشروط يمكن اشتقاقها من الرمز الأولي . ويمكن على العكس أن ننطلق من الكلمة وباستخدام الشروط التوصل إلى الرمز الأولي .

6.1.5. تعريف .

تعرّف **القواعد التحليلية** بأنها رباعية مرتبة $G = (V_N , V_T , s , P)$ ، حيث

V_N , V_T , s معرفة كما في تعريف القواعد المولدة و P مجموعة منتهية من الثنائيات المرتبة (u , v) (أو $u \to v$) حيث ،

$u , v \in (V_N , V_T)^*$ و v تحوي رمزاً واحداً على الأقل غير طرفي (متغير) .

تعين أيضا القواعد التحليلية نظاما ناسخاً $(V_N \cup V_T , P)$ ، يمكن بمساعدته إجراء العلاقات $\Rightarrow_G$ ، $\Rightarrow_G^*$.

وتعرف اللغة $L (G)$ المميزة بالقواعد التحليلية بالشكل :

$$L (G) = \{ w ; w \in V_T^* \ ; s \Rightarrow^* w \} .$$

يتضح من النظرة الأولى أن مفهومي القواعد المولدة والقواعد التحليلية متطابقة ، ونوضح ذلك بالتعريف التالي:

7.1.5. تعريف .

نقول أن القواعد G_1 , G_2 **متكافئة** ، سواء كانت مولدة أو تحليلية ، إذا وفقط إذا كان L $.L (G_1) = (G)$

وإذا كانت مثلا G_1 تحليلية و G_2 مولدة ، فإن G_1 تكافئ G_2 إذا كانت اللغة الممثلة بالقواعد G_1 تتولد بالقواعد G_2 .

8.1.5. نظرية .

لتكن $G = (V_N , V_T , s , P_1)$ قواعد مولدة (تحليلية ، على التوالي) . ولتكن P_2 مجموعة جميع الشروط $u \to v$ ، حيث $v \to u$ تنتمي إلى P_1 . عندئذ من أجل G $= (V_N , V_T , s , P_1)$ يكون $L (G_1) = L (G_2)$.

البرهان . بفرض أن G_1 قواعد مولدة ، عندئذ تكون G_2 قواعد تحليلية ، ولنبرهن صحة العلاقة :

$$\forall w \in (V_N \cup V_T)^* : \ s \Rightarrow_{G1}^* w \ \Leftrightarrow \ w \Rightarrow_{G2}^* s . \qquad (i)$$

9.1.5. تمارين.

1.1.5. أوجد مشتقاً ما للكلمة abbacccab انطلاقا من القاعدة s المبينة في الشروط المعطاة في المثال 5.1.5.(2) . هل يوجد مشتق أصغري وحيد لهذه الكلمة ؟

2.1.5. اقترح قواعد تولد ، تولد اللغات التالية:

a) $\{ 0^n 1^m 0^n \;\; ; \;\; n , m \geq 0 \}$.

b) $\{ a^n b^n c^n \;\; ; \;\; n \geq 0 \}$.

c) $\{ w (w)^R \;\; ; \;\; w \in \{ 0 , 1 \}^* \}$.

d) $\{ ww \;\; ; \;\; w \in \{ 0 , 1 \}^* \}$.

3.1.5. برهن أن القواعد $G = (\{ s , B , C \} , \{ a , b , c \} , s , P)$ ، حيث

P : $s \to asBC$ $\quad$ $s \to abC$

$CB \to BC$ $\quad$ $bB \to bb$

$bC \to bc$ $\quad$ $cC \to cc$

تولد اللغة $\{ a^n b^n c^n \;\; ; \;\; n \geq 1 \}$.

4.1.5. صف اللغة المولدة بالقواعد $G = (\{ s \} , \{ a , b \} , s , P \}$ ، حيث

$P = \{ s \to bss , s \to a \}$.

ويكفي لذلك أن نبرهن ،بالنسبة لـ n عدد اختياري، أن :

(ii) $w_0 \Rightarrow w_1 \Rightarrow \ldots \Rightarrow w_n$ مشتقة وفق G_1 إذا وفقط إذا

$w_n \Rightarrow \ldots \Rightarrow w_1 \Rightarrow w_0$ مشتقة وفق G_2.

من أجل $n = 1$ ، ينتج البرهان مباشرة من تعريف P_2 .

نفرض الآن العلاقة (ii) صحيحة من أجل $k \geq 1$ ، عندئذ يكون $w_k \Rightarrow_{G1} w_{k+1}$

إذا وفقط إذا كان $w_{k+1} \Rightarrow_{G2} w_k$ حسب الفرض .

بهذا يكون البرهان قد تم حتى من أجل $k+1$ وبالتالي تكون (ii) محققة.

ينتج من ذلك صحة العلاقة (i) ، وإذا حددنا $w \in V_T^*$ فيعني ذلك أن

$L (G_1) = L (G_2)$.

نفرض الآن العكس، أي أن G_1 قواعد تحليلية ، فتكون G_2 مولدة . و القواعد " المرافقة "

للقواعد G_2 والمنشأة وفق النظرية تكون G_1 .

وحسب الجزء الأول من البرهان نجد أن $L (G_1) = L (G_2)$.

تمكننا النظرية السابقة من تحديد الدراسة إما على القواعد المولدة أو القواعد التحليلية. ولهذا

سنتناول لاحقا القواعد المولدة ، وسنختصر القول على القواعد .

2.5. هيكلية تشومسكي (Chomsky Hierarchy).

يمكن توليد بعض اللغات بالقواعد ، التي تملك قوانين ناسخة بأشكال خاصة، أي لا تستخدم جميع الإمكانيات التي يتطلبها التعريف العام للقواعد (4.1.5). ويمكن أن نلتقي في المراجع مع العديد من الأشكال الخاصة للقواعد. وسنعرض في هذه الفقرة أشهر التصنيفات للقواعد واللغات ، التي تسمى هيكلية تشومسكي .

1.2.5. تعريف.

1) تدعى القواعد G المقدمة في التعريف 4.1.5. قواعد شكل (نمط) 0 . وتسمى اللغات المولدة بالقواعد شكل 0 لغات شكل 0 .

2) تدعى القواعد $G = (V_N , V_T , s , P)$ قواعد شكل 1 أو قواعد قرينة إذا وفقط إذا كانت جميع الشروط الناسخة في P بالشكل :

$$w_1 x w_2 \to w_1 y w_2 ,$$

حيث $w_1 , w_2 \in V_N \cup V_T$ ، $x \in V_N$ و y كلمة غير فارغة من $(V_N \cup V_T)^+$.

يمكن أن تكون الحالة الشاذة الوحيدة الشرط $s \to \varepsilon$ ، ووجودها يعني عدم إمكانية ظهور s على الجهة اليمنى لأية قاعدة ناسخة من P .

وتسمى أية لغة مولدة بقواعد قرينة لغة شكل 1 أو لغة قرينة .

3) تعرف القواعد شكل 2 أو القواعد غير القرينة بأنها قواعد من الشكل $G = (V_N , V_T , s , P)$ ، حيث p تحوي فقط شروطاً بالشكل

$$x \to w \; ; \; x \in V_N \; , \; w \in (V_N , V_T) .$$

تسمى اللغة لغة غير قرينة أو شكل 2 ، إذا وفقط إذا أمكن توليدها بقواعد غير قرينة ما .

4) تدعى القواعد $G = (V_N , V_T , s , P)$ قواعد شكل 3 أو قواعد منتظمة (أو خطية يمنى) ، إذا كان كل شرط من P بأحد الشكلين

$$X \to w \quad , \qquad X \to w\,Y ,$$

حيث $X , Y \in V_N \quad , \quad w \in V_T^{*}$.

تدعى اللغة لغة منتظمة أو لغة شكل 3 ، إذا أمكن توليدها بقواعد منتظمة .

أوردنا اسم " اللغة المنتظمة " في الفصل الثاني بالنسبة للغة الممثلة بتعبير منتظم . وسنبين حالا أننا في كلا الحالتين نتحدث عن الشعبة نفسها من اللغات .

كما نلاحظ أن القواعد الناسخة القرينة والقواعد غير القرينة تكون متشابهة الشكل . وأثناء التطبيقات في كلا الحالتين تنسخ القاعدة متحولا (غير طرفي) معين X في كلمة ما من $(V_N \cup V_T)^{*}$.

بالنسبة للقواعد شكل 1 ، يوجد إمكانية ربط المتحول الناسخ X بالقرينة التي يظهر فيها .
وهذه الإمكانية غير متوفرة بالنسبة للقواعد شكل 2 . ويمكن بالنسبة لهذه القواعد نسخ أيا من المتحولات الملائمة الناتجة ، بغض النظر عن القرينة التي يظهر فيها .
توضح هذه الملاحظة ، لماذا تدعى القواعد شكل 1 قرينة والقواعد شكل 2 غير قرينة .

2.2.5. تعريف.

نرمز لشعبة جميع اللغات شكل i (حيث : $i = 0 , 1 , 2 , 3$) بالرمز $£_i$.

3.2.5. تأكيد . $£_0 \supseteq £_1 \supseteq £_2 \supseteq £_3$

*) برهان صحة $£_0 \supseteq £_1$ ينتج مباشرة من التعريف 1.2.5.

**) برهان صحة $\pounds_2 \supseteq \pounds_1$ غير واضح من العريف ، لأن القواعد غير القرينة تستطيع مثلا ، احتواء عدد أكبر من القواعد من الشكل $Y \to \varepsilon$. أما القواعد القرينة فيمكن أن تحوي قاعدة وحيدة بكلمة فارغة على الجهة اليمنى . ولكن تنتج العلاقة $\pounds_2 \supseteq \pounds_1$ مباشرة من النظرية التالية . وهذه النظرية ستكون مفيدة فيما بعد لدراسة اللغات غير القرينة .

4.2.5. تعريف .

تدعى القواعد غير القرينة مقيدة ، إذا كانت لا تحوي أي قاعدة ناسخة بالشكل $Y \to \varepsilon$.

5.2.5. نظرية .

لكل قواعد غير قرينة G يوجد قواعد غير قرينة مقيدة G_1 حيث $L(G_1) = L(G) - \{\varepsilon\}$.

إذا كان $\varepsilon \in L(G)$ ، عندئذ يوجد قواعد غير قرينة $G' = (V_N', V_T', B', P')$ ، حيث لا تظهر في الجهة اليمنى لأي من الشروط في P' .

البرهان.

نشكل للقواعد $G = (V_N, V_T, B, P)$ قواعد $G_1 = (V_N, V_T, B, P_1)$ بالطريقة التالية :

نعرف $U_1 = \{X ; X \to w \in P\}$

$U_{i+1} = U_i \cup \{X ;\ X \to w \in P, w \in U_i^*\}$.

من الواضح أن $U_1 \subseteq U_2 \subseteq \ldots \subseteq V_N$. من هذا ينتج ، أنه يوجد عدد طبيعي n حيث يكون $U_n = U_{n+1}$.

من تعريف المتوالية $U_1 , U_2 ....$ ينتج ، أنه بالنسبة للعدد n يكون $U_n = U_{n+k}$ ،
حيث $k \geq 1$ عدد طبيعي اختياري . ويكون

$$X \Rightarrow_G^* \varepsilon , X \in U_n \qquad (*)$$

نشكل مجموعة الشروط P_1 كما يلي :

1) $w \neq \varepsilon$ ،

2) P تحوي الشرط $X \rightarrow y$ ، حيث $y = u_1Y_1u_2Y_2 \ldots u_mY_mu_{m+1}$ ،
عندما $Y_i \in U_n$ و $u_i \in [(V_N \cup V_T) - U_n]^*$.

وخلال ذلك يمكن الحصول على w بحذف بعض الرموز من $Y_1 , \ldots Y_m$. عندئذ تتحقق المساواة

$L (G_1) = L (G) - \{ \varepsilon \}$. ينتج الاحتواء $L (G_1) \subseteq L (G) - \{ \varepsilon \}$ مباشرة من (*) ومن تعريف P_1 .

ولنبرهن الآن العكس . $L (G_1) \supseteq L (G) - \{ \varepsilon \}$.

نعرف القواعد $G_2 = (V_N , V_T , B , P \cup p_1)$.

من المعلوم أن $L (G) \subseteq L (G_2)$. ومن أجل $\varepsilon \neq w \in L (G_2)$ يكون أيضا $w \in L (G_1)$.

لأن كل استخدام لشرط بالشكل $X \rightarrow \varepsilon$ ، يمكن تعويضه باستخدام مناسب لشروط من P_1 . وإذا كان أثناء الاشتقاق مستخدما الشرط $X \rightarrow \varepsilon$ ، فهذا يعني أنه يجب أن يكون قبل ذلك مستخدما شرطاً بالشكل

$Y \to u_1 \, x \, u_2$ و $u_1u_2 \in (V_N \cup V_T)^*$. وبدلا من هذا يمكن استخدام $Y \to u_1 \, u_2$ والتي تنتمي حتما إلى P_1 . وبهذا يتم برهان الجزء الأول من النظرية.

نفرض الآن أن $\varepsilon \in L(G)$. نشكل أولا القواعد G_1 كما في الجزء الأول من البرهان ، ثم نشكل القواعد

$$G' = (V_N \cup \{B'\}, V_N, B', P_1 \cup \{B' \to \varepsilon, B' \to B\}).$$

عندئذ يكون $L(G') = L(G_1) \cup \{\varepsilon\}$ ، ومجموعة الشروط تحقق القوانين المطلوبة .

6.2.5. تمارين .

1.2.5. أنشئ قواعد منتظمة مولدة للغة $01110(01)^*(111)^*$.

2.2.5. أنشئ قواعد غير قرينة مولدة للغات :

a) $\{ 0^n 1 0^n \quad ; \quad n \geq 0 \}$.

b) $\{ 0^n 1^m 0^n \; ; \quad n, m \geq 1 \}$.

c) $\{ ww^R \quad ; \quad w \in \{ 0, 1 \}^* \}$.

d) $\{ 0^n 1 0^m \; ; \quad 0 \leq n \leq m \}$.

e) $\{ 0^n 1^n \quad ; \quad 1 \leq n \leq m \leq 2n \}$.

f) $RV(\{0,1\}), (\{0,1\}$ مجموعة التعابير المنتظمة على الأبجدية) .

3.2.5. أنشئ قواعد قرينة مولدة للغات :

a) $\{ a^n b^n c^n \quad ; \quad n \geq 0 \}$.

b) $\{ ww \; , w \in \{ 0, 1 \}^* \}$.

4.2.5. أنشئ بالنسبة للحالات a) , b) , e) , f) في المثال 2.2.5. قواعد غير قرينة مقيدة .

3.5. القواعد شكل 3 واللغات المنتظمة .

بالعودة إلى اللغات شكل 3 ، نرى أنها لغات مميزة بحاسبات منتهية . وبالتالي فهي " لغات منتظمة " موصفة في التعريف 1.2.5.

1.3.5. نظرية .

لتكن L لغة مميزة بحاسبة منتهية ، عندئذ تكون L لغة بالشكل 3 .

البرهان . لتكن $L(M)$ لغة ممثلة لحاسبة منتية ما $M = (S, I, \delta, s_0, F)$.

نعرف القواعد $G = (S, I, s_0, P)$ بالشكل التالي :

$$P = \{ p \to x : \delta(p, x) = q \,;\ p, q \in S, x \in I \} \cup \{ p \to \varepsilon \,;\ p \in F \}.$$

ولنبرهن أن $L(M) = L(G)$. من الواضح أن

$$\varepsilon \in L(M) \Leftrightarrow q_0 \in F \Leftrightarrow q_0 \to \varepsilon \in P \Leftrightarrow \varepsilon \in L(G).$$

ينتج أن $\varepsilon \in L(M) \Leftrightarrow \varepsilon \in L(G)$. ولهذا يكفي معالجة الكلمات غير الفارغة .

لتكن $w = x_1 \dots x_n \in L(M)$ ، عندئذ يوجد متوالية حالات $q_0, q_1, \dots, q_n$ للحاسبة M حيث يكون

$$\delta(q_i, x_{i+1}) = q_{i+1} \,; \quad \forall (0 \le i \le n-1, \quad q_n \in F).$$

ينتج أن الشروط $q_i \to x_{i+1} q_{i+1}$ ، حيث $(0 \le i \le n-1)$ ، والشرط $q_n \to \varepsilon$ تنتمي إلى P .

إذا يوجد مشتق بالشكل $q_0 \Rightarrow x_1 q_1 \Rightarrow x_1 x_2 q_2 \Rightarrow \dots \Rightarrow x_1 \dots x_n q_n \Rightarrow$ $x_1 \dots x_n$ ، وبالتالي يكون $w \in L(G)$.

وبالعكس ، لتكن $w = x_1 \dots x_n \in L(G)$ ، عندئذ يوجد مشتق بالشكل

$$q_0 \Rightarrow x_1q_1 \Rightarrow x_1x_2q_2 \Rightarrow \dots \Rightarrow x_1 \dots x_n q_n \Rightarrow x_1 \dots x_n .$$

ينتج أن p تحوي الشروط

$q_i \rightarrow x_{n+1}\, q_{i+1}$ ، حيث $(p \le i \le n\text{-}1)$ و $q_n \rightarrow \varepsilon$.

يعني هذا أن $q_n \in F$, $\forall (0 \le i \le n\text{-}1)$; $\delta(q_i, x_{i+1}) = q_{i+1}$.

وبالتالي يكون $w \in L(M)$.*

2.3.5. مبرهنة .

لكل قواعد $G = (V_N, V_T, B, P)$ شكل 3 ، يوجد قواعد مكافئة

$G' = (V_N', V_T', B, P')$ أيضا من الشكل 3 ، و تحوي فقط شروطاً من الشكل

$$X \rightarrow aY \ , X \rightarrow \varepsilon : \ X, Y \in V_N' \ , a \in V_T .$$

البرهان . نشكل القواعد G' بالطريقة التالية :

(i) تكون V_T, B كما في القواعد G .

(ii) نشكل مجموعة القواعد P' كما يلي :

ننشئ مجموعة الشروط P'' ، حيث تحوي جميع الشروط من الشكل $X \rightarrow \varepsilon$, $X \rightarrow aY$ والمحتواة في P . وتكون جميع الشروط بالشكل

$$X \rightarrow x_1 \dots x_n Y \ ; \quad X, Y \in V_N \ , \quad x_1 \dots x_n \in V_T \ , \quad n \ge 2 .$$

نأخذ جملة الشروط $X \rightarrow x_1 Y_2$, $Y_2 \rightarrow x_2 Y_3$, ... , $Y_n \rightarrow x_n Y$ ،

حيث $Y_1, Y_2, \dots, Y_n$ رموز غير طرفية جديدة مضافة .

كل شرط بالشكل $Z \rightarrow z_1 \dots z_m$ ،

حيث $m \geq 1$, $z_1 \ldots z_m \in V_T^*$, $Z \in V_N$ ، تعوض بجملة الشروط

$Z \rightarrow z_1 Z_1$

$Z_1 \rightarrow z_2 Z_2$

....

$Z_{m-1} \rightarrow z_m Z_m$

$Z_m \rightarrow \varepsilon$,

حيث $Z_1 \ldots Z_m$ تكون من جديد رموزاً غير طرفية مضافة جديدة . وهكذا نحصل على مجموعة شروط بالشكل المطلوب . والمجموعة p تستطيع احتواء شروط بالشكل

$X \rightarrow Y$; $X, Y \in V_N$ (*)

نعالج هذه القوانين بالطريقة التالية :

ننشئ ، لكل متحول (غير طرفي) X يظهر على الجهة اليسرى للقاعدة بالشكل (*) ، المجموعة

$U(X) = \{ Y ; \exists\ X \Rightarrow X_1 \Rightarrow X_2 \Rightarrow \ldots \Rightarrow X_n = Y , X_1 , \ldots , X_n \in V_n , n \geq 0 \}$.

ويمكن تشكيل هذه المجموعة بشكل مكافئ . ننشئ أيضا بالتدريج المجموعات

$U_1(X) = \{ Y ; Y \in V_N \ \& \ X \Rightarrow Y \}$

....

$U_{i+1}(X) = U_i(X) \cup \{ Y ; Y \in N_N \ \& \ Z \Rightarrow Y , \forall z \in U_i(X) \}$.

بما أن

$$U_1(X) \subseteq U_2(X) \subseteq \ldots \subseteq U_i(X) \subseteq \ldots \subseteq V_N$$

فإنه يوجد n ، حيث يكون $U_n(X) = U_{n+1}(X)$.

إذا من الواضح أن $U(X) = U_n(X)$ ، وبالتالي يكون قد تم إنشاء $U(X)$.

والآن نضيف إلى المجموعة P' ، من أجل كل شرط من الشكل $Z \to zX$ ، جميع الشروط بالشكل $Z \to zY$ ، وذلك من أجل جميع $Y \in U(X)$. بذلك يتم تشكيل P' .

V_N' تحوي جميع المتحولات (غير الطرفية) الظاهرة في القواعد P' . والقواعد تكون من جديد بالشكل 3. وبحسب طريقة تشكيلها تكون مكافئة للقواعد G . *

3.3.5. نظرية .

كل لغة بالشكل 3 تكون مميزة بحاسبة منتهية .

البرهان . لتكن $G = (V_N, V_T, B, P)$ قواعد بالشكل 3 . بمساعدة المبرهنة 2.3.5 نستطيع أن نفرض بأن G تحوي شروطاً فقط بالشكل $X \to \varepsilon$, $X \to aY$.

ننشئ الآن حاسبة منتهية غير محدودة $M = (S, I, \delta, s_0, F)$ ، حيث

$$(*) \quad L(M) = L(G)$$

كما يلي :

$$I =_{df} V_T, \quad S =_{df} V_N, \quad s_0 = B, \quad F = \{X ; X \to \varepsilon \in P\},$$

$$\delta(X, a) = \{Y ; X \to aY \in P\}.$$

من الواضح أن

$$\varepsilon \in L(G) \Leftrightarrow B \to \varepsilon \in P \Leftrightarrow B \in F \Leftrightarrow \varepsilon \in L(M).$$

وبهذا تكون العلاقة (*) محققة بالنسبة للكلمة الفارغة .

لتكن $n \geq 1$, $x_1 \ldots x_n \in L(M)$. عندئذ يوجد مشتق

$$B \Rightarrow x_1 B_1 \Rightarrow x_1 x_2 B_2 \Rightarrow \ldots \Rightarrow x_1 \ldots x_{n-1} B_{n-1} \Rightarrow x_1 \ldots x_n B_n \Rightarrow x_1 \ldots x_n .$$

وهذا يعني أن P تحوي الشروط :

$$B \Rightarrow x_1 B_1 \ , \ B_1 \to x_2 B_2 \Rightarrow \ldots \Rightarrow B_{n-1} \to x_n B_n , \quad B_n \to \varepsilon .$$

وحسب تعريف الحاسبة M يكون

$$B_1 \in \delta(B, x_1) ; \quad B_{i+1} \in \delta(B_i, x_{i+1}) \ , \forall\, 1 \leq i \leq n ; \ B_n \in F .$$

ينتج أن $x_1 \ldots x_n \in L(M)$. وبالتالي $L(M) \subseteq L(G)$. *

4.3.5. تعريف .

تسمى القواعد $G = (V_N, V_T, B, P)$ قواعد خطية يسارية ، إذا كانت تحوي فقط شروطاً بالشكل

$$X \to w \ , \quad X \to Y w \ ; \qquad X, Y \in V_n \quad \& \quad w \in V_T^* .$$

5.3.5. نظرية .

تكون اللغة مولدة بقواعد خطية يسارية فقط إذا كانت منتظمة .

البرهان. لتكن $G = (V_N, V_T, B, P)$ قواعد خطية يسارية .

نعرّف القواعد $G' = (V_N, V_T, B, P')$ بالشكل

$$u \to v \in P' \quad \Leftrightarrow \quad u \to v^R \in P.$$

عندئذ تكون G' قواعد خطية يمينية ، أي قواعد بالشكل 3 . كما يتحقق

$$L = L(G) \Leftrightarrow L^R = L(G').$$

إذا تتولد اللغة L بقواعد خطية يسارية فقط عندما تكون L^R بالشكل 3 ، أي عندما تكون L منتظمة .*

وهكذا استطعنا توصيف مجموعة اللغات المنتظمة كمجموعة لغات ، يمكن توليدها بقواعد تحقق الخاصتين التاليتين :

a) يظهر خلال مشتقة اختيارية في كل سلسلة متحول (غير طرفي) واحد على الأكثر .

b) يظهر المتحول (غير الطرفي) دائما على الطرف الأيمن أو الأيسر للسلسلة .

6.3.5. تعريف .

تسمى القواعد $G = (V_N, V_T, B, P)$ قواعد خطية ، إذا كانت تحوي فقط شروطاً بالشكل

$$X \rightarrow u Y v, \quad X \rightarrow u; \qquad X, Y \in V_n \;\&\; u, v \in V_T^*.$$

وتدعى اللغة لغة خطية ، إذا أمكن توليدها بقواعد خطية ما .

7.3.5. نظرية .

صف اللغات المنتظمة يشكل مجموعة جزئية خاصة من اللغات الخطية .

البرهان . تشكل $£_3$ مجموعة جزئية من اللغات الخطية حسب 1.3.5. أو 5.3.5. ومع ذلك ، لا تكون اللغة

$L = \{ 0^n 1^n \;;\; n \geq 1 \}$ وفق 3.4.4 منتظمة . بينما تكون لغة مولدة بقواعد خطية بالشروط

$B \rightarrow 0 1 \;,\; B \rightarrow 0 B 1$ ، حيث B تكون غير طرفية و 0 ، 1 طرفية .

8.3.5. تمارين .

1.3.5. أنشئ القواعد المنتظمة المولدة للغات :

a) $((01+101)^*+(1+00)^*)^* 01^*0$.

b) $((a+bc)(aa^*+(ab)^*c+a)^*$.

c) $[(0^*10+((01)^*100)^*+0)^*(101(10010)^*+(01)^*1(001)^*)^*]^*$.

2.3.5. اقترح خوارزمية لنقل تعابير منتظمة إلى قواعد منتظمة مساوية لها .

(توجيه : بيِّن كيف يمكن من القواعد المنتظمة للغات المولدة L_1 , L_2 إنشاء قواعد منتظمة للغات $L_1 \cup L_2$, $L_1 . L_2$, L_1) .

3.3.5. برهن أن اللغة $\{ ww \; ; w \in \{0,1\}^* \}$ غير قابلة للتوليد بأي قواعد خطية يسارية .

4.3.5. برهن أنه يوجد قواعد $G = (V_N , V_T , B , P)$ تحوي فقط شروطاً بالشكل

$X \to Y w \; , X \to w Y \; , \; X \to w \; ; X , Y \in V_N , w \in {V_T}^*$,

وتولد لغة غير منتظمة .

(توجيه : اختبر اللغة $L = \{ 0^n 1^n \; ; \; n \geq 1 \}$).

5.3.5. برهن أن القواعد التي تحوي فقط شروط بالشكل ($X \to xY$, $X \to \varepsilon$) ، حيث X , Y غير طرفية و x طرفية ، تولد جميع اللغات المنتظمة .\

الفصل السادس
نظرية اللغات غير القرينة
Noncontextual langueges theory

محتوى الفصل:

1.6. شجرات الاشتقاق ، المشتقة من اليسار ، القواعد المختزلة .

2.6. نموذج تشومسكي الطبيعي "uvwxy" .

3.6. نموذج كريباخوف الطبيعي.

4.6. الحاسبة المكدس (المخزن) .

الهدف من الفصل :

يهدف هذا الفصل إلى اكتساب المعارف التالية :

1. مفهوم شجرات الاشتقاق ، الشجرة المرتبة ، الشجرة المقيمة ، المسرب ، مفهوم المشتقة من اليسار ، مفهوم القواعد المختزلة .
2. مفهوم نموذج تشومسكي الطبيعي ، طرق تشكيل قواعد ولغات باستخدام هذا النموذج .
3. مفهوم نموذج كريباخوف الطبيعي ، طرق تشكيل قواعد ولغات باستخدام هذا النموذج .
4. مفهوم الحاسبة المكدس ، وضع الحاسبة الآني ، طرق تشكيل قواعد ولغات باستخدام نموذج الحاسبة المكدس .

سنولي في هذا الفصل اللغات غير القرينة (Noncontextual langueges theory) اهتماما كبيرا. و ينبع هذا الاهتمام من حقيقة أن قواعد اللغة (syntax) لغالبية اللغات البرمجية (ابتداء من لغة algool 60) يمكن تعريفها بقواعد غير قرينة (noncontextual Grammers) وبالتالي يمكننا استخدام معلوماتنا حول اللغات غير القرينة ، مثلا أثناء تحليل القواعد .

سنكتب للسهولة القواعد التي لها الجهة اليسرى نفسها في سطر واحد ، وسنفصل بين الشروط في الطرف الأيمن بخط عمودي . مثلا ، سنكتب الشروط :

$$X \to w_1$$

$$X \to w_2$$

$$\dots\dots$$

$$X \to w_n$$

بالشكل :

$$X \to w_1 \mid w_2 \mid \dots \mid w_n$$

وسنعطي صف القواعد فقط بمجموعة الشروط . كما سنرمز لغير الطرفيات بأحرف كبيرة وبقية الرموز تكون طرفية . وسنرمز بـ B للرمز الابتدائي .

1.6. شجرات الاشتقاق ، المشتقة من اليسار ، القواعد المختزلة .

لتوضيح الاشتقاق وفق القواعد غير القرينة ، تستخدم غالبا شجرات الاشتقاق التي سنتعرف عليها فيما يلي :

1.1.6. **تعريف** .

لتكن (V , E) ثنائية ، حيث V مجموعة منتهية غير فارغة (مجموعة الرؤوس) و E $\subseteq V \times V$ مجموعة الأضلاع . نقول إن الضلع $(v_1 , v_2) \in E$ يخرج من الرأس v_1 ويدخل إلى الرأس v_2 .

تدعى متوالية الأضلاع $(v_1 , v_2) , \; (v_2 , v_3) \; , \ldots , (v_{m-1} , v_m)$ مسرباً من v_1 إلى v_m .

وتدعى الثنائية $T = (V , E)$ شجرة إذا كان :

(i) يوجد رأس وحيد لا يدخل إلية أي ضلع ، يسمى مثل هذا الرأس جذر الشجرة .

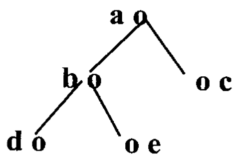
(ii) لكل رأس مختلف عن الجذر v يوجد مسرب واحد فقط من الجذر إلى v . وتسمى الرؤوس التي لا يخرج منها أي ضلع أوراق الشجرة ، أما بقية الرؤوس فتدعى رؤوس داخلية .

مثال .

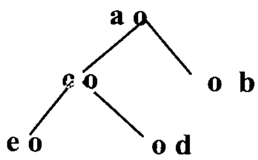
تشكل الثنائية (V , E) ، حيث

$$E = \{ (a , b) , (a , c) , (c , d) , (c , e) \} \text{ ، } V = \{ a , b , c , d , e \}$$

شجرة يمكن تمثيلها بالشكل 6 . 1 .(a) أو بالشكل 6 . 1 . (b) . ولكن شجرات الاشتقاق لن تكون اختيارية وسيتعين عليها ترتيب محدد للرؤوس .



(a)



(b)

(شكل 1.6)

2.1.6. تعريف.

تسمى شجرة مرتبة كل ثلاثية $T = (V , E , <)$ تحقق الشروط :

1) (V , E) تشكل شجرة .

2) الرمز $<$ يشكل ترتيب خطي للمجموعة V .

3) $< \subseteq E$. أي لكل $v_1 , v_2 \in V$ ، يتحقق الشرط $v_1 < v_2 \Rightarrow (v_1 , v_2) \in E$.

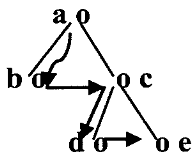
مثال .

لتكن $T = (V , E , <)$ شجرة مرتبة ، حيث

$$V = \{ a, b, c, d, e\} , \quad E = \{ (a , b), (a , c), (c , e), (c , d) \} ;$$

$$a < b < c < d < e \ .$$

نمثل هذه الشجرة بالشكل 2.6.



(شكل 2.6)

هذا التمثيل يحافظ على الشرط : من أجل كل رأس ، يكون الرأسان التاليان له المرسومان إلى اليسار وإلى اليمين متوافقين مع الترتيب $<$.

3.1.6. تعريف.

لتكن $T' = (V, E, <)$ شجرة مرتبة ، و X مجموعة منتهية و $f : V \to X$. عندئذ تدعى الثنائية

$T = (T', f)$ شجرة مقيمة ، حيث تقيم الدالة f الشجرة T بعناصر من X . كما نقول إن T مقيمة بعناصر من X ، وإن كل رأس v من الشجرة T مقيم بالعنصر $f(v)$.

4.1.6. تعريف.

لتكن $G = (V_N, V_T, B, P)$ قواعد غير قرينة ، ولتكن $T = (V, E, <, [\,])$ شجرة مرتبة مقيمة بعناصر من المجموعة $V_N \cup V_T \cup \varepsilon$. نرمز بـ $[v]$ لقيمة كل رأس v من الشجرة T.

نقول إن T تصف بنية السلسلة $\alpha \in (V_N \cup V_T)^*$ وفق القواعد G ، إذا كانت تحقق الشرطين التاليين :

1) إذا كانت $v_1 < v_2 < \ldots < v_n$ جميع أوراق الشجرة T، عندئذ يكون $[v_1] . [v_2] . \ldots . [v_n] = \alpha$.

2) ليكن v رأساً داخلياً اختيارياً للشجرة T و $u_1 < u_2 < \ldots < u_m$ جميع الرؤوس التي يدخل إليها أضلاع من الرأس v ، عندئذ يكون $[\, v\,] \to [\, u_1\,] . [\, u_2\,] . \ldots . [\, u_m\,] \in P$.

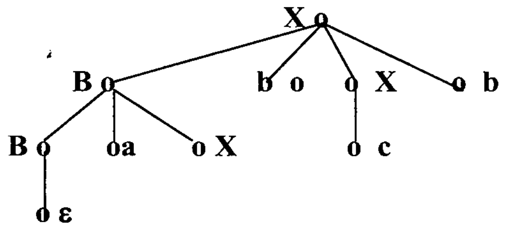
إذا كانت $m > 1$ ، عندئذ يكون $[\, u_i\,] \neq \varepsilon$ بالنسبة لكل $1 \leq i \leq m$.

5.1.6. مثال .

تصف الشجرة في الشكل 3.6 بنية السلسلة aXbcb وفق القواعد

$$B \to BaX \mid \varepsilon$$

$$X \to BbXb \mid c$$



(شكل 3.6)

6.1.6. مبرهنة .

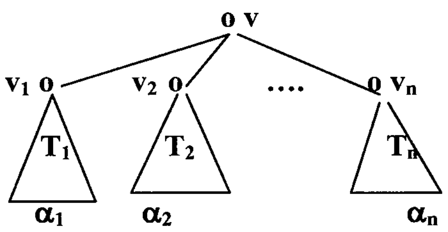
لتكن $G = (V_N , V_T , B , P)$ قواعد غير قرينة ، ولتكن $\alpha \in (V_N \cup V_T)^*$. إذا كانت توجد شجرة T تصف بنية السلسلة α ، عندئذٍ $X \Rightarrow^* \alpha$ حيث X رمز يمثل جذر الشجرة .

البرهان . لنبرهن بالتدرج وفق ارتفاع الشجرة T ، أي حسب طول المسرب الأطول في الشجرة .

1. لتكن T بطول يساوي 0 ، عندئذ تحوي T رأساً وحيداً v ، يكون بالوقت نفسه جذراً وورقة في الشجرة . من الواضح عندئذ أن $X = [\ v\ ] = \alpha$ ، وحسب تعريف $\Rightarrow^*$ يكون $[\ v\ ] \Rightarrow^* [\ v\ ]$.

2. ليكن $k \geq 0$ ولتكن المبرهنة محققة بالنسبة لكل شجرة ملائمة بارتفاع k على الأكثر. لتكن $T\ ,\ \alpha\ ,\ G$ كما في فرضيات المبرهنة ، ولتكن T بارتفاع $k+1$. عندئذ يمكننا تمثيل T كما في الشكل 4.6 .

كل من الشجرات $T_1\ ,\ T_2\ ,\ \ldots\ ,\ T_n$ تكون بارتفاع k على الأكثر ، كذلك $[\ v_1\ ]\ .\ \ldots\ .[\ v_n\ ]$ $[\ v\ ] \rightarrow$ يكون شرطا من القواعد G ، و $\alpha_1\ .\ \alpha_2\ \ldots\ \alpha_n = \alpha$. وحسب الفرض يكون $[\ v_i\ ] \Rightarrow^* \alpha_i$ بالنسبة لجميع $1 \leq i \leq n$. لأن



(شكل 4.6)

$$X = [\ v\ ] \Rightarrow [\ v_1\ ]\ .\ \ldots\ .[\ v_n\ ] \Rightarrow^* \alpha_1\ [\ v_1\ ]\ .\ \ldots\ .[\ v_n] \Rightarrow^* \alpha_1\alpha_2\ \ldots\ [\ v_n]$$

$$\Rightarrow^* \alpha_1\ \alpha_2\ \ldots\ \alpha_n = \alpha\ .\ *$$

7.1.6. مبرهنة .

لتكن $G = (V_N , V_T , B , P)$ قواعد غير قرينة ، ولتكن

$X \in V_N \quad , \alpha \in (V_N \cup V_T)^*,$

$\alpha \Rightarrow^* X$. عندئذ يوجد شجرة تصنيف بنية α وفق G ، جذرها مقيم بـ X .

البرهان .

نبرهن حسب طول المشتق :

$$(*) \qquad X = \beta_1 \Rightarrow \beta_2 \Rightarrow \ldots \Rightarrow \beta_m = \alpha$$

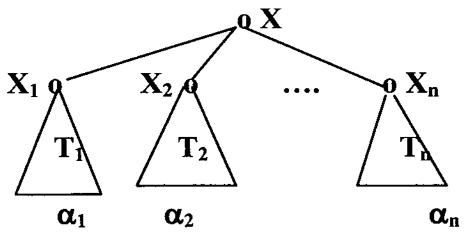
1) بالنسبة للمشتق بطول 0 تكون المبرهنة واضحة ويكون للشجرة رأس وحيد مقيم بالرمز $X \rightarrow \alpha$.

2) نفرض أن المبرهنة محققة بالنسبة للمشتق $X \Rightarrow^* \alpha$ بطول k على الأكثر . إذا كانت G , X , α كما في فرضيات المبرهنة والمشتق $(*)$ بطول $k+1$ ، عندئذ يكون :

$\beta_1 = X_1 \ldots X_n$ ، من أجل $X_i \in (V_N \cup V_T)$ معينة ،

$X \rightarrow X_1 \ldots X_n$ ،

و α يمكن كتابتها بالشكل $\alpha = \alpha_1 \ldots \alpha_n$ ، حيث $X_i \Rightarrow^* \alpha_i$ ، بالنسبة لجميع $1 \leq i \leq n$ ، حيث يوجد مشتق لـ α_i من X_i بطول k على الأكثر .



(شكل 5.6)

حسب الفرض ، يوجد لكل α_i شجرة T_i تصف بنية α_i وفق G ، جذرها مقيم بـ X_i ، عندئذ الشجرة T_i تصف بنية α_i وفق G ، جذرها مقيم بـ X_i . عندئذ الشجرة شكل 5.6 تصف البنية $\alpha = \alpha_1 \ldots \alpha_n$ وفق G . *

يوجد أهمية خاصة للأشجار التي تصف بنية السلاسل الطرفية ، والتي تكون جذورها مقيمة بالرمز الأولي للقواعد .

8.1.6. تعريف .

لتكن $G = (V_N , V_T , B , P)$ قواعد غير قرينة ، و $w \in V_T^*$, ولتكن T شجرة تصف بنية w وفق G . ولها جذر مقيم بالرمز B . عندئذ تسمى T شجرة اشتقاق w وفق G . وتدعى أحيانا مثل هذه الشجرة T تحليل w وفق G .

استنادا إلى المبرهنتين السابقتين تتحقق النتيجة التالية :

يوجد للسلسلة w شجرة اشتقاق وفق G إذا وفقط إذا $w \in L (G)$.

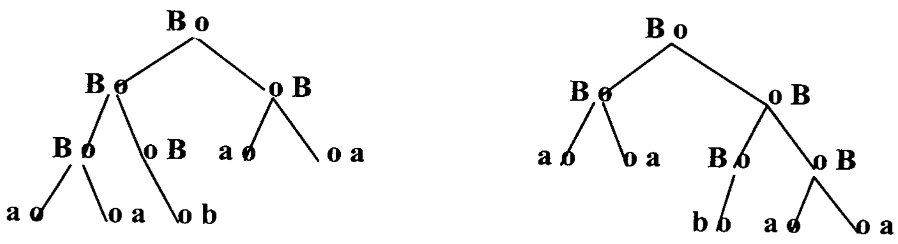
كل اشتقاق لـ w من قوانين G تقابل شجرة مشتقة وعلى العكس .

9.1.6. مثال .

لتكن G قواعد معطية بالشكل : $B \to BB \mid aa \mid b$.

إن المشتق $B \Rightarrow BB \Rightarrow BBB \Rightarrow aaBB \Rightarrow aabB \Rightarrow aabaa$

يقابل كلا الشجرتين بالشكل 6.6.



(شكل 6.6)

يمكن الرمز للشجرة اليسرى بالمشتق :

$$B \rightarrow BB \Rightarrow Baa \Rightarrow BBaa \Rightarrow aaBaa \Rightarrow aabaa .$$

يمكن للقارئ أن يجد بنفسه مشتقاً آخر يرمز للشجرة الثانية .

من المثال السابق ، يتضح أنه بين مشتقات سلاسل الطرفيات والأشجار المشتقة لا يوجد تقابل وحيد التعيين . ولكن سنقوي مثل هذا التقابل ، إذا استخدمنا أحد الأشكال القانونية للمشتق ، الذي سنتعرف عليه فيما يلي :

11.1.6 . تعريف.

تدعى القواعد غير القرينة G كثيرة التعيين ، إذا وجد كلمة من L (G) ، لها مشتقان يساريان مختلفان وفق القواعد G . وفي الحالات المختلفة تدعى وحيدة التعيين .

تدعى اللغة غير القرينة وحيدة التعيين ، إذا وجد قواعد وحيدة التعيين G ، حيث L = L (G) .

في الحالة المعاكسة تكون L كثيرة التعيين .

مثلا ، القواعد في المثال 9.1.6 تكون كثيرة التعيين ، لأنه يوجد بالنسبة للكلمة aabaa مشتقان يساريان مختلفان. واللغة التي تولد هذه القواعد أساسا ليست وحيدة التعيين، لأنه يمكن توليدها مثلا بالقواعد وحيدة التعيين التالية :

$$B \rightarrow aaB \mid bB \mid aa \mid b .$$

والبرهان بأن لغة ما كثيرة التعيين ، يعني البرهان بأن أيا من القواعد التي تولدها لا تكون وحيدة التعيين . وهذه البراهين تكون معقدة جدا .

كمثال للغة أمكن البرهان على أنها كثيرة التعيين نقدم اللغة

$$\{ a^i b^j c^k \ ; \ i = j \ \text{ or } \ j = k \} .$$

كل قواعد مولدة لهذه اللغة يجب أن تولد كلمات ، يكون من أجلها $i = j$ ، بطريقة تتميز عن مشتق الكلمات التي تحقق الخاصية $j = k$.

يوجد دوما طريقان للاشتقاق أثناء وصف اللغات غير القرينة ، ويكون من الأفضل العمل فقط مع ما يسمى القواعد المختزلة .

12.1.6. **تعريف**.

تدعى القواعد غير القرينة $G = (V_N , V_T , B , P)$ ، حيث $L (G) \neq \phi$ ، قواعد مختزلة إذا كانت تحقق الشرطين التاليين :

1) لكل متحول غير طرفي X ، يوجد على الأقل كلمة طرفية واحدة w ، حيث $X \Rightarrow_G^* w$.

2) لكل متحول $X \neq B$ ، يوجد كلمتان $u , v \in (V_N \cup V_T)^*$ ، حيث $X \Rightarrow_G^* u X v$.

إذاً القواعد المختزلة لا تحوي متغيرات عديمة الفائدة ، أي متغيرات لا يمكن ظهورها خلال أي مشتق لسلسلة الرموز الطرفية من الرمز B .

13.1.6. نظرية .

لكل قواعد غير قرينة $G = (V_N , V_T , B , P)$ ، حيث $L (G) \neq \phi$ ، يمكن بناء قواعد مختزلة مكافئة .

14.1.6. مبرهنة.

يوجد خوارزمية نستطيع بموجبها التحقق بالنسبة لقواعد غير قرينة اختيارية $G = (V_N , V_T , B , P)$ ، فيما إذا $L (G) \neq \phi$.

البرهان . نستخدم الطريقة التالية :

$$V_0 = V_T ,$$

$$V_{i+1} = V_i \cup \{ X \in V_N ; \exists w \in V_i^* : X \Rightarrow w \} .$$

من الواضح أن $V_i \subseteq V_N \cup V_T$ ، & $V_i \subseteq V_{i+1}$, $\forall i \geq 0$

وبالتالي يوجد n ، حيث $V_n = V_{n+k}$ من أجل $k \geq 1$ اختيارية . ويتحقق بذلك

$$L (G) = \phi \Leftrightarrow B \notin V_N .$$

وبما أن $V_n = V_{n+1}$ ، فيمكن فعلا برهان صحة المبرهنة.*

برهان النظرية 13.1.6.

1) ننشئ القواعد $G' = (V_N' , V_T , B , P')$ المكافئة للقواعد G والتي تحقق الشرط (1) من تعريف القواعد المختزلة ، ويتم ذلك كما يلي :

لكل $A \in V_N$ ننشئ قواعد $G_A = (V_N , V_T , A , P)$. إذا كانت $L (G_A) = \phi$ ، نحذف A من مجموعة المتحولات ، ومن مجموعة الشروط نحذف جميع الشروط التي تظهر فيها A على الطرف الأيمن أو الأيسر . بهذه الطريقة نحصل على القواعد G' .

من الواضح أن $L (G') = L (G)$. لأن كل مشتق وفق G' يكون أيضا مشتق وفق G . نفرض الآن وجود $w \in L (G) - L (G')$ ، عندئذ يجب أن يحوي مشتق الكلمة w وفق G كلمة بالشكل uAv ، حيث $A \in V_N - V_N'$. إذن يكون $B \Rightarrow_G^* uAv$ $\Rightarrow_G^* w$. ولكن $A \Rightarrow^* w_1$ بالسبة لسلسة طرفيات w_1 ، وهذا يناقض الفرض بأن $A \in V_N - V_N'$.

2) ننتقل الآن من القواعد G' إلى القواعد $G'' = (V_N'' , V_T , B , P'')$ ، حيث نحذف جميع الرموز غير الطرفية ، التي لا يمكن ظهورها في أي مشتق .

نشكل أولاً مجموعة جميع الرموز غير الطرفية من V_N' ، حيث يكون $B \Rightarrow^* \alpha_1 A \alpha 2$ من أجل $\alpha_1 , \alpha 2$ اختيارية .نرمز لهذه المجموعة بـ V_N'' ونشكلها كما يلي : نضع في المجموعة V_N'' الرمز B ، وبالتدريج نضيف جميع الرموز غير الطرفية التي تظهر في الجهة اليمنى من الشروط $A \to w$ ، حيث A غير طرفي، وقد تحققنا من انتمائه إلى V_N'' . ينتهي بناء V_N'' عندما لا نستطيع إضافة أي متحول (ويتم ذلك بشكل مشابه لبرهان المبرهنة 14.1.6).

مجموعة الشروط P'' ، تتعين من P' بحذف جميع الشروط منها والتي لا يظهر على طرفها الأيمن أو الأيسر أحد الرموز من $V_N' - V_N''$.

بشكل مشابه للجزء الأول من البرهان ، نبرهن أن $L (G) = L (G'')$.*

15.1.6. تمارين.

1.1.6. أنشئ الأشجار التي تصف بنية السلاسل التالية وفق القواعد التالية ، حيث B ترمز للرمز الأولي :

$B \to B + T \mid T$

$T \to T + F \mid F$

$F \to (B) \mid a .$

1) $a + a * a .$

2) $a * (a + a + a) * a .$

3) $(B) * a * a .$

4) $a * (a + a + a) + (a) .$

5) $a * (* a) .$

2.1.6. أنشئ لكل شجرة مشتقة مشكلة في التمرين 1.1.6 ، المشتق اليساري والمشتق اليميني .

3.1.6. برهن أن كل لغة منتظمة تكون وحيدة التعيين .

4.1.6. أنشئ القواعد المختزلة للقواعد التالية ، حيث A الرمز الأولي :

$A \to a A b \mid a A C \mid a \mid b B a .$

$B \to b C a \mid A C .$

$C \to b C B \mid A B .$

2.6. نموذج تشومسكي الطبيعي "uvwxy" .

1.2.6. تعريف .

تكون القواعد غير القرينة بصيغة نموذج تشومسكي الطبيعي إذا وفقط إذا كانت جميع شروطها بالشكل

$X \to a$ أو $X \to YZ$ ، حيث X , Y , Z رموز غير طرفية و a رمز طرفي .

نرى من التعريف ، أنه لا يمكن للغة المولدة بقواعد بصيغة نموذج تشومسكي الطبيعي أن تحوي كلمة فارغة ومع ذلك سنبرهن بأن كل لغة غير قرينة يمكن توليدها بقواعد ما بصيغة نموذج تشومسكي الطبيعي .

2.2.6. مبرهنة .

لكل قواعد غير قرينة G يمكن إنشاء قواعد مكافئة غير قرينة G_1 لا تحوي أية قاعدة بالشكل $A \to B$ ، حيث A , B رموز غير طرفية .

البرهان . ليكن $G = (V_N , V_T , S , P)$ ، ندعو القاعدات بالشكل $A \to B$ ، حيث $A , B \in V_N$

" ممنوعة " أما بقية القاعدات فتسمى " مسموحة " .

لنشكل مجموعة جديدة P_1 من القاعدات ، حيث نضع فيها أولا جميع القاعدات المسموحة من P .

إذا كان بالنسبة لرموز غير طرفية A , B

(*) يوجد $X_1 , X_2 , \ldots, X_n \in V_N , n \geq 0$ ،

حيث $A \Rightarrow X_1 \Rightarrow \ldots \Rightarrow X_n \Rightarrow B$ ،

نضع في P_1 إضافة إلى القاعدات المسموحة $B \to u$ أيضا القاعدة $A \to u$.

لنقرر فيما إذا بالنسبة لـ A , B معطية تتحقق العلاقة (*) ، يمكن الاستعانة بالخوارزمية المشكلة في برهان النظرية (2.3.5) .

ننشئ القواعد $G = (V_N , V_T , S , P_1)$. و لنبرهن أن $L (G) = L (G_1)$.

إذا كانت $A \to u$ قاعدة من G_1 ، عندئذ يكون $A \Rightarrow^*_G u$. ينتج من ذلك أن $L (G) \subseteq L (G_1)$.

لنفرض الآن أن $w \in L (G)$ ، إذا يوجد مشتقة يسارية

$$S = w_0 \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_n = w \qquad (**)$$

ما دامت جميع القاعدات المستخدمة مسموحة ، تكون (**) أيضا مشتقة في G_1 .

إذا كان مستخدما في المشتقة قاعدة ممنوعة ، مثلا في الخطوة $w_j \Rightarrow w_{j+1}$ ، نجد أكبر $i < j$ وأصغر $k > j$ ، حيث يكون في الخطوة $w_i \Rightarrow w_{i+1}$ وفي الخطوة $w_k \Rightarrow w_{k+1}$ مستخدم قاعدة مسموحة .

يكون إذا في مقطع المشتقة $w_{i+1} \Rightarrow \dots \Rightarrow w_{k+1}$ مستخدما فقط قاعدات ممنوعة . وبما أن (**) مشتقة يسارية ، يكون في هذا المقطع دوما فقط موصوف الرمز غير الطرفي الأيسر في الكلمات $w_{i+1} , \dots , w_k$.

ولهذا يكون $w_{i+1} \Rightarrow_{G1} w_k$ حسب تعريف P_1 ، لذلك نضع في المشتقة (**) جميع القواعد الممنوعة المستخدمة باستخدام قاعدات من P_1 . وبالتالي يكون $w \in L (G_1)$.

وبذلك يتم برهان أن $L (G) = L (G_1)$.

3.2.6. **نظرية** .

لكل لغة غير قرينة L يوجد قواعد G بصيغة نموذج تشومسكي بالشكل

$. L - \{ \varepsilon \} = L (G)$

البرهان . ليكن $G' = (V'_N , V'_T , S , P')$ قواعد مولدة للغة $L - \{ \varepsilon \}$. بوساطة النظرية (5.2.5) والمبرهنة (2.2.6) نستطيع اختيار G' ، حيث تكون واضحة ولا تحوي شروطاً بالشكل $A \to B$ ، حيث A , B رموز غير طرفية .

إذا كانت القواعد تحوي شرطاً بالشكل $A \to x$ حيث x رمز وحيد ، عندئذ يكون x طرفياً والشرط يكون بشكل مسموح.

ولهذا يكفي الحفاظ على الشروط بالشكل $A \to B_1 , \dots , B_n$ حيث $n \geq 2$. وهذا الشرط نستبدله بمجموعة شروط بشكل ملائم . نعرف المتغيرات $C_1 , \dots , C_n$ بالشكل التالي :

إذا كان B_i متغيراً غير طرفي ، يكون $C_i = B_i$. بينما إذا كان B_i طرفياً ،يكون C_i متغيراً معطى من جديد .

لنشكل قواعد جديدة $G'' = (V''_N , V''_T , S , P'')$ ، حيث كل شرط في G' من الشكل $A \to B_1 , \dots , B_n$ يستبدل بمجموعة الشروط

$\{ A \to C_1 \dots C_n \} \cup \{ C_i \to B_i \}$ ، حيث B_i طرفي وموصف أعلاه.

إذا كان $L (G') = L (G'')$, وكان أيضا $u \Rightarrow^*_{G'} v$ ، عندئذ يكون $u \Rightarrow^*_{G''} v$ ، وبالتالي ينتج

$L(G') \subseteq L(G'')$.

لنبرهن الآن بالتحريض حسب طول المشتقة ، أنه من المشتقة $A \Rightarrow^*_{G''} w$ (حيث $w \in V'_T, A \in V'_N$) تنتج المشتقة $A \Rightarrow^*_{G'} w$.

- من أجل مشتقة بطول 1 يكون البرهان واضحاً .

- نفرض أنه من أجل مشتقة بطول $(k \geq 2)$ k تكون محققة .

ليكن $A = w_0 \Rightarrow w_1 \ldots w_k = w$ $\quad (*)$

مشتقة في G'' . أول خطوة يجب أن تكون بالشكل $(n \geq 2)$ $A \Rightarrow C_1 \ldots C_n$.

عندئذ يمكن كتابة

$W = u_1 u_2 \ldots u_n$ ، حيث $C_i \Rightarrow^*_{G''} u_i$ ، $1 \leq i \leq n$.

القواعد G' تحوي شرطاً ملائماً $A \rightarrow B_1 \ldots B_n$ ، حيث $B_i = C_i$ ، من أجل B_i غير طرفي و C_i متغير جديد بالنسبة لـ B_i طرفي . إذا كان C_i متغيراً جديداً ، يمكن عبر مسار المشتقة حسب القواعد G'' استخدام فقط شروط بالشكل $C_i \rightarrow B_i$ وبالتالي يكون $u_i = B_i$.

إذا كان $B_i = C_i$ فإنه يوجد مشتقة $C_i \Rightarrow^*_{G''} w_i$ بطول k على الأكثر وحسب الفرض التحريضي يكون $C_i \Rightarrow^*_{G'} w_i$. ينتج من هذا أن $A \Rightarrow^*_{G'} w_i$.

وبالتالي ينتج $L(G'') = L(G')$.

نشكل للقواعد G''' الآن قواعد مكافئة $G = (V_N , V_T , S , P)$ بصيغة نموذج تشومسكي .

نضع في سجل الشروط P في الصف الأول جميع شروط P'' من الشكل $A \to a$.

ونستبدل كل شرط في P'' من الشكل $A \to C_1 \dots C_n$ $(n \geq 3)$ بمجموعة شروط بالشكل:

$A \to C_1 D_1$

$D_1 \to C_2 D_2$

....

$D_{n-3} \to C_{n-2} D_{n-2}$

$D_{n-2} \to C_{n-1} C_n$

من الواضح أن $L(G) = L(G'')$ وبذلك يتم برهان النظرية.

4.2.6. مثال .

لننشئ قواعد بصيغة نموذج تشومسكي الطبيعي ، تولد اللغة $L = \{ 0^n 1^n ; n \geq 1 \}$.

تولد اللغة L مثلا بالقواعد

$G = (\{ S \} , \{ 0 , 1 \} , S , \{ S \to 0S1 , S \to 01 \})$.

نجري التشكيل نفسه الموصوف في برهان النظرية السابقة على الشروط بصيغة نموذج تشومسكي الطبيعي .

نستبدل الشرط $S \to 0S1$ بجملة الشروط $S \to C_1SC_2$, $C_1 \to 0$, $C_2 \to 1$.

ونستبدل الشرط $S \to 01$ بجملة الشروط $S \to C_3C_4$, $C_3 \to 0$, $C_4 \to 1$.

بقي الآن تصحيح الشرط $S \to C_1SC_2$ ، حيث نستبدله بالشرطين $S \to C_1D1$, $D_1 \to SC_2$.

نحصل في النتيجة على قواعد بصيغة نموذج تشومسكي بالشروط :

$S \to C_1 D_1$ $\qquad C_1 \to 0$

$D_1 \to S C_2$ $\qquad C_2 \to 1$

$S \to C_3 C_4$ $\qquad C_3 \to 0$ $\qquad C_4 \to 1$

كل شجرة اشتقاق مشكلة وفق قواعد بصيغة نموذج تشومسكي الطبيعي تملك خاصتين :

a) من كل عقدة داخلية يخرج ضلعان أو ضلع واحد ،

b) من العقدة يخرج ضلع واحد إذا وفقط إذا دخل هذا الضلع إلى ورقة .

تمكننا هذه الخواص غالبا من تبسيط براهين بعض التأكيدات عن اللغات غير القرينة .

نعرض فيما يلي برهان تأكيد حول المعنى الأساسي لدراسة اللغات غير القرينة . وتعرف النظرية التي سنوردها عادةِ بـ " نظرية – uvwxy " أو " نظرية الضخ ".

5.2.6. نظرية .

لتكن L لغة غير قرينة ، عندئذ من أجل كل كلمة $z \in L$ توجد كلمات مسموحة p , q ، $|z| > p$. و يمكن وصف z بالشكل $z = uvwxy$ ،حيث $|vwx| \leq q$ ، (واحدة على الأقل من الكلمات v , x مختلفة عن ε) ويكون $uv^iwx^iy \in L$ من أجل جميع $i \geq 0$.

البرهان . لتكن $G = (V_N , V_T , S , P)$ قواعد غير قرينة بصيغة نموذج تشومسكي الطبيعي مولدة للغة L . وليكن k عدد الرموز غير الطرفية في هذه القواعد . نضع $p = 2^{k-1}$, $q = 2^k$. ونستخدم فيما يلي خواص مختلفة لأشجار مشتقات القواعد المناسبة بصيغة نموذج تشومسكي الطبيعي .

إذا لم تحو مثل هذه الشجرة أي مسار أطول من j ، عندئذ سلسلة الطرفيات الناتجة وفق هذه الأسس تكون بطول 2^{j-1} على الأكثر . من كل عقدة يخرج ضلعان على الأكثر . الشجرة ،

التي يخرج من كل عقدة فيها ضلعان وكل مسار فيها يكون بطول j ، يملك 2^j ورقة . عند دراسة أشجار المشتقات نرى أن الضلع الذي يقود إلى ورقة اختيارية تقابل شرطاً من الشكل $A \to a \quad (a \in V_T \;,\; A \in V_N)$.

إذن يكون عدد الورقات يساوي عدد العقد ، التي تسبق تلك الورقات ، وكل منها يساوي 2^{j-1} على الأكثر .

ينتج ، إذا كانت $z \in L$ كلمة بطول أكبر من p ، فإنه يوجد في شجرة اشتقاق اختيارية مناسبة لمشتقة w ، مسار بطول أكبر من k . نختار شجرة T واحدة فقط من تلك الشجرات المشتقة ونختار إحدى المسارات C فيها والتي تملك أكبر طول . يمكن على هذا المسار أن نختار ثلاث عقد u_1 , u_2 , u_3 بالخواص التالية :

1) كلا العقدتين u_1 , u_2 موصوفتان بغير طرفي A .

2) العقدة u_1 تقع أقرب إلى جزر الشجرة من العقدة u_2 .

3) u_3 ورقة .

4) المسار من u_1 إلى u_3 يكون بطول $k+1$ على الأكثر .

العقدة u_1 تعين في الشجرة T شجرة جزئية T_1 ، تملك جذراً هو العقدة u_1 . وبشكل مشابه العقدة u_2 تعين شجرة جزئية T_2 . في النهاية تكون T_2 شجرة جزئية من T_1 . الشجرة T_1 تقابل مشتقة يسارية لكلمة ما z_1. طول هذه الكلمة يكون 2^k على الأكثر .

إذا كان في T_1 يوجد مسار ما بطول أكبر من $k+1$ من u_1 إلى ورقة ما u_4 ، عندئذ المسار في T من الجذر إلى الورقة u_4 يكون أطول من المسار المختار C. وهذا يناقض الفرض . ينتج أن $|z_1| \le q$. والشجرة T_2 تمثل مشتقة لكلمة ما z_2 . بما أن T_2

شجرة جزئية من T_1 يكون $z_1 = z_3\ z_2\ z_4$ ، من أجل كلمات ما z_3، z_4. والشرط المقابل للضلع الخارج من u_1 يجب أن يكون بالشكل $A \to BC$. لهذا واحدة من الكلمات z_3 ، z_4 على الأقل تكون غير فارغة . وبهذا نكون برهنا أن

$$(*) \quad A \Rightarrow^*_G z_3\ A\ z_4 \Rightarrow^*_G z_3\ z_2\ z_4\ , \quad |\ z_3\ z_2\ z_4\ | \le q\ , \quad z_3\ z_4 \neq \varepsilon\ .$$

ينتج من (*) أن :

$$A \Rightarrow^* z^i_3\ z_2\ z^i_4 \quad , i \ge 1$$

كما نعلم أن $A \Rightarrow_G^* z^0_3\ z_2\ z^0_4$. وبالتالي يكون $A \Rightarrow_G^* z^i_3\ z_2\ z^i_4$ من أجل $i \ge 0$.

الكلمة z يمكن بالطبع وصفها بالشكل $u\ z_3\ z_2\ z_4\ y$. نضع $z_3 = v\ , z_2 = w\ , z_4 = x$ وينتهي بذلك البرهان .

6.2.6. مثال .

برهن أن اللغة $L = \{ a^n b^n c^n \ ;\ n \ge 1 \}$ لا تشكل لغة غير قرينة .

لبرهان ذلك نفرض العكس ، أي أن L لغة غير قرينة ، وبالتالي توجد أعداد $p\ ,\ q$ بالخواص المدونة في برهان النظرية السابقة . نختار $k \ge p/3$ ، عندئذ يمكن كتابة الكلمة بالشكل التالي $p_1 = a^k\ b^k\ c^k$

$P_1 = uvwxy\ \ , vx \neq \varepsilon$ ، وكل كلمة بالشكل $P_i = uv^i\ wx^i\ y$ تنتمي إلى L .

بما أنه في جميع الكلمات في اللغة L تظهر الرموز $a\ , b\ , c$ بترتيب حرفي ، وبما أن $p_i \in L\ \ , I > 1$ ، يمكن كل من $v\ , x$ أن تحوي الرموز نفسها فقط (مثلا الكلمة v تحوي فقط رموز b ، و x تحوي رموز c) . إذا في الكلمات p_i مع تنامي i يتزايد عدد

رمزين فقط ، أما الرموز من النوع الثالث يبقى دوما بالعدد نفسه ، وهذا يناقض تعريف L

. وبالتالي الفرض المؤقت غير صحيح ، واللغة L تكون غير قرينة .

ينتج مما سبق أنه يمكن تقوية التأكيد 3.2.5.

7.2.6. تأكيد .

$$£_1 \supset £_2 \supset £_3$$

البرهان . اللغة $L = \{ a^n b^n c^n \; ; \; n \geq 1 \}$ لغة قرينة ، مولدة مثلا بقواعد قرينة بالشروط :

$X_0 \to X_0 Z_1$	$X_0 \to X_1 Z_1$	$X_1 \to a Y_1$
$X_1 \to a X_1 Y_1$	$Y_1 Z_1 \to Y_1 Z_3$	$Y_1 Z_3 \to Y Z_3$
$YZ_3 \to Y Z$	$Y_1 Y \to Y_1 Y_2$	$Y_1 Y_2 \to Y Y_2$
$YY_2 \to Y Y_1$	$ZZ_1 \to Z_2 Z_1$	$Z_2 Z_1 \to Z_2 Z$
$Z_2 Z \to Z_1 Z$	$Y \to b$	$Z \to c$

عندما يكون X_0 رمزاً أولياً و a , b , c رموزاً طرفية ، ينتج من المثال 6.2.6. أن L

$\in £_1 - £_2$.

وبالتالي ينتج أن $£_1 \supset £_2$. بشكل مشابه ينتج من 2.4.4. و 4.2.6 أن اللغة

$\{ 0^n 1^n \; ; n \geq 1 \}$ غير قرينة ، ولكنها غير منتظمة وبالتالي يكون $£_2 \supset £_3$.

8.2.6. نظرية .

لكل قواعد غير قرينة G يوجد خوارزمية ، نستطيع بموجبها إيجاد أعداد m , n ، حيث تكون $L (G)$ لغة غير منتهية إذا وفقط إذا كان يوجد $z \in L (G)$ ، حيث $m < | z | \leq n$.

البرهان .

يمكن الفرض بأن القواعد G بصيغة نموذج تشومسكي الطبيعي (بفضل الإجراء المتبع في برهان النظرية 3.2.6) . وحسب نظرية - $uvwxy$ ، يوجد أعداد مناسبة p , q . نضع $n = p + q , \ m = p$.

إذا كانت $L (G)$ تحوي كلمة z حيث $p < | z | < p+q$ ، عندئذ يمكن كتابة z بالشكل

$$(*) \qquad z = uvwxy \ , \ v \neq \varepsilon \ , \text{or} \ x \neq \varepsilon \ \text{and} \ uv^i wx^i y \in L (G)$$

من أجل كل $i \geq 0$. ينتج أن $L (G)$ غير قرينة .

نفرض على العكس $L (G)$ غير منتهية . عندئذ تحوي عددا كبيراً من الكلمات بطول أكبر من p . من مجموعة هذه الكلمات نختار كلمة ما z بطول أصغري . ولنبرهن أن: $p < | z | \leq p+q$.

إذا كان $| z | > p+q$ ، عندئذ يمكن كتابة z بالشكل (*) ، حيث من جديد تكون v أو x غير فارغة و $| vwx | \leq q$ و $uv^i wx^i y \in L (G)$ ، من أجل جميع $i \geq 0$ ، خاصة من أجل $i = 0$.

وبالتالي $uwy \in L(G)$ ولكن $| uvwxy | < | uwy |$ ، في حين

$| uwy | > (p + q) - q = p$ ، وهذا يناقض تشكيل الكلمة z .

9.2.6. تمارين .

1.2.6. أنشئ قواعد بصيغة نموذج تشومسكي الطبيعي مولدة للغة

$\{ 0^i 1^i 0^j 1^k \ ; i \geq 1 , j \geq 1 , k = 2j \}$.

2.2.6. برهن أن اللغات التالية لا تمثل لغات غير قرينة :

a) $\{ 0^i 1^j 0^i \ ; 0 \leq j \leq i \}$.

b) $\{ 0^i 1^j 0^i \cancel{1^j} ; i , j \geq 0 \}$.

c) $\{ ww \ ; w \in \{ 0 , 1 \}^* >$

d) $\{ a^i b^j c^k \ ; 0 \leq i \leq j \leq k \}$.

e) $\{ a^i b^j c^k \ ; \ i \neq j \ , j \neq k , i \neq k \}$.

3.6. نموذج كريباخوف الطبيعي.

1.3.6. تعريف .

تكون القواعد غير القرينة G بصيغة نموذج كريباخوف الطبيعي فقط إذا كانت جميع شروطها بالشكل $A \rightarrow au$ ، حيث A رمز غير طرفي ، a طرفي و u سلسلة رموز غير طرفية (أو فارغة) .

2.3.6. مبرهنة .

تدعى " A – شروط " جميع شروط القواعد غير القرينة ، التي تحوي A على الجهة اليسرى .

لتكن $G = (V_N , V_T , S , P)$ قواعد غير قرينة . نأخذ $A \to u_1Bu_2$ أحد شروط القواعد G ،

ولتكن $B \to v_1 , B \to v_2 , \ldots , B \to v_n$ جميع B – شروطها .

نشكل القواعد $G' = (V_N , V_T , S , P')$ ، حيث نأخذ من P الشروط $A \to u_1 B u_2$ ، ونضيف لها الشروط $A \to u_1v_1u_2 , \ldots , A \to u_1v_n u_2$. عندئذ ينتج أن $L (G) = L (G')$.

3.3.6. مبرهنة .

لتكن $G = (V_N , V_T , S , P)$ قواعد غير قرينة . ولتكن $A \to A u_1 , \ldots , A \to A u_r$ جميع A – شروطها ، التي تحوي في الجهة اليمنى A كرمز أيسر .

ولتكن $A \to v_1 , A \to v_2 , \ldots , A \to v_m$ جميع A – شروطها الباقية من القواعد G .

نشكل على أساس القواعد G قواعد جديدة $G' = (V_N \cup \{ Z \} , V_T , S , P')$ ، حيث نضيف رمزاً غير طرفي جديد Z ونستبدل جميع A – شروط القواعد G بمجموعة الشروط :

1) $A \to v_i \ , \ A \to v_i Z \quad , \quad 1 \le i \le m .$

2) $Z \to u_i \;,\; Z \to u_i Z \;,\quad 1 \le i \le r$.

عندئذ يكون $L(G) = L(G')$.

من المبرهنتين السابقتين تنتج النظرية التالية :

4.3.6. نظرية .

لكل لغة غير قرينة L توجد قواعد بصيغة نموذج كريياغوف الطبيعي ، حيث

$L(G) = L - \{\varepsilon\}$.

5.3.6. مثال .

لدينا قواعد G برمز أولي A_1 ورموز طرفية 1 , 0 وجملة الشروط التالية :

$A_1 \to A_2 A_1 \qquad A_3 \to A_1 A_3$

$A_1 \to A_2 A_3 \qquad A_2 \to 0$

$A_2 \to A_1 A_2 \qquad A_3 \to 1$.

أوجد قواعد G' بصيغة نموذج كريياغوف الطبيعي تقابل القواعد المعطية G .

خطوة 1 :

A_1 – شروط : لا تحتاج إلى تعديل ، لأن الجهة اليمنى تبدأ برمز غير طرفي بأعلى دليل .

A_2 – شروط : الشرط $A_2 \to 0$ أيضا لا يحتاج إلى تعديل . ولنبدأ بالتعديل إذا من الشرط $A_2 \to A_1 A_2$ ، حيث نحصل منه على الشروط

$A_2 \to A_2 A_1 A_2 \;,\; A_2 \to A_2 A_3 A_2$.

وتصبح جملة الشروط بالشكل :

$A_1 \to A_2 A_1 \qquad A_3 \to A_1 A_3$

$A_1 \rightarrow A_2\ A_3 \qquad A_2 \rightarrow 0$

$A_2 \rightarrow A_2\ A_1\ A_2 \qquad A_3 \rightarrow 1$

$A_2 \rightarrow A_2\ A_3\ A_2$

A_2 - شروط : تعدل وفق المبرهنة 3.3.6 ، حيث نضيف رمزاً غير طرفي جديد Z ،

ونستبدل A_2 - شروط

الناتجة بين شروط القواعد بجملة الشروط :

$A_2 \rightarrow 0 \qquad Z_2 \rightarrow A_1\ A_2 \qquad Z_2 \rightarrow A_3\ A_2$

$A_2 \rightarrow 0\ Z_2 \qquad Z_2 \rightarrow A_1\ A_2\ Z_2 \qquad Z_2 \rightarrow A_3\ A_2\ Z_2$

وتصبح كامل جملة الشروط بالشكل :

$A_1 \rightarrow A_2\ A_1 \qquad Z_2 \rightarrow A_1\ A_2\ Z_2$

$A_1 \rightarrow A_2\ A_3 \qquad Z_2 \rightarrow A_3\ A_2$

$A_2 \rightarrow 0 \qquad Z_2 \rightarrow A_3\ A_2\ Z_2$

$A_2 \rightarrow 0\ Z_2 \qquad A_3 \rightarrow A_1\ A_3$

$Z_2 \rightarrow A_1\ A_2 \qquad A_3 \rightarrow 1$

حسب المبرهنة 2.3.6 نصلح A_3 – شرط ، حيث نستبدل الشرط $A_3 \rightarrow A_1\ A_3$ بجملة الشروط

$A_3 \rightarrow A_2\ A_1\ A_3$ ، $A_3 \rightarrow A_2\ A_3\ A_3$

نعوض هذه الشروط (وفق المبرهنة 2.3.6) بالجملة

$A_3 \rightarrow 0\ A_1\ A_3 \qquad A_3 \rightarrow 0\ A_3\ A_3$

$A_3 \to 0\, Z_2\, A_1\, A_3 \qquad A_3 \to 0\, Z_2\, A_3\, A_3$

وتصبح كامل جملة شروط القواعد بالشكل :

$A_1 \to A_2\, A_1 \qquad A_3 \to 0\, A_1\, A_3$

$A_1 \to A_2\, A_3 \qquad A_3 \to 0\, Z_2\, A_1\, A_3$

$A_2 \to 0 \qquad A_3 \to 0\, A_3\, A_3$

$A_2 \to 0\, Z_2 \qquad A_3 \to 0\, Z_2\, A_3\, A_3$

$A_2 \to A_1\, A_2 \qquad A_3 \to 1$

$Z_2 \to A_1\, A_2\, Z_2 \qquad Z_2 \to A_3\, A_2$

$Z_2 \to A_3\, A_2\, Z_2$

خطوة 2 .

جميع A_2 - و A_3 - شروط تحوي على الجهة اليمنى كلمات تبدأ برمز طرفي . نستعين بذلك لتصنيف A_1 - شروط ، حيث تبدأ الأطراف اليمنى برمز طرفي .

من الشرط $A_1 \to A_2\, A_1$ نحصل على $A_1 \to 0\, Z_2\, A_1$, $A_1 \to 0\, A_1$.

من الشرط $A_1 \to A_2\, A_3$ نحصل على $A_1 \to 0\, Z_2\, A_3$, $A_1 \to 0\, A_3$.

وتصبح كامل جملة الشروط بالشكل :

$A_1 \to 0\, A_1 \qquad A_2 \to 0\, Z_2 \qquad A_3 \to 1$

$A_1 \to 0\, Z_2\, A_1 \qquad A_3 \to 0\, A_1\, A_3 \qquad Z_2 \to A_1\, A_2$

$A_1 \to 0\, A_3 \qquad A_3 \to 0\, Z_2\, A_1\, A_3 \qquad Z_2 \to A_1\, A_2\, Z_2$

$A_1 \to 0\ Z_2\ A_3$ $\quad$ $A_3 \to 0\ A_3\ A_3$ $\quad$ $Z_2 \to A_3\ Z_2$

$A_3 \to 0$ $\quad$ $A_3 \to 0\ Z_2\ A_3\ A_3$ $\quad$ $Z_2 \to A_3\ A_2\ Z_2$

الخطوة 3 .

الجزء الأول من الشروط أصبح في الشكل المناسب . بقي تصحيح Z_2 - شروط .

نبدل الشرط $Z_2 \to A_1\ A_2$ بجملة الشروط

$Z_2 \to 0\ A_1\ A_2$ $\quad$ $Z_2 \to 0\ A_3\ A_2$

$Z_2 \to 0Z_2\ A_1\ A_2$ $\quad$ $Z_2 \to 0Z_2\ A_2\ A_3$

كما نبدل بشكل مشابه الشروط

$Z_2 \to A_1\ A_2\ Z_2$ $\quad$ $Z_2 \to A_3\ A_2$ $\quad$ $Z_2 \to A_3\ A_2\ Z_2$

على التسلسل بجملة الشروط

$Z_2 \to 0A_1\ A_2\ Z_2$ $\quad$ $Z_2 \to 0Z_2\ A_2 Z_2$ $\quad$ $Z_2 \to 0\ A_3\ A_2\ Z_2$

$Z_2 \to 0Z_2\ A_3\ A_2\ Z_2$ $\quad$ $Z_2 \to 0\ A_1\ A_3 A_2$ $\quad$ $Z_2 \to 0Z_2 A_1\ A_3\ A_2$

$Z_2 \to 0A_3\ A_3\ Z_2$ $\quad$ $Z_2 \to 0Z_2\ A_3\ A_3 A_2$ $\quad$ $Z_2 \to 1\ A_2$;

$Z_2 \to 0A_1\ A_3\ A_2\ Z_2$;

$Z_2 \to 0Z_2 A_1\ A_3\ A_2\ Z_2$ $\quad$ $Z_2 \to 0A_3\ A_3\ A_2\ Z_2$

$Z_2 \to 0Z_2 A_3 A_3\ A_2\ Z_2$ $\quad$ $Z_2 \to 1\ A_2\ Z_2$;

وتصبح الجملة النهائية للشروط بالشكل :

$A_1 \rightarrow 0\ A_1$ $\qquad$ $A_1 \rightarrow 0\ Z_2\ A_1$

$A_1 \rightarrow 0\ A_3$ $\qquad$ $A_1 \rightarrow 0\ Z_2\ A_3$

$A_2 \rightarrow 0$ $\qquad$ $A_2 \rightarrow 0\ Z_2$

$A_3 \rightarrow 0\ A_1\ A_3$ $\qquad$ $A_3 \rightarrow 0\ Z_2\ A_1\ A_3$

$A_3 \rightarrow 0\ A_3\ A_3$ $\qquad$ $A_3 \rightarrow 0\ Z_2\ A_3\ A_3$

$A_3 \rightarrow 1$

$Z_2 \rightarrow 0\ A_1\ A_2$ $\qquad$ $Z_2 \rightarrow 0 Z_2\ A_1\ A_1$

$Z_2 \rightarrow 0\ A_3\ A_2$ $\qquad$ $Z_2 \rightarrow 0 Z_2\ A_3\ A_2$

$Z_2 \rightarrow 0\ A_1\ A_2\ Z_2$ $\qquad$ $Z_2 \rightarrow 0 Z_2\ A_1\ A_2 Z_2$

$Z_2 \rightarrow 0\ A_3\ A_2\ Z_2$ $\qquad$ $Z_2 \rightarrow 0 Z_2\ A_3\ A_2 Z_2$

$Z_2 \rightarrow 0\ A_1\ A_3\ A_2$ $\qquad$ $Z_2 \rightarrow 0 Z_2\ A_1\ A_3 A_2$

$Z_2 \rightarrow 0\ A_3\ A_3\ A_2$ $\qquad$ $Z_2 \rightarrow 0 Z_2\ A_3\ A_3 A_2$

$Z_2 \rightarrow 1\ A_2$ $\qquad$ $Z_2 \rightarrow 0 A_1\ A_3\ A_2 Z_2$

$Z_2 \rightarrow 0 Z_2 A_1\ A_3\ A_2 Z_2$ $\qquad$ $Z_2 \rightarrow 0 A_3\ A_3\ A_2 Z_2$

$Z_2 \rightarrow 0 Z_2 A_3\ A_3\ A_2 Z_2$ $\qquad$ $Z_2 \rightarrow A_2 Z_2$.

6.3.6. تمارين.

1.3.6. أنشئ قواعد بصيغة نموذج كريباخوف الطبيعي للغات التالية :

(a $\{ w (w)^R \; ; w \in \{ 0 , 1 \}^* \}$

(b أصغر مجموعة B من الكلمات على الأبجدية $\{ p , q , \supset , \neg , [,] \}$ ،

حيث:

1. $p , q \in B$.
2. $u \in B \Rightarrow \neg u \in B$.
3. $u , v \in B \Rightarrow [u \supset v] \in B$.

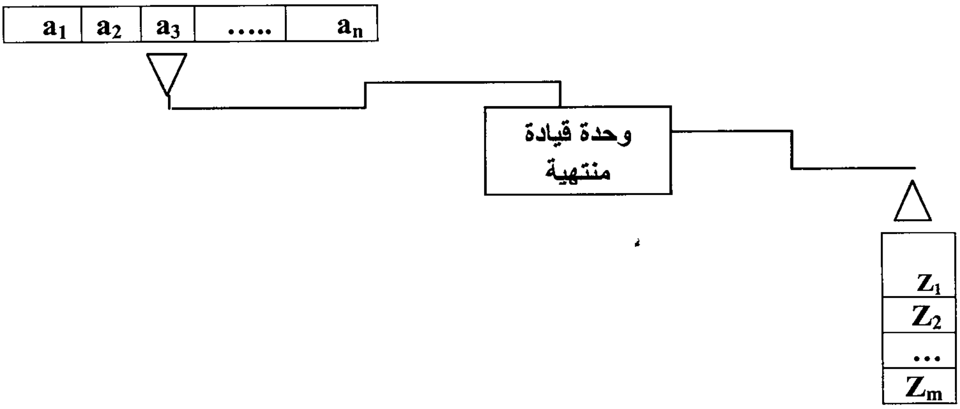
4.6. الحاسبة المكدس (المخزن Pushdown automaton) .

سنتعرف في هذه الفقرة على الحاسبة المكدس ، التي تلعب دوراً هاماً في نظرية اللغات الصورية (الشكلية) كنوع من الحاسبات المميزة بلغات غير قرينة .

تختلف الحاسبة المكدس عن الحاسبة المنتهية ، بأنها مزودة بنوع خاص من الذاكرة غير منتهية – مكدس (مخزن) (أنظر الشكل 7.6) . تعمل الحاسبة مع المكدس ، بحيث تأخذ من المكدس الرمز الأخير الذي وضع به . يمكن تخيل المكدس وكأنه وعاء نضع فيه من الأعلى رمزا فوق آخر ، وبذلك يحتوي المكدس في كل لحظة عموداً من الرموز .

عندما تقرأ الحاسبة الرمز الذي يقع في العمود في الموقع n من الأعلى ، فيجب أن تخرج كامل الجزء الذي يقع فوقه ، أي n-1 رمزاً . وسنفترض بأن المكدس يملك سعة غير محددة ، أي نستطيع تشكيل عمود غير محدود الطول .

عند تعريف الحاسبة المنتهية تم مقابلتها بتجهيزات ، تقبل في كل لحظة رمز دخل وحيد، مغاير للحالة الداخلية، ثم يقبل رمزاً آخر . الخ . أما الحاسبة المكدس فيملك مقابل ذلك إمكانية إيقاف قبول رموز دخل أخرى لفترة ما. ولهذا جرت العادة بأن يتم وصل تجهيزات الدخل المتوقعة عند الحاسبة المكدس بشريط دخل وحيد المسار ، أي بشريط يقرأ برأس الدخل من اليسار إلى اليمين . يستطيع رأس الدخل بعد الإيقاف (الفاصلة) الإزاحة إلى اليمين حقل واحد ، أو يبقى ثابتاً (أنظر الشكل 7.6) .



(شكل 7.6)

لكل حاسبة مكدس يوجد مجموعة منتهية من الحالات الداخلية . إذا وصلت الحاسبة إلى حالة معينة فيجب معرفة ما يلي :

1. رمز الدخل ،

2. الحالة التي وصلت إليها الحاسبة ،

3. الرمز الأعلى في المكدس .

حالما تتغير الحالة ، يحذف من المكدس الرمز الأعلى ويعوض بعمود رموز معين (أو فارغ) ، ويزاح الرأس على شريط الدخل حقل واحد على الأكثر إلى اليمين . ونقدم فيما يلي التعريف الدقيق للحاسبة المكدس :

1.4.6. تعريف .

يسمى حاسبة مكدس كل سباعية $M = (Q , X , \Sigma , \delta , q_0 , z_0 , F)$ ، حيث

Q مجموعة منتهية (مجموعة الحالات) ،

X مجموعة منتهية (أبجدية الدخل) ،

Σ أبجدية المكدس (مجموعة منتهية) ،

$q_0 \in Q$ الحالة الأولية ،

$z_0 \in \Sigma$ الرمز الأولي ، حيث عند بدء القراءة يحتوي المكدس فقط الرمز z_0 ،

$F \subseteq Q$ مجموعة الحالات النهائية ،

δ دالة من المجموعة $\Sigma \times (X \cup \{ \varepsilon \}) \times Q$ إلى مجموعة جميع المجموعات الجزئية المنتهية $Q \times \Sigma^*$.

نلاحظ من التعريف ، أن الحاسبة المكدس تكون بشكل عام غير محدودة . كما نوضح فيما يلي كيفية عرض الدال δ :

$$\delta (q , a , z) = \{ (p_1 , \gamma_1) , (p_2 , \gamma_2) , \ldots , (p_m , \gamma_m) \} ;$$

$q \in Q \ , \ a \in X \ , \ z \in \Sigma \ , p_i \in Q \ , \ \gamma_I \in \Sigma^* \ , \ I = 1 , 2 , \ldots , m$.

إذا كانت الحاسبة M في الحالة q ، تقرأ حرف الدخل ، ويكون في أعلى المكدس الرمز z ، عندئذ يمكن الانتقال إلى الحالة p_i ويعوض الرمز z بالسلسلة γ_i ، من أجل قيمة ما لـ $1 \le I \le m$.

عندما يطلب إلينا وضع كامل الكلمة في المكدس ، فيجب علينا أن نقرر من أية جهة سنضعها فيه . والأفضل أن نضع الرمز الأيسر من السلسلة γ في أعلى المكدس ، والرمز الثاني من اليسار يوضع في المكدس كرمز ثاني من الأعلى ، الخ. حتى يتم وضع الرمز الأخير (من اليمين) في أسفل المكدس .

2.4.6. تعريف .

ندعو وضع الحاسبة المكدس $M = (Q , X , \Sigma , \delta , q_0 , z_0 , F)$ كل ثلاثية (q , w , γ) ، حيث $q \in Q \ , \ w \in X^* \ , \gamma \in \Sigma^*$.

نقول إن الوضع $(q , a w , z \gamma)$ للحاسبة M يقود مباشرة إلى الوضع $(q' , w , \alpha \gamma)$ ، ونرمز لذلك

$(q , a w , z \gamma) \Rightarrow_M (q' , w , \alpha \gamma)$ ، $a \in X \cup \{ \varepsilon \}$ ، إذا كان $(q' , \alpha) \in \delta (q , a , z)$.

إذا كان E' , E وضعين للحاسبة M ، نقول إن الوضع E يقود إلى الوضع E' (رمزيا $E \Rightarrow_M^* E'$) فقط عندما يوجد أوضاع $E_1 , E_2 , \ldots , E_n$ للحاسبة M ، حيث $E_n = E'$, $E = E_1$ و

$E_i \Rightarrow_M E_{i+1}$ ، بالنسبة لجميع $1 \leq I \leq n-1$.

نستنج أن وضع الحاسبة المكدس يعطي :

1) الحالة التي تكون فيها الحاسبة ،

2) جزء كلمة الدخل المعتبر للحاسبة ،

3) محتوى المكدس .

يعرف قبول كلمة بالحاسبة المكدس بطريقتين مختلفتين :

a) بشكل مشابه للحاسبة المنتهية ، تكون الكلمة مقبولة إذا كان بعد شغل الرمز الأخير من كلمة الدخل وصلت إلى حالة نهائية – وهذا يدعى القبول بالحالة النهائية .

b) تكون الكلمة مقبولة فقط عندما تكون مشغولة كاملة ، ومكدس الحاسبة يكون فارغا – وهذا يدعى القبول بالمكدس الفارغ .

3.4.6. تعريف .

لتكن $M = (Q , X , \Sigma , \delta , q_0 , z_0 , F)$ حاسبة مكدس .

1) نعرف اللغة $T (M)$ المميزة بالحالة النهائية كما يلي :

$$T (M) =_{df} \{ w ; (q_0 , w , z_0) \Rightarrow_M^* (q , \varepsilon , \gamma) , q \in F , \gamma \in \Sigma^* \} .$$

2) نعرف اللغة $N (M)$ المميزة بالمكدس الفارغ كما يلي ك

$$N (M) =_{df} \{ w ; (q_0 , w , z_0) \Rightarrow_M^* (q , \varepsilon , \varepsilon) , q \in F \} .$$

إذا أردنا اقتراح حاسبة مكدس مميزة بمكدس فارغ ، تكون مجموعة الحالات النهائية F غير ضرورية . وفي مثل هذه الحالة نضع عادة $F = \phi$.

4.4.6. مثال.

أنشئ حاسبة مكدس تميز اللغة التالية $L(M) = \{ w(w)^{*} \; ; \; w \in \{0, 1\} \}$ بمكدس فارغ .

نأخذ $M = (\{q_1, q_2\}, \{0, 1\}, \{z, A, B\}, \delta, q_1, z, \phi)$ ،

حيث دالة النقل معرفة على النحو التالي :

1. $\delta(q_1, 0, z) = \{(q_1, Az)\}$
2. $\delta(q_1, 1, z) = \{(q_1, Bz)\}$
3. $\delta(q_1, 0, z) = \{(q_1, AA), (q_2, \varepsilon)\}$
4. $\delta(q_1, 0, B) = \{(q_1, AB)\}$
5. $\delta(q_1, 1, A) = \{(q_1, BA)\}$
6. $\delta(q_1, 1, B) = \{(q_1, BB), (q_2, \varepsilon)\}$
7. $\delta(q_2, 0, A) = \{(q_2, \varepsilon)\}$
8. $\delta(q_2, 1, B) = \{(q_2, \varepsilon)\}$
9. $\delta(q_1, \varepsilon, z) = \{(q_2, \varepsilon)\}$
10. $\delta(q_2, \varepsilon, z) = \{(q_2, \varepsilon)\}$.

لنبين فيما يلي طريقة تمييز اللغة المعطاة بالحاسبة المقترحة .

في بداية عمل الحاسبة ، نضع في المكدس قيد الأجزاء المقروءة من الكلمة. يشفر الرمز 0 برمز المكدس A ، والرمز 1 يشفر بالرمز B . ما دام يعتقد بأن الحاسبة تقرأ من النصف الأول من الكلمة ، يوضع في المكدس قيد الشفرة ونبقى في الحالة q_1 . وهذا يقابل الشروط 1. – 6 . وبكلام أوضح ، عند الشرطين 3. و 6. حيث يعتقد بأن الحاسبة وصلت إلى منتصف الكلمة ننتقل إلى الحالة q_2 . إذا كانت بالفعل الحاسبة وصلت إلى منتصف الكلمة ، وإذا كانت الكلمة المقروءة تنتمي إلى اللغة L ، عندئذ رمز الدخل الآن يجب أن يتطابق مع الرمز الذي كان مقروءاً في المرة السابقة . ولهذا يكون الانتقال إلى الحالة q_2 ممكنا فقط عند الشرطين 3. و 6. وعندما تصل M إلى الحالة q_2 تأخذ من المكدس الرموز وتقارنها مع رموز الدخل .

كما تختبر الشروط 7. و 8. فيما إذا كان الجزء الثاني للكلمة صورة عكسية للجزء الأول .

في الحالة المعاكسة، تصل M إلى وضع لا تكون فيه الدالة δ معرفة .

إذا قيمت الحاسبة وسط الكلمة بشكل صحيح إذا كانت الكلمة المعطاة تنتمي إلى L ، عندما تعمل الحاسبة الرمز الأخير في كلمة الدخل يبقى في المكدس فقط الرمز z . بعدئذ يمكن تطبيق الشرط 10. وتكون الكلمة مقبولة . أما الشرط 9. فيمكننا من قبول الكلمة الفارغة .

كمثال هام من هذا النوع ، نذكر الحاسبة المكدس المحدود ، أي الحاسبة التي يكون لها في كل وضع خيار واحد ممكن على الأكثر .

5.4.6. تعريف.

تدعى الحاسبة المكدس $M = (Q, X, \Sigma, \delta, q_0, z_0, F)$ حاسبة محدودة فقط عندما يتحقق الشرطان التاليان :

1) إذا كان $\delta(q, \varepsilon, z) \neq \phi$ ، فإن $\delta(q, a, z) = \phi$ ، من أجل $a \in Q$ اختيارية و $(q \in Q, \ z \in \Sigma)$.

2) تحوي $\delta(q, a, z)$ عنصراً واحداً على الأكثر ، من أجل $a \in Q$ اختيارية و $q \in Q, \ z \in \Sigma$.

6.4.6. **مثال** .

أنشئ حاسبة مكدس محدودة تميز اللغة $L(M) = \{0^n 1^n \ ; \ n \leq 1\}$ بطريقة المكدس الفارغ .

نأخذ الحاسبة $M = (\{q_1, q_2\}, \{0, 1\}, \{z, A\}, \delta, q_1, z, \phi)$ ،

حيث دالة النقل δ معرفة كما يلي :

$\delta(q_1, 0, z) = (q_1, Az)$

$\delta(q_1, 0, A) = (q_1, AA)$

$\delta(q_1, 1, A) = (q_2, \varepsilon)$

$\delta(q_1, 1, A) = (q_2, \varepsilon)$

$\delta(q_2, \varepsilon, z) = (q_2, \varepsilon)$

7.4.6. **تمارين**.

1.4.6. اقترح حاسبة مكدس تميز اللغة

$$L = \{ a^i b^i \ ; \ 0 \leq i \leq 3j \}.$$

2.4.6. برهن أن اللغات

a) $\{ a^i b^j c^i \ ; \ j > Ii \geq 0 \}$.

b) $\{ ww \ ; w \in \{ 0 , 1 \}^* \}$

قائمة المصطلحات العلمية

لغة الغول60	Algool 60
خوارزمية	Algorithm
مجموعة الدخل أو أبجدية الدخل	Alphabet
الأبجديات	Alphabets
تحليل الحاسبات	Analysis of Calculations
تطبيق	Application
تعقيد التقاربات	Asymptotic complexity
نظرية التشغيل الذاتي	Automata Theory
أبجدية ثنائية	Binary Alphabet
العد والحساب	Calculating And Counting
التنظيم القانوني	Canonical Ordering
الأصغر قانونيا	Canonically Smaller
الجداء الديكارتي	Cartesian product
هيكلية تشومسكي	Chomsky Hierarchy
إغلاق	closure
المترجمات، المفسرات	Compilers
المتممة	Complementation

نظرية التعقيد الحسابي	Complexity Theory
تركيب	Composition
نظرية التحسيب	Computability Theory
الحوسبة	Computation
عملية حوسبة	Computation Operation
النموذج الحاسوبي	Computational Model
التعقيد الحسابي	Computational Complexity
نظرية التعقيد الحسابية	Computational Complexity Theory
المسائل القابلة للحل حاسوبياً	Computationally Solvable Problems
الحوسبة	Computing
تتابع أو توالي	Concatenation
ضرب	Cross Product
معطيات	Data
الاشتقاق	Derivation
الحاسبة المنتهية المحددة	Deterministic Finite Automaton
الاجراء الفعلي	Effective procedure
متسلسلة خالية	Empty String
تكافؤ الحاسبات	Equivalence of Calculations
مجموعة الحالات النهائية(الهدفية)	Final States

التشغيل الذاتي المنته	Finite Automaton
لغة شكلية	Formal Language
نظام شكلي	Formal System
التشكيل الصوري	Formalization
جدول الدالة	Function Table
القواعد	Grammars
تشاكل تبولوجي	Homomorphism
المعلومات	Information
الحالة الابتدائية	Initial State
مدخلات	Inputs
الأوامر أو التعليمات	Instructions
الخطوات الوسطية	Intermediate Steps
مترجم	Interpretation
تقاطع	Intersection
تطابق حاسوبي	Isomorphism
لغة	Language
حاسبة ميلي التعاقبية	Mealys Sequential Machine
الذاكرة	Memory
حاسبة مور التعاقبية	Moors Sequential Machine

الحاسبة المنتهية غير المحددة	Non Deterministic Finite Automaton
رموز غير طرفية	Nonterminal Symbols
ترتيب السلاسل	Ordering Of Strings
مخرجات	Outputs
الوصل على التوازي	Parallel Connection
الأطراف أو الجهات	Parties
متسلسلات تبادلية	Permutations Strings
قواعد بناء التعابير	Phrase Structure Grammars
مخطط	Plan
مشاكل	Problems
العملية	Process
المعالجات	Processors
قواعد إنتاج	Production Rules
اللغة البرمجية	Programming Language
البرنامج	Programs
متسلسلات بادئة تامة	Proper Prefixes Strings
متسلسلات لاحقة تامة	Proper Suffixes Strings
حاسبة مكدس	Pushdown automaton
ذاكرة الوصول العشوائي	Random-Access Memory(RAM)

تحقيق	Realization
المستقبل	Receiver
الاختزال	Reducibility
مختزلة	Reductive
الحاسبات المختزلة	Reductive Calculations
اللغات المنتظمة	Regular Language
تكرار	Repeat
التمثيل	Representation
تمثيل المعلومات	Representation Of Information
متسلسلة عكسية	Reverse String
القوانين أو الأحكام	Rules
عملية البحث	Searching Operation
المرسل	Sender
جملة	Sentence
الوصل التسلسلي	Sequential Connection
مجموعة الحالات	Set Of States
تجهيزات برمجية	software
قابلة للحل	Solvable
البداية	Start

مخطط الحالات	States Diagram
شجرة الحالات	States Tree
متسلسلة	String
طول المتسلسلة	String Length
مجموعة جزئية	Subset
خاصية إبدالية	Substitution Property
متسلسلة فرعية	Substring
اقتراح لحل مسألة	Suggestion Of Solving Problem
قواعد اللغة (تركيب عبارة)	Syntax
معارف منظمة	Systematized
رموز طرفية	Terminal Symbols
الحاسوب العلمي النظري	Theoretical Science Computer
دالة النقل	Transition Function
آلة تورينغ	Turing Machine
قواعد من النوع 0	Type 0 Grammars
النموذجية	typical
أبجدية أحادية	Unary Alphabet
متسلسلة أحادية	Unary String
اتحاد	Union

كلمة	Word
الكتابة	Writing

.(references) المراجع الأجنبية

1. ADACHI A. Foundation of Computation Theory, Ohmsha , Led. , 1990.

2. AHO A. , V. , SETHIR. , ULLMANJ.D, Compilers , Principles , Techniques and TOOLS Addison – Wesley publishing company , 1986 .

3. CHOMSKY , N. On Certion Formal Properties Gammers Information and Control, 1995.

4. FLOYD R.W. , BEIGELR. The Language of Machines – An Intraduction to Computability and Formal Languages, Computer Science Prss-1994.

5. GINSGURG S. The Mathematical Theory of Context-Free Languages, McGraw Hill , 1966.

6. GINSGURG S. Algebraic and Automata – Theoretic Properties of Formal Languages, North Holland , 2005

7. HOPERCROFT J .E., ULLMAN J. D., Formal Languages and their Relation to Automata , ADDison-Wesley Publishing company , 1979 .

8. HOPERCROFT J. E. , ULLMAN J. D., Introduction to Automata theory , Languages , and computation, ADDISON-Wesley Publishing company , 2009.

9. SALAMAA A., Formal languages , Academic Press , 1973.

10. HARRISON M.A., I . M . Havel, Strict deterministic grammars, Journal of computer and system sciences , 1973.

11. HENNIE F., Finite-State Models For Logical Machines, J. Wiley , 1969.

12. HARTMANIS J., R. E. Stearns, Algebraic Theory Of Sequential Machines, Prentice- Hall , 1996.

13. ARBIB M.A. , Theories of abstract automata , prentice- Hall , 1969.

المراجع العربية

14. عمار خيربيك ، منصور فرح . مدخل إلى الأتومات واللغات الصورية . منشورات جامعة دمشق ، 2000 .

الشكر الجزيل للسادة المساهمين في إنجاز هذا الكتاب وهم :

المدقق اللغوي

الدكتورة وفاء جمعة المدرسة بكلية الآداب والعلوم الإنسانية

المدققون العلميون

الدكتور محمود عثمان أستاذ بكلية العلوم بجامعة تشرين

الدكتور رضوان دندة أستاذ بكلية الهندسة المعلوماتية بجامعة تشرين

الدكتور جبر حنا أستاذ بكلية الهندسة الميكانيكية والكهربائية بجامعة تشرين