



Rule Based Expert Systems النظم الخبيرة

د. خالد العمر

محتوى مجاني غير مخصص للبيع التجاري

مبادئ الذكاء الاصطناعي RB Informatics ; 25/04/2022

محاور المحاضرة

التضارب واستراتيجيات حلّه

مراجعة وحل مسائل عن

النظم الخبيرة وأنواعها

Forward chaining

Backward chaining

المعرفة: هي الخبرة العملية أو النظرية عن مجال / domain محدد بحيث يمكننا تحديد الصح والخطأ، ويدعى الأشخاص الذين يملكون هذه المعرفة ضمن المجال المحدد بال خبراء expert.

شكل قواعد النظم الخبيرة

If (condition) then → Action

if part

Action part

أي:

فعل → then (شرط) If

مثال:

If the traffic light is green, then the action is go.

Note: من الممكن أن تحتوي القواعد على عمليات منطقية، مثال:

If <condition> and
<condition> or
<condition> then
Action



هنا ال if part احتوى على عدد من ال condition بينها عمليات منطقية.

أنواع قواعد النظم الخبيرة Types of rules is expert system

1. Relation(العلاقة):

Ex: If the fuel tank is empty then the car is dead.

إذا كان خزان الوقود فارغاً، فإن السيارة لن تعمل (لن تتحرك).

2. Recommendation(شيء موصى به):

Ex: If the season is autumn and the sky is cloudy then the advice is taking an umbrella.

إذا كان الفصل هو الخريف والسماء مبلدة بالغيوم فالنصيحة هي أن تأخذ المظلة.

3. Directive(توجيه):

Ex: If the car is dead and the fuel tank is empty then the action is a refuel the tank.

إذا كانت السيارة لا تعمل وخزان الوقود فارغ فالحمل هو إعادة تزويد الخزان بالوقود.

4. Strategy(الاستراتيجية):

Ex: If the car is dead then the action is check fuel tank.

إذا كانت السيارة لا تعمل عندها يجب فحص خزان الوقود. هذا المثال يدعى (step one)

Ex: If step one is complete and the fuel tank is full then the action is checking the battery.

إذا اكتملت الخطوة الأولى وخزان الوقود ممتلئ فعندها يجب فحص البطارية.

5. Heuristic(تجريبية):

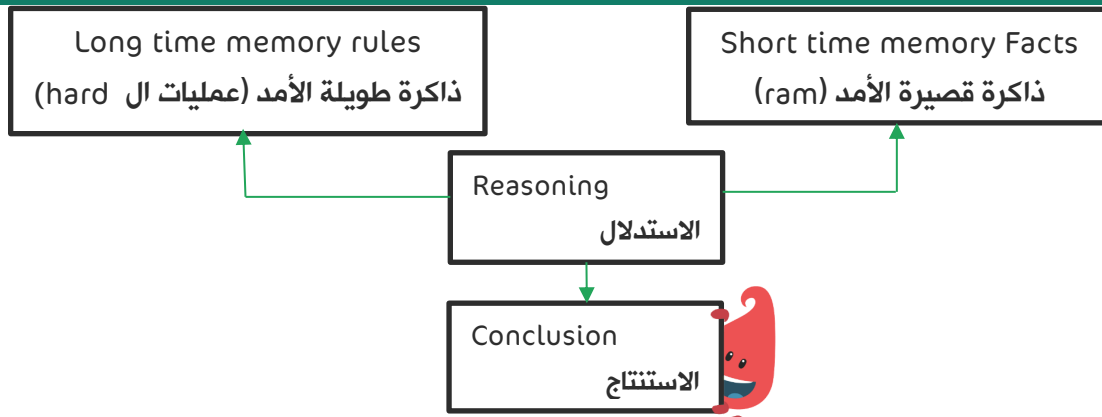
Ex: If the spill is liquid and the "spill phc6 then the spill material is "acetic acid".

Players in development of expert system:

- Project manager: مدير المشروع يقوم بالإشراف على المشروع
- Domain expert: الشخص الذي يملك خبرة بمجال محدد
- Knowledge engineer: هو الشخص الذي يتحدث مع الخبير ويستخلص منه المعرفة ويحولها إلى قواعد rules.
- Programmer: المبرمج
- End user: المستخدم النهائي للنظام

structure of rules based on expert system

بنية النظام الخبير المعتمد على القواعد



هل تخطئ النظم الخبيرة؟؟
يوجد فيها خطأ بالقواعد وبالتالي ينتج عنها خطأ بالاستنتاج أو النتائج.

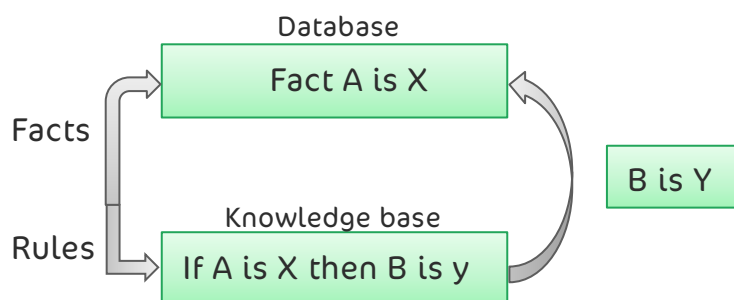
ملاحظات:

- ❖ عادة يتم فصل الحقائق عن ال Rules ، ويتم تخزين ال facts في قاعدة المعطيات database، وتخزين ال rules في قاعدة المعرفة knowledge base.
- ❖ إن قاعدة المعرفة تحتوي على rules ، وبالتالي عند تنفيذ أي قاعدة فإنه ينتج عنها أحد الأمرين: New Facts, Actions.
- ❖ ال reasoning يقوم به محرك الاستدلال inference engine وهو استنتاج facts جديدة بالاعتماد على rules, facts (عملية تطبيق المنطق).
- ❖ عند تشغيل البرنامج يتم تحميل facts على RAM ← short-time memory، أما rules فتكون مخزنة ب files على hard disk ← long-time memory.
- ❖ المعرفة مفصولة عن المعالجة.
- ❖ المعرفة هي: **facts** مخزنة ضمن ال database **Rules** مخزنة ضمن ال knowledge base
- ❖ المعالجة: هي inference engine تحمله ال facts و ال rules ليقوم بمعالجتها وفق ما ذُكر في المحاضرة السابقة إما ب forward أو backward.

تذكرة:

- ال forward يمر على ال rules بالتسلسل للبحث عن rule طرفها اليساري محقق فيقوم بعمل fire لها وإضافة ال fact الناتجة إلى قاعدة المعرفة.
- أما ال backward تنطلق من الهدف المنشود وتتشكل لدينا أهداف مرحلية ليبرهن على هذه الأهداف.
- **إذاً:** نستخدم forward إذا كان الهدف هو جمع المعلومات (facts)، ونستخدم backward إذا كان هناك هدف نريد إثبات أنه من ال facts.

مثال:



هنا تتم المعالجة بتغيير ال rules ، هل A is B موجودة، الجواب هنا نعم (لأنها موجودة بال facts)، لذلك يقوم بعمل fire فالنتيجة هي B is y أي نتيجة تنفيذ ال rules نضيفها إلى ال database ك fact جديدة، فبالتالي B is y أصبحت fact، تدعى هذه العملية ب (سلسلة الاستدلال).

🔗 تمارين:

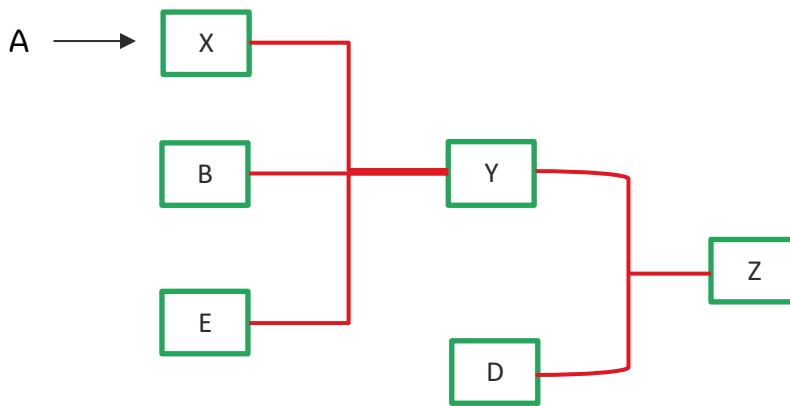
database
Facts: A, B, E, D

Rule1: if y is true, and D is true, then Z is true

Rule2: if X is true, and B is true, and E is true then Y is true.

R3: if A is true, then X is true.

الحل بطريقة السلسلة الأمامية: forward



شرح الحل: ننتقل من ال facts ونقوم ب scan for rules (نبدأ بالطرف اليساري if part).

Iteration1:

إن القاعدة رقم 3 محققة وبالتالي X محققة

Facts: A, B, E, D, X → X أصبح fact

Iteration2:

إن القاعدة 2 محققة حيث E, B, X

Facts: A, B, E, D, X, Y وبالتالي Y محققة

Iteration3:

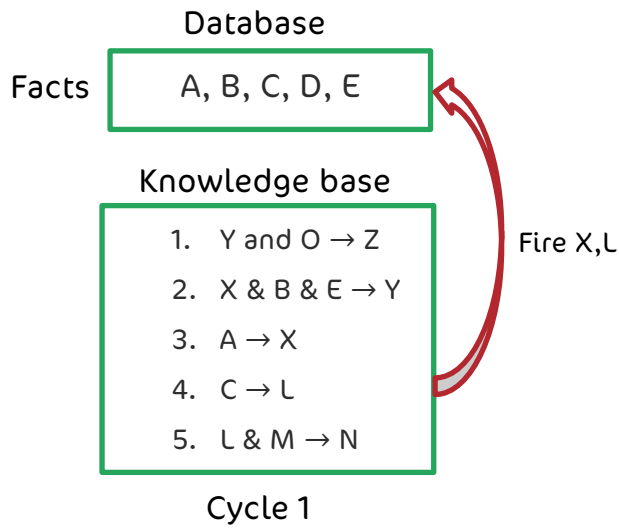
إن القاعدة 1 محققة حيث Y, D محققين

Facts: A, B, E, D, X, Y, Z وبالتالي Z محققة

إن ال facts تتضمن جميع atoms
المسألة، وهو المطلوب.



مثال هام عن ال forward chaining: الحل باستخدام ال forward:



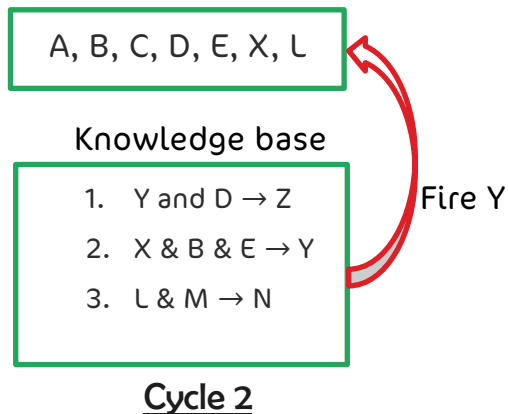
ليكن لدينا database تحوي A, B, C, D, E
ولدينا knowledge base تحوي ال rules التالية:



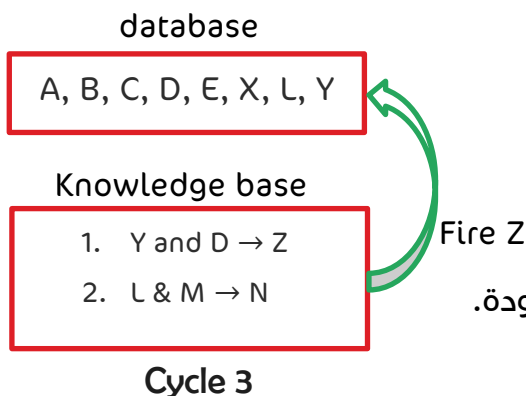
Goal is: Z

بتطبيق ال forward chaining على هذا المثال نقوم بالنظر
إلى ال facts وتحميلها.

ثم نقوم بفحص كل rule على حدى وبالترتيب: بالنظر إلى القاعدة 1 نجد أن طرفها اليساري يحوي على Y وال Y غير موجودة ضمن ال facts لذلك ننتقل إلى القاعدة التي تليها (القاعدة 2)، هنا نجد أنها تحوي بطرفها اليساري على X وهي غير موجودة ضمن ال facts أيضاً، نفحص القاعدة رقم 3، نجد أن طرفها اليساري يحوي على A وهي موجودة ضمن ال facts، **هنا** نقوم بعمل **fire** لإضافة X على ال facts (database)، أي X أصبحت fact، وبالتالى نقوم بإضافة L إلى database ثم ننتقل إلى القاعدة رقم 5 نجد أنها غير قابلة للتنفيذ لأنها تحتوي على M أي ضمن ال cycle1 تم تنفيذ قاعدتين فقط، ننتقل الآن إلى ال cycle2 مع حذف القواعد التي تم تنفيذها مع إعادة نفس الخطوات وعمل fire لإضافة كل جديد.



ضمن ال cycle 2 قمنا بإضافة ال Y إلى الحقائق،
نتقل إلى cycle 3 مع حذف القاعدة رقم 2.



ضمن ال Cycle 3 قمنا بإضافة ال Z إلى الحقائق
مع حذف القاعدة رقم 1.

هنا نجد أن L موجودة ضمن الحقائق لكن M غير موجودة.

database

A, B, C, D, E, X, L, Y, Z

Knowledge base

 $L \& M \rightarrow N$

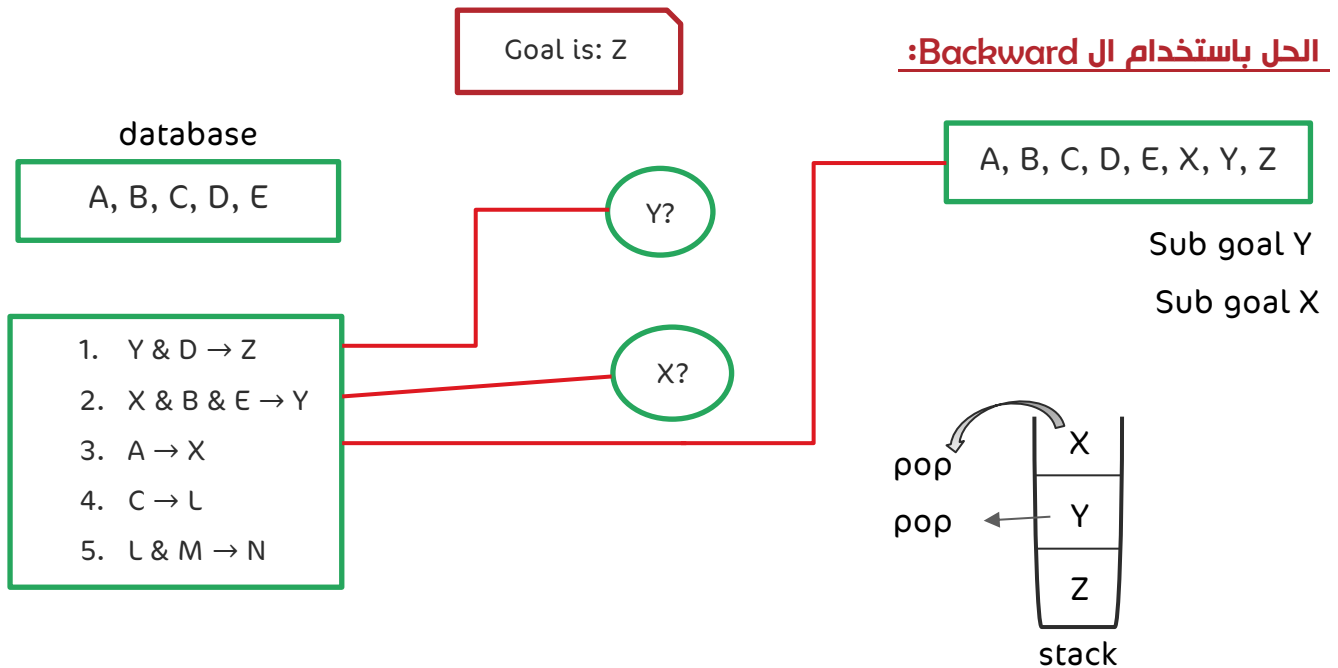
سنتوقف عن الحل هنا إذ أن الهدف المطلوب قد تمت إضافته إلى database.



عند الحل ب forward chaining، نتوقف في حالتين هما إما الوصول إلى ال goal (كمثالنا السابق)، أو عندما تنتهي القواعد لدينا، وإلا فإنها غير قابلة للتنفيذ.

Backward is goal driven reasoning

أي السلسلة الخلفية هي استنتاج مُقاد بالهدف.



شرح الحل: بداية نبحث عن قاعدة طرفها اليميني يحوي على ال goal، وهنا في حالتنا هو Z ، إذا القاعدة الأولى هي المطلوبة نجد أن D موجودة ضمن ال facts على عكس ال Y ، نقوم بإضافة Z إلى المكس ثم إضافة ال Y وهو sub goal، الآن نبحث عن القاعدة التي طرفها اليميني هو Y (القاعدة الثانية) نجد أن E, B موجود ضمن ال fact لكن X غير موجودة، نضيفها على المكس أيضاً أي ال sub goal أصبح X ، الآن نبحث عن كل القواعد التي طرفها اليميني هو X (القاعدة الثالثة هي المطلوبة)، لدينا A موجودة لذلك نقوم بعمل fire لإضافة ال X إلى ال fact ونقوم بعمل pop للمكس لكي يصبح ال Y هو sub goal مجدداً، أي نعود للقاعدة رقم 2 لكي يصبح كل من E, B, X موجود ضمن ال fact، لذلك نقوم بعمل fire لإضافة Y إلى ال facts ونقوم بعمل pop لل Y من المكس، الآن أصبح ال goal هو Z (القاعدة الأولى)، نجد أن Y موجود و D موجود فنقوم بعمل fire لإضافة Z إلى ال fact، وأصبح ال stack فارغاً.

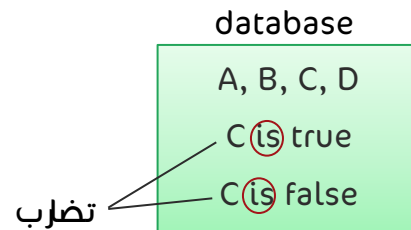
نتيجة:

1. إذا كان هدف الخبير هو جمع المعرفة والحصول على جميع الاستنتاجات من هذه المعرفة، عندها نستخدم العملية الأمامية forward، قد يتم تنفيذ قواعد ليس لها أي دور في النتيجة المراد الوصول إليها.
2. إذا كان الهدف هو إثبات هدف محدد فقط، فعندها يتم الانطلاق من هذا الهدف ومحاولة إثباته (backward).

حل التضارب

R1: if A is true and B is true then C is true.

R2: if A is true and D is true then C is false.



هنا من ال iteration الأولى ينتج لدينا C is true ومن ال iteration الثانية ينتج أن C is false، هنا ينتج لدينا تضارب.

استراتيجيات حل التضارب:

1. إعطاء أولويات للقواعد: عندها عند تحقق قاعدتين معاً، يتم تنفيذ القاعدة ذات الأولوية الأعلى. على سبيل المثال: في ال prolog القاعدة المكتوبة أولاً تكون ذات أولوية أعلى.
2. يمكن عند تحقق قاعدتين معاً أن نقوم بتنفيذ القاعدة الأكثر تخصصاً، مثال:

R1: if A is true and B is true then C is true.

R2: if A is true and B is true and D is true then C is false.

database
A is true
B is true
D is true

حسب الحقائق القاعدتين محققتين، لكن هنا نأخذ القاعدة الأكثر تخصصاً (الأشمل/الأطول) لأن شروطها أكثر وهي R2 لأنها ال Rule الأطول.

من مساوئ ال rule expert system (النظم الخبيرة) أنها غير قابلة للتعلم.



انتهت المحاضرة