

الغاية من الجلسة: التعرف على البنية الأساسية لقواعد المعرفة وكيفية التعامل مع برنامج SWI-Prolog.

مقدمة:

الذكاء الصناعي (AI) هو مجال يرتبط بإنشاء أنظمة كمبيوتر تتيح للأجهزة والبرمجيات أداء المهام التي تتطلب تفكيراً ذكياً. يعتبر الذكاء الصناعي مجالاً مهماً للغاية وله أهمية كبيرة في مجموعة متنوعة من المجالات والقطاعات لقدراته المختلفة في تحسين الأداء والكفاءة وأيضاً التنبؤ والتحليل وكذلك توفير الوقت والموارد بالإضافة إلى تحقيقه السلامة والأمان.

برولوج ومدى أهميته:

برولوج هو لغة برمجة منطقية تُستخدم في مجالات متعددة، وهي جزء هام من مجال الذكاء الصناعي. إليك بعض مجالات استخدام برولوج ومدى أهميته:

1. المنطق والذكاء الصناعي: برولوج تستخدم بشكل شائع في تطوير أنظمة الذكاء الاصطناعي والتعرف على النصوص وأنظمة الاستدلال والمزيد. إنها تتيح التعبير عن المنطق بطريقة تساعد على حل المشكلات المعقدة.
 2. التعلم الآلي والذكاء الاصطناعي: برولوج تستخدم في تطبيقات التعلم الآلي والتعلم العميق حيث تساعد في تمثيل القواعد المعرفية والمنطقية بطريقة تجعل النظم تتعلم وتتكيف بفعالية.
 3. قواعد المعرفة وأنظمة الخبراء: تُستخدم برولوج لتطوير أنظمة قواعد المعرفة وأنظمة الخبراء التي تستند إلى المنطق لاتخاذ قرارات ذكية في مجموعة متنوعة من المجالات.
 4. الأتمتة ومعالجة اللغة الطبيعية: يمكن استخدام برولوج في تطبيقات الأتمتة ومعالجة اللغة الطبيعية لإنشاء نظم تفهم وتوليد اللغة البشرية.
- برولوج هي لغة برمجة قوية لتمثيل المنطق والاستدلال وتطوير نظم الذكاء الاصطناعي. يمكن استخدامها في حل مشاكل معقدة وتطبيقات متنوعة، مما يجعلها مهمة في مجال الذكاء الاصطناعي وتطوير التطبيقات الذكية.
- اللغة Prolog هي لغة برمجة تستند إلى المنطق وتتميز بالنقاط التالية:

1. برمجة بناءً على المنطق (Programming with Logic): تعتمد Prolog على مفهوم المنطق وقواعد المنطق للبرمجة. تتيح البرمجة بالاستدلال اللوجيكي حلاً للمشكلات من خلال تحديد العلاقات والقواعد اللوجيكية بدلاً من توجيه الحاسوب خطوة بخطوة.
2. لغة إعلانية (Declarative): هي لغة إعلانية، مما يعني أنه يتم تحديد ما يجب فعله وليس كيفية فعله. يمكن للمبرمج توصيف المشكلة والعلاقات بين العناصر بشكل منطقي دون الحاجة إلى تفصيل كيفية تنفيذ البرنامج.
3. مختلفة جداً عن لغات البرمجة الأخرى الإجرائية (Procedural): تختلف بشكل كبير عن لغات البرمجة الإجرائية مثل C++ أو Java. في Prolog، يتم التركيز على تصفية البيانات وتحليلها بناءً على القواعد والاستدلال اللوجيكي بدلاً من تحديد تسلسل الخطوات.

4. مناسبة للمهام الغنية بالمعرفة (Good for knowledge-rich tasks): تعتبر مناسبة بشكل خاص لمهام تتضمن كميات كبيرة من المعرفة والمنطق. يمكن استخدامها بشكل فعال في تطبيقات تتعامل مع قواعد البيانات اللوجيكية واستخدامها لاستخراج المعلومات واتخاذ القرارات بناءً على البيانات المتاحة.

باختصار، Prolog هي لغة برمجة تميزت بقواعد المنطق والبرمجة الإعلامية وهي مفيدة بشكل خاص في مجالات تحتاج إلى معالجة المعرفة والمنطق بشكل أساسي.

آلية العمل باختصار:

بالبرولوج (Prolog)، يتم تنفيذ العمليات التالية:

1. وصف المسألة التي تهتمك (Describe the situation of interest): يتعين عليك أولاً وصف الوضع أو السياق الذي ترغب في دراسته أو التحقق منه باستخدام القواعد والحقائق المتاحة في البرنامج.
2. طرح سؤال (Ask a question): بمجرد وصف المسألة بشكل مناسب، يمكنك طرح أسئلة محددة تريد الإجابة عليها باستخدام Prolog. هذه الأسئلة تتعامل مع المعلومات والعلاقات الموجودة في الوضع المهم الذي تم وصفه.
3. الاستدلال اللوجيكي للاستنتاجات (Prolog logically deduces new facts): يستخدم القواعد والحقائق الموجودة في البرنامج للبحث عن إجابات على الأسئلة المطروحة. يقوم بالاستدلال اللوجيكي لاستنتاج المزيد من الحقائق أو المعلومات بناءً على المعرفة المتاحة.
4. تقديم الإجابات (Prolog gives us its deductions back as answers): بمجرد أن يجد Prolog الإجابات أو الاستنتاجات المنطقية، يقدم هذه الإجابات إليك كنتائج. يمكنك استخدام هذه الإجابات لفهم الوضع بشكل أفضل أو اتخاذ الإجراءات الضرورية.

كيف ننشئ قاعدة معرفة ونستعلم منها في البرولوج؟

1. إنشاء ملف بصيغة (.pl): قم بإنشاء ملف نصي جديد واحفظه بصيغة .pl. على سبيل المثال، يمكنك تسميته "family.pl".
2. وضع القواعد في الملف: اكتب القواعد اللوجيكية داخل الملف. مثلاً:

```
human(ahmed).
human(mary).
parent(mary, sara).
parent(ahmed, sara).
```

3. في هذا المثال، تم وضع حقائق لتحديد أن "ahmed" و "mary" هم بشر، وأن "mary" هي أم "sara" و "ahmed" هو أب "sara".
3. حفظ الملف: احفظ الملف بعد كتابة القواعد.

4. استخدام البرولوج للتفاعل: يمكنك تشغيل برنامج البرولوج وتحميل الملف "family.pl" باستخدام الأمر التالي:

```
consult('family.pl').
```

أو يمكن أن نحمل الملف (قاعدة المعرفة) من خيار consult في file.

5. استفسار البرولوج: الآن يمكنك طرح الاستفسارات باستخدام البرولوج. مثلاً:

```
-? human(ahmed).
```

سيعيد البرولوج "true" كإجابة لأن "ahmed" هو إنسان.

```
-? parent(mary, sara).
```

سيعيد البرولوج "true" كإجابة لأن "mary" هي أم "sara".

إليك مثال آخر:

- قاعدة المعرفة الآتية تحتوي على معلومات حول النساء وهل هم يلعبون على الجيتار الهوائي (Air Guitar) أم لا:

```
woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
party.
```

1. `woman(mia)` : هذه الحقيقة تشير إلى أن "mia" هي امرأة.

2. `woman(jody)` : هذه الحقيقة تشير إلى أن "jody" هي امرأة.

3. `woman(yolanda)` : هذه الحقيقة تشير إلى أن "yolanda" هي امرأة.

4. `playsAirGuitar(jody)` : هذه الحقيقة تشير إلى أن "jody" تلعب على الجيتار الهوائي.

5. `party` : هذه الحقيقة تشير إلى أن هناك حفلة (party) تجري. إن وجود النقطة في نهاية العبارة يشير إلى حقيقة وجود الحفلة.

الآن، يمكننا طرح استعلامات على هذه القاعدة لاستخراج المعلومات أو الاستدلال على علاقات. إليك بعض الأمثلة على الاستعلامات:

1. "هل mia امرأة؟"

الاستعلام هو: woman(mia) - ?

الإجابة المتوقعة: "نعم."

2. "هل jody امرأة؟"

الاستعلام هو: woman(jody) - ?

الإجابة المتوقعة: "نعم."

3. "هل yolanda امرأة؟"

الاستعلام هو: woman(yolanda) - ?

الإجابة المتوقعة: "نعم."

4. "هل jody تلعب على الجيتار الهوائي؟"

الاستعلام هو: playsAirGuitar(jody) - ?

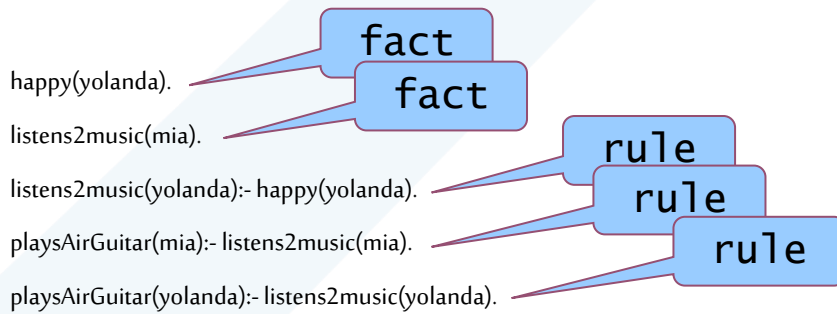
الإجابة المتوقعة: "نعم."

5. "هل هناك حفلة (party)؟"

الاستعلام هو: party - ?

الإجابة المتوقعة: "نعم."

الشكل العام لقواعد المعرفة هو حقائق facts وقواعد rules وتتميز القواعد بأنها تتكون (بدءاً من اليسار) من رأس يليه :- ثم الجسم ومعناه أن الرأس محقق في حال فقط تحقق الجسم.



There are five clauses in this knowledge base:

two facts, and three rules.

The end of a clause is marked with a full stop.

happy(yolanda).
listens2music(mia).
listens2music(yolanda):- happy(yolanda).
playsAirGuitar(mia):- listens2music(mia).
playsAirGuitar(yolanda):- listens2music(yolanda).

There are three predicates in this knowledge base:

happy, listens2music, and playsAirGuitar

نقصد بال predicates أي إسناديات أو functors أي الاسم خارج القوسين وغالباً ما يكون وصف للوسيط داخل القوسين أو وصف للعلاقة بين الوسطاء.

مثال آخر:

happy(vincent).
listens2music(butch).
playsAirGuitar(vincent):- listens2music(vincent), happy(vincent).
playsAirGuitar(butch):- happy(butch).
playsAirGuitar(butch):- listens2music(butch).

القاعدة الأولى في المثال السابق تقول أن:

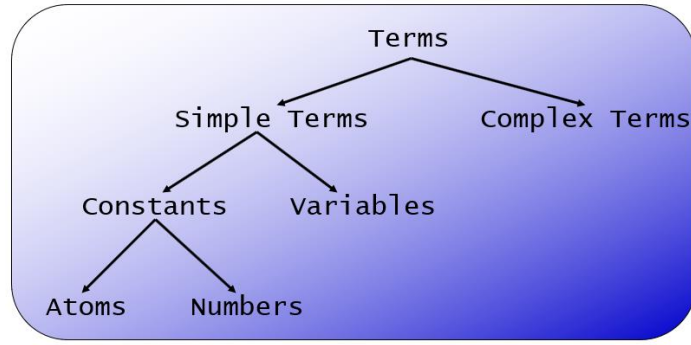
إن "vincent" يلعب على الجيتار الهوائي إذا كان يستمع إلى الموسيقى وهو سعيد. بمعنى آخر، للعب على الجيتار الهوائي، يجب على "vincent" أن يستمتع بالموسيقى وأن يكون سعيداً.

أي الفاصلة تعني و and بينما الفاصلة المنقوطة فتعني أو or.

هنا مثال على ال or:

playsAirGuitar(butch):- happy(butch); listens2music(butch).

أي أن "butch" يلعب على الجيتار الهوائي إذا كان يستمع إلى الموسيقى أو هو سعيد.



الذرات atoms:

1. متسلسلة من الأحرف (A sequence of characters): هذا يشمل أحرف كبيرة وصغيرة وأرقام وشرطة سفلية (.), ويجب أن تبدأ بحرف صغير. أمثلة على ذلك هي: butch, big_kahuna_burger, playGuitar.
2. تكوينية متغيرة (An arbitrary sequence of characters enclosed in single quotes): يمكن تمثيل أي سلسلة من الأحرف أو الأرقام أو الرموز داخل علامات اقتباس واحدة (') كقيمة نصية. مثل: '%\$@', 'Vincent', 'Five dollar shake'.
3. متسلسلة من الرموز الخاصة (A sequence of special characters): هذا يشمل الرموز الخاصة مثل ":", ";", ",", ".", "-", ":", وغيرها. تُستخدم هذه الرموز لتحديد هيكل الجمل والقواعد في برمجة البرولوج.

الأعداد، أمثلة:

- Integers: 12, -34, 22342
- Floats: 34573.3234

المتحولات variables:

- يمكن أن يبدأ هذا المصطلح بإما:
- حرف كبير.
- شرطة سفلية (.).

مثل:

- X, Y, MyVariable (حيث تبدأ بحرف كبير).
- _private_variable (حيث تبدأ بشرطة سفلية).

المصطلحات المعقدة (Complex Terms): المصطلحات المعقدة تُبنى من خلال الجمع بين مفهوم ال Functor ومتغيرات. يتم تكوينها من خلال وجود مصطلح رئيسي يُسمى Functor متبوعاً بسلسلة من الوسائط (Arguments). المصطلحات المعقدة تُستخدم لتمثيل بيانات معقدة وهي أساسية في البرولوج. عادة، يتم توجيه الوسائط بين أقواس مستديرة ويتم فصلها بفواصل. ال Functor يجب أن يكون عبارة عن أصل (Atom)، أي الاسم خارج القوسين يجب أن يكون atom، أما الوطاء قممك أن تكون أي نوع.

مثال: father(john, mike) هنا "father" هو Functor و john, mike هي الوسائط (Arguments) والتي تمثل الأب "john" للابن "mike".

مثال آخر: hide(X,father(father(father(butch))))).

أما عدد الوطاء فيسمى arity ففي مثال ال father(john, mike) فإن $arity=2$ ويعبر عنها بالشكل: father/2 أما مثال ال hide فأيضاً $arity=2$ أي hide/2.