

Intro to AI and Prolog

عملي مشترك



29/03/2021

RB Informatics;

مبادئ الذكاء الصناعي

محتويات المحاضرة

تعريف الذكاء الصناعي

مجالات الذكاء الصناعي

تحقيق الذكاء الصناعي من خلال تقنيات عديدة

Artificial intelligence Applications

Prolog and use it in make Expert System

سيتم توزيع علامة القسم العملي 30 بالشكل التالي:

18 للتقييمات و الوظائف خلال الفصل الدراسي.

12 للمشروع الأخير.



الذكاء الصناعي:

هو بناء آلة أو سوفت ويبر تسلك سلوك وتصرفات البشر وذلك من أجل استهدافها للقيام بخدمة معينة.
حيث تحاكي:

السلوك البشري:

- ✓ تفكير
- ✓ تنبؤ بناء على معطيات معينة
- ✓ إيجاد أفضل حل
- ✓ إدراك لأمر معين

حواس البشر:

- ✓ تحدث
- ✓ إحساس
- ✓ رؤية
- ✓ سماع

حيث هذه الإجراءات لا يمكن للبرمجيات الاعتيادية القيام بها بل يجب أن يكون لدينا برمجيات لديها هذا الذكاء الصناعي الذي يحاكي ذكاء البشر وتصرفاته إلى حد معين.

ملاحظة:

يمكننا تطبيق الذكاء الصناعي على قسم الهارد ويبر أو السوفت ويبر ولكن سيكون اهتمامنا على قسم السوفت ويبر ضمن دراستنا الأكاديمية.

البرمجيات الذكية:

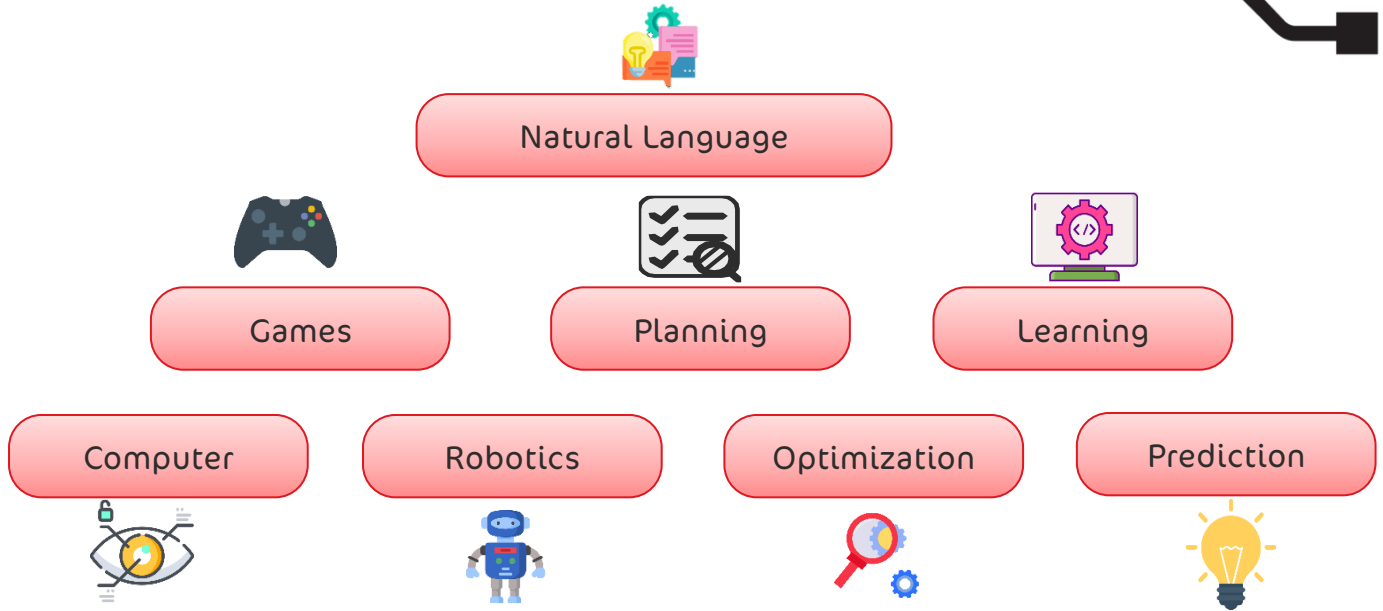
مثال:

تحديد وجود قلم في المشهد .

لجعل برمجية (software) ذكية ليس من خلال مجموعة تعليمات برمجية اعتيادية فقط، بل يجب علينا دراسة مجال (Domain) بكامل التفاصيل و المعطيات (Data) والخصائص التابعة له من معرفة كيفية تحويل هذه البرمجية التي تمتلك خوارزميات وتعليمات برمجية إلى برمجية ذكية تستطيع استخدام هذا الذكاء المحاكي للبشر في خدمة معينة كتحديد قلم في مشهد حيث الداتا هنا عبارة عن صور القلم .

مجالات الذكاء الصناعي Domains :

لدينا عدة Domains ضمن الذكاء الصناعي أهمها:



علم الرؤية الحاسوبية : Computer Vision

هذا العلم يهتم باستخراج المعلومات من الصور (كاستخراج الأشياء Objects أو استخراج إنسان، حيوان، غرض معين،.... الخ) والداتا في هذا المجال هي صور حيث يقوم هذا المجال بتفصيل و معالجة الصور بأسلوب يحاكي البشر بتعرفه على الصور وإدراك المشاهد.

علم معالجة اللغات الطبيعية (NLP (Natural Language Processing

الداتا في هذا القسم عبارة عن نصوص ويقوم هذا المجال بإستخراج القواعد والمعلومات والترابطات المعنوية بين مفردات هذه اللغة و ذلك من أجل القيام بعمليات معينة عليها كاستخدامها في الترجمة. من أشهر التطبيقات التي تستعمل هذا المجال Google Translate.

:Robotics

هو علم يهتم بمحاكاة حركة المفاصل.

:Predication

مجموعة من المنهجيات والخوارزميات التي تحاكي البشر في إنشاء نظام Predication.

:Optimization

هو علم يهتم **يكيفية** إيجاد أفضل حل و **ليس** إيجاد أفضل حل , أي إيجاد المنهجية في كيفية اتخاذ الطريقة التي تؤدي إلى أفضل حل.

:Gaming

بناء ألعاب ذكية قدر الإمكان من خلال بناء عدو ضمن الألعاب يحاكي تصرفات ذكية في اتخاذ القرارات والإستراتيجيات ضد عدوه الإنسان .

■ إثراء :

■ OpenAI في مجال تطوير نظام لعب ذكي:

<https://youtu.be/Lu56xVLZ40M>

■ قدرة الذكاء الصناعي في هزيمة أفضل اللاعبين ضمن إحدى الألعاب:

<https://youtu.be/eaBYhLttETw>

تحقيق الذكاء الصناعي في تطوير برمجيات ذكية:

يجب بناء نموذج رياضي model وهذا النموذج الرياضي يقوم بأخذ إدخال معين ليقوم بإخراج معين اعتماداً على هذا الإدخال.

وهنا تكمن وظيفة الذكاء الصناعي ببناء هذا النموذج، ولكي يقوم الذكاء الصناعي ببناء هذه النماذج فهو يعتمد على مجموعة من التقنيات التي تساعده على بناءها ضمن المجال المستهدف في الاستفادة منها

■ ومن هذه التقنيات :

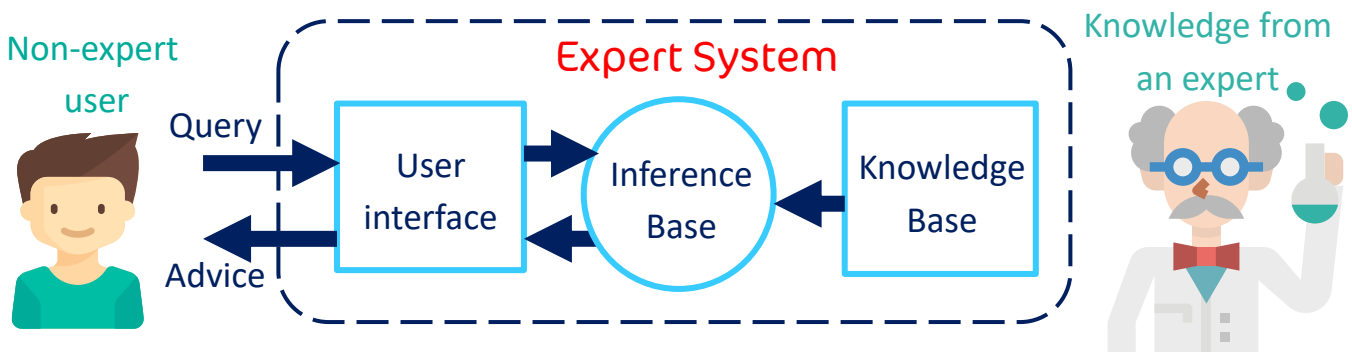
- 1- Knowledge Base System/Rule Base System
- 2- Fuzzy Logic
- 3- Machine Learning
- 4- Intelligent Search Algorithms

1. Knowledge Base System/Rule Base System:

حيث لدي ضمن مجال معين خبرة معينة من الخبير ، هذه الخبرة قابلة للتوصيف بقواعد من الشكل (If...else...then) ضمن Inference Engine حيث يقوم بإظهار النتائج إن وجدت.

مثال : نظام (التنبؤ) بالمرض بناءً على أعراض معينة.

الإدخال هو الأعراض والإخراج هو المرض، لبناء هذا النظام نقوم بتوصيف خبرة الخبير (الدكتور) حيث يخبرنا بخبرته (يخبرنا بكل مرض يعرفه وأعراضه) و تحويلها إلى قواعد ضمن النظام Inference Engine حيث يقوم المستخدم أو المريض بإدخال الأعراض ويقوم النظام بإعطائه المرض ضمن المنطق الكلاسيكي.



2. Fuzzy Logic:

تكون اللغة الطبيعية موصفة أكثر ويكون المصطلح ضبابي (غير مفهوم) بالنسبة للحاسوب لذلك نقوم بتعريف هذا المصطلح ضمن مجال معين على عكس التعريف السابق الذي يعتمد المصطلح على وجوده أو عدمه (True or False).
مثال:

إذا كنا نريد قول أن السيارة سريعة فنقول أن سرعتها تجاوزت الـ 150 لذلك نقوم بتعريف كلمة سريعة ضمن الحاسوب على أنها مجال السرعة فوق 150 كيلو متر بالساعة.

3. Machine Learning:

مجموعة من الخوارزميات والمنهجيات الهدف منها أن تأخذ داتا لمشاهدة يمكننا التعرف عليها , هذه المشاهدة قد تكون صورة، نص، صوت، و قد تكون من أجل غرض ما أو لأكثر من غرض.
حيث يقوم مبدؤه بتعليم الآلة التنبؤ بسلوك أو معادلة ما من خلال تعليم البرمجية بواسطة خوارزمية ما نقوم بإعطائها مجموعة من المشاهدات التي تنتمي للحدث أو الشيء المراد التعرف عليه.

مثال:

OCR (Optical Character Recognition)

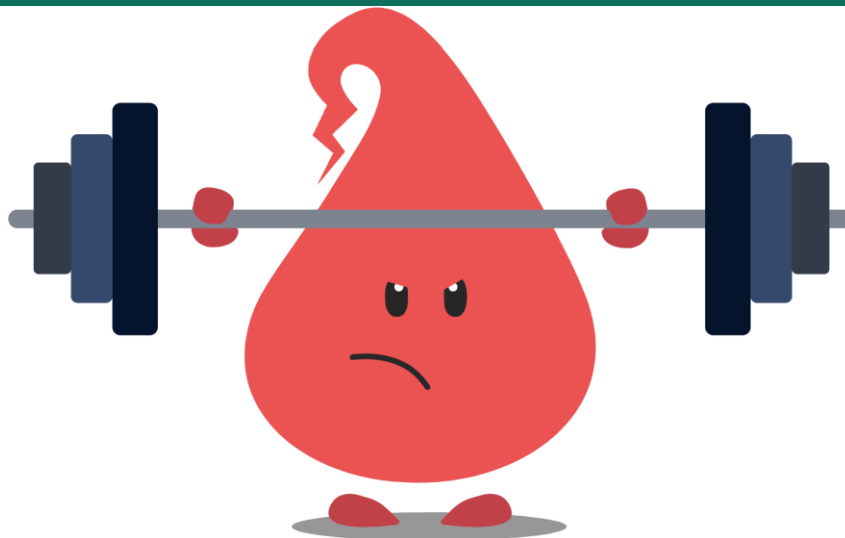
تقوم باستخراج نص من صورة، حيث إذا أردنا إخراج رقم معين من الصور نقوم بتعليمها هذا الرقم حيث نقوم بإعطاء النظام مجموعة مشاهدات لهذا الرقم من أجل التعلم على أن هذا المشهد هو مشهد ينتمي إلى الرقم المعطى بهدف اكتشاف هذا الرقم في الصور المراد استخراج هذا الرقم منها.

يوجد تخصصات فرعية مثل الـ classification, clustering, deep learning

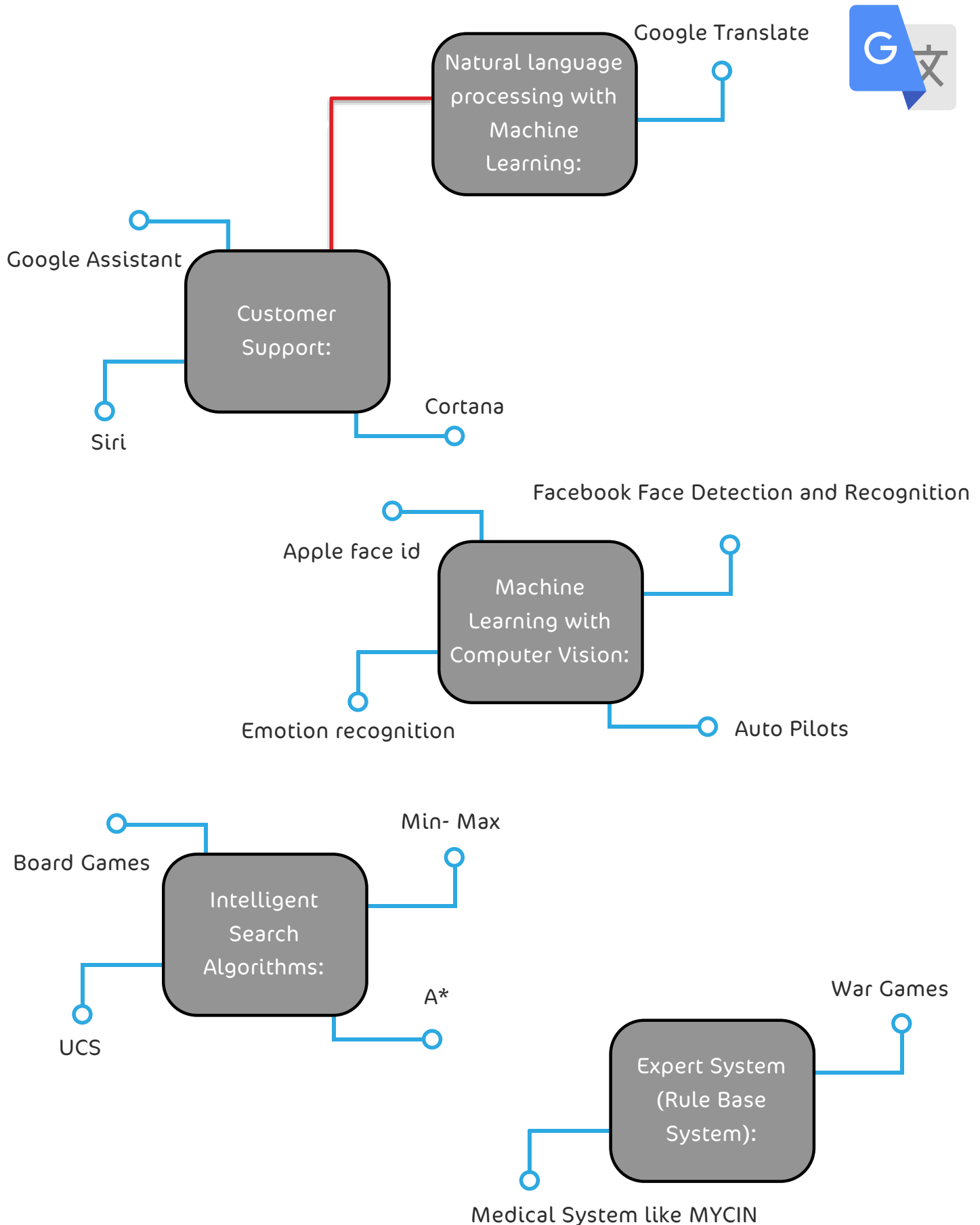
و يوجد ما يسمى بالـ transfer learning ضمن ML وهو طريقة تجعل من البرنامج الذي قمنا بعمله اعتماداً على مشاهد سابقة قادراً على الاستمرار بالتعلم عن طريق تطوير الشبكة العصبونية من خلال تدريب البرنامج الموجود على مشاهدة جديدة.

■ اثناء عن الشبكة العصبونية :

<https://youtu.be/rEDzUT3ymw4>



Artificial intelligence Applications



Prolog language:

هي لغة تستخدم لتوصيف خبرة الخبير ضمن مجال معين و التي قابلة للتوصيف على شكل Rules وهذه الـ Rules تكون من الشكل if...else...then
حيث prolog هي لغة تصريحية (declarative language) تمتلك ما يدعى inference engine
حيث الـ inference engine هي عبارة عن منهجية تحاول أن تحاكي طريقة الاستدلال للعقل البشري في استخلاص النتائج
حيث تعتمد على مفهوم Knowledge Base system في توصيف هذه الخبرة (Rules).
و سنتناول في دراستنا العمل على برنامج SWI Prolog.



SWI Prolog

رابط تحميل البرنامج: <https://www.swi-prolog.org/download/stable>

لكتابة برنامج منطقي ضمن prolog يجب أن يحوي على:

1. Rules
2. Facts
3. Variable
4. Querys

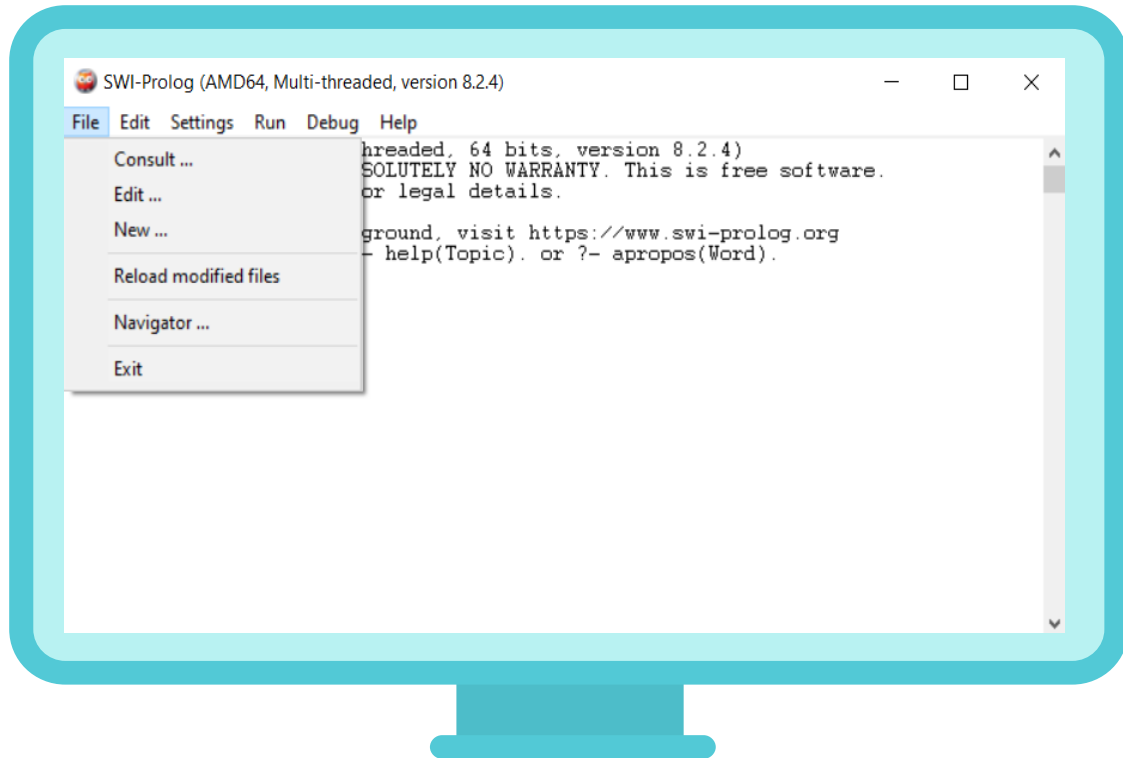
Rules: وهي القاعدة التي تحتوي على متغيرات مثال: father(X,Y).
Facts: وهي الحقائق التي تكتب بشكل صريح مثال: parent(omar,ahmed).



تشغيل SWI-Prolog

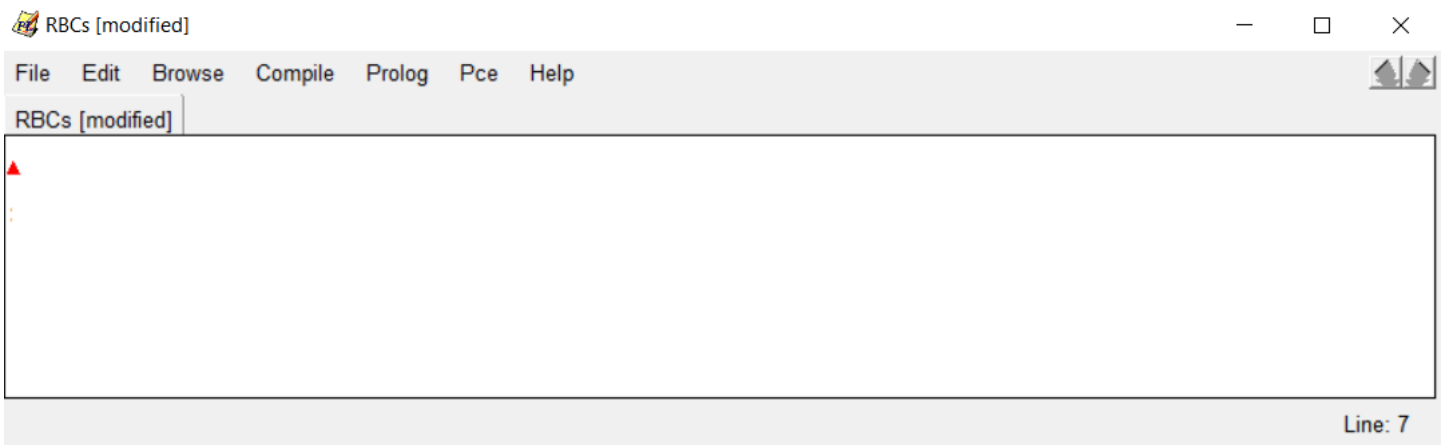
1. تظهر لدينا واجهة الاستعلامات التالية نقوم بما يلي :

File->new



2. ثم نقوم بتسمية الملف الذي نريد الكتابة عليه.

3. ثم تظهر لدينا واجهة ثانية هي واجهة الحقائق والقواعد:



و هنا نقوم بكتابة الـ facts والـ rules.



Syntax:

- ✓ Facts: first letter is small
- ✓ Rules: first letter is small
- ✓ Dot end of rules and facts
- ✓ Constants: first letter is small
- ✓ Variable first letter is big
- ✓ Logic Operations :
 - And ,
 - Or ;
 - Not =\
 - When :-



مجموعة من
قواعد اللغة في
كتابة المعطيات:

- يكون لدي **Syntax error** عندما يظهر اللون الأورنج.
- عندما تكون fact باللون الأسود فهذا يعني أنه قد تم استخدامها.
- عندما تكون fact باللون الأحمر فهذا يعني أنه لم يتم استخدامها بعد.

```

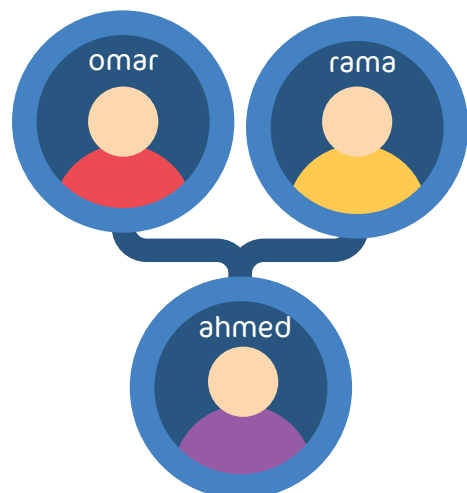
RBCs [modified]
File Edit Browse Compile Prolog Pce Help
RBCs [modified]
parent(omar, ahmed) .
parent(rama, ahmed) .
male(omar) .
female(rama)
father(X, Y) :- parent(X, Y), male(X) .
    
```

■ مثال:

➤ نقوم بكتابة ما يلي ضمن واجهة القواعد والحقائق:

```

parent(omar,ahmed).
parent(rama,ahmed).
male(omar).
female(rama).
father(X,Y):-parent(X,Y),male(X).
    
```





➤ ثم نقوم بـ:

File -> save buffer

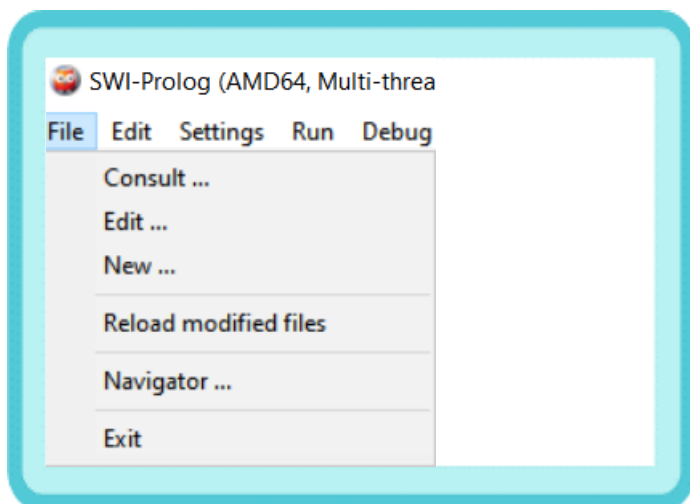
أو عوضاً عن ذلك نضغط :

Ctrl+s

➤ ثم أما نقوم بالذهاب إلى واجهة الاستعلامات (الواجهة الأولى):

File -> Consult ...

و ثم نقوم باختيار الملف الذي قمنا بحفظه:



أو نقوم من واجهة القواعد والحقائق بـ Compile->Compile buffer

أو نستخدم الاختصار Ctrl+c ثم نضغط Ctrl+b

تظهر في واجهة الاستعلامات جملة بالأخضر ف هذا يعني أن الحقائق مكتوبة بشكل صحيح ولا يوجد **syntax error**

أما إذا ظهر جملة **حمر** فهذا يعني أنه يوجد **syntax error**

ملاحظة:

إذا قمنا بأي تعديل على الحقائق نحتاج لحفظ الملف من جديد ومن ثم القيام مجدداً بـ Consult



بعد القيام بربط الملف المكتوب به الحقائق نقوم بالاستعلام عن الطلبات:

← إما أن تكون الاستعلامات صريحة بثوابت فتعيد true or false

← أو بمتغيرات فتعيد أما قيم هذه المتغيرات أو false

```
SWI-Prolog (AMD64, Multi-threaded, version 8.2.4)
File Edit Settings Run Debug Help

?- father(omar,ahmed).
true.

?- father(X,Y).
X = omar,
Y = ahmed ,

?- father(X,Y),female(X).
false.
```

ملاحظة:

👉 المفتاح  في الكيبورد يظهر آخر استعلام قمنا بإدخاله.

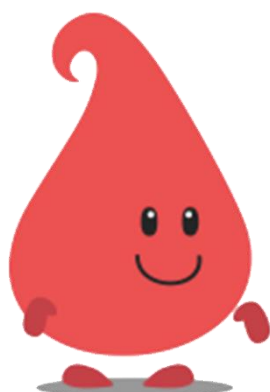
👉 إذا وجد أكثر من حل فإن اللغة تقوم بإظهار الحل الأول بدون أن يكون منهي بنقطة ف لكي نظهر الحل الممكن التالي نضغط ;

مثال:

```
SWI-Prolog (AMD64, Multi-threaded, version 8.2.4)
File Edit Settings Run Debug Help

?- parent(X,Y).
X = omar,
Y = ahmed ;
X = rama,
Y = ahmed.
```

أن مفسر الـ prolog يقوم بالمرور على الحقائق والقواعد forward أي بالترتيب من الأمام واحدة واحدة إلى النهاية فالإجابات التي تستخرج من الاستعلامات تعتمد على ترتيب الحقائق التي قمنا بإدخالها.



انتهت المحاضرة