ماذا نعني بالذكاء الاصطناعي؟

الذكاء الاصطناعي هو عبارة عن برامج الكمبيوتر التي تحاكي طريقة تفكير الانسان. وعلم الذكاء الاصطناعي هو أحد علوم الحاسب الآلي الحديثة التي تبحث عن أساليب متطورة لبرمجة الالة للقيام بأعمال واستنتاجات تشابه ولو في حدود ضيقة تلك الأساليب التي تنسب لذكاء الإنسان ،فهو بذلك علم يبحث أولاً في تعريف الذكاء الإنساني وتحديد أبعاده، ومن ثم محاكاة بعض خواصه .

العلوم التي أثرت في بناء الذكاء الاصطناعي:

- اللغة (الكلام) Linguistic
- علوم الحاسب Computer sciences
- المنطق الحسابي Mathematical Logic
- علم النفس Psychology
- الفلسفة Philosophy
- المعماريات العليا في الحاسب High–level Computer Architecture

أمثلة على تطبيقات الذكاء الاصطناعي:

- فهم اللغات الطبيعية Natural Languages Understanding
- تمييز الصور و الأشكال Image and pattern Recognition
- صناعة الأصوات و تمييزها Voice Synthesis and Recognition
- النظم الخبيرة Expert Systems
- الإنسان الآلي Robotics Activities
- تعلم الماكينة Machine Learning
- ألعاب الحاسب الذكية Computer Intelligent Games

صممت لغات الذكاء الاصطناعي لتناسب الطبيعة البرمجية لهذا المجال الجديد لكي تغطي متطلبات التصميم و البرمجة التالية:-

1) معالجة الرموز و تمثيلها (Symbolic Manipulation) بصورة مباشرة و هي القابلية التي تمكننا من معالجة المسميات بأسلوب رمزي و كما يتم تناوله في اللغات الطبيعية و بعيداً عن التقيد بإطار المعالجة الرقمية التقليدي.

2) سهولة تغيير مسارات التحكم و تبديلها (Control Flexibility) حيث لا يصلح مع نظم الذكاء الاصطناعي لغات البرمجة التي يكون لها مسار تحكم محدد ينفذ عبارة بعد أخرى بشكل تسلسلي. بينما تحتاج نظم الذكاء الاصطناعي إلى تعدد مسارات التحكم مما يسمح بإيجاد كل الحلول الممكنة.

3) سهولة اعتماد أكثر من طريقة لبرامج الذكاء الاصطناعي المهيكلة. حيث أن الوصف المعتمد في أغلب نظم الذكاء الاصطناعي هو وصف ناقص و لا يوجد وصف كامل للمشكلة كما في النظم المبنية على أساس الهيكلة المعروفة بصورتها التدريجية.

وبناءً على المتطلبات السابقة فلقد تم تطوير العديد من لغات الذكاء الاصطناعي و هي مستخدمة على نطاق واسع مثل :

- لغة البر ولوج (PROLOG).
- لغة لسب (LISP).
- لغة (C++).
- لغة جافا (JAVA).

و اللغة الأكثر شيوعاً في عالم الذكاء الاصطناعي هي لغة البر ولوج (PROLOG). و هي تضم إصدارات عديدة منها:

- Turbo PROLOG
- PROLOG 1& 2
- VISUAL PROLOG
- LPA PROLOG
- PD PROLOG

و هي تعتمد على فلسفة منطقية مختلفة عن لغة لسب (LISP). و كلتا اللغتين لبت حاجة المبرمجين في العقود السابقة و لا يزال تأثيرهما فعالاً في مختلف تطبيقات الذكاء الاصطناعي.

اشتقت كلمة (PROLOG) من الكلمات الإنجليزية التالية : البرمجة حسب المنطق

(PROgramming in LOGic) و هي لغة برمجة مخصصة لتوجيه الحواسب للقيام بمهام مختلفة.

ولقد تزامن نمو لغة (PROLOG) مع تطور أنظمة قواعد المعرفة (Knowledge Based System) و هي نظم حاسوبية تتعامل مع كميات كبيرة من المعارف في قاعدة معطيات تعتمد قواعد خاصة. أحد فروع أنظمة قواعد المعرفة هي النظم الخبيرة (Expert System). و تعتبر لغة (PROLOG) ملائمة جداً لبناء النظم الخبيرة و نظم قواعد المعرفة و ذلك لأنها تتألف من مجموعة من القواعد (و أجزاء أخرى تساعد على البحث في قاعدة المعطيات).

و تتميز لغة (PROLOG) بإمكانية استخدامها في العديد من تقنيات البرمجة في الذكاء الاصطناعي (Artificial Intelligence) مثل الشبكات الدلالية و معالجة اللغات الطبيعية. فهي استحقت بجدارة أن تتوج "لغة الذكاء الاصطناعي".

وجدير بالذكر أن استخدام لغة (PROLOG) يتزايد في اللغات التطبيقية و البرامج التجارية و المحاكاة. و هي لغة تفسيرية تستخدم بشكل واسع في أبحاث الذكاء الاصطناعي. و تصمم البرامج من خلالها اعتماداً على مبادئ حل المسائل المنطقية بدلاً من كتابتها بطريقة تعتمد على عمل الحاسب.

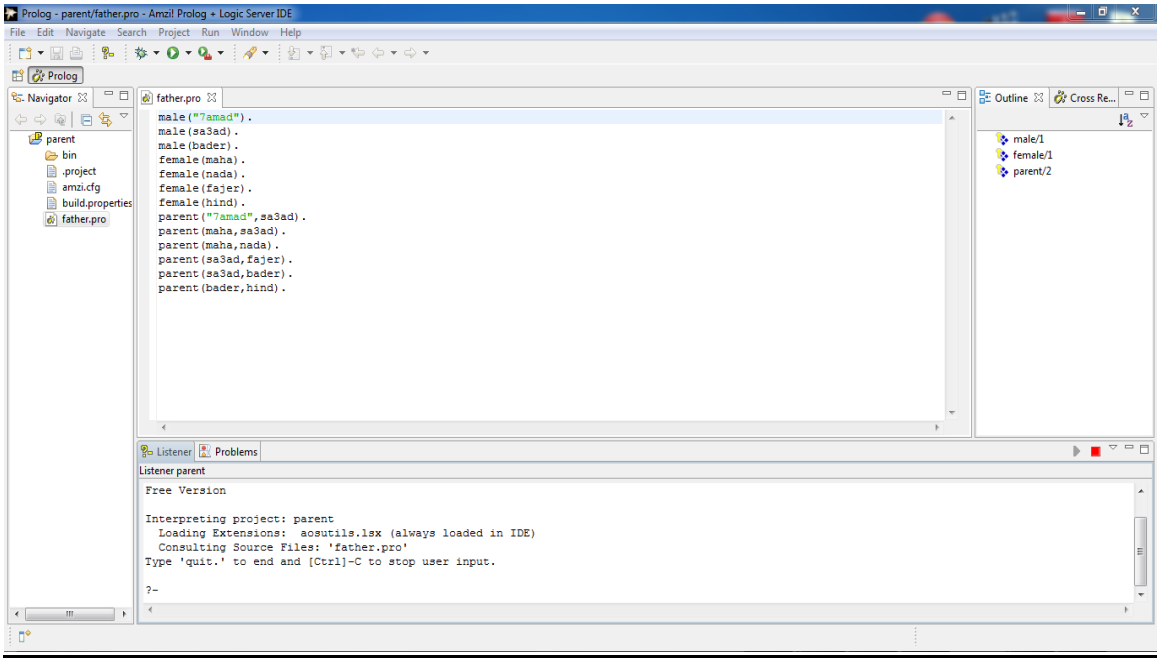
فالبرمجة بلغة (PROLOG) تصريحية تعتمد على التصريح عن المعطيات و الحقائق المتعلقة بالمسألة المعرفة في قاعدة المعطيات. ثم تصمم القواعد لاستخراج المعلومات من قاعدة المعطيات حسب الحاجة. و من هنا يكتشف الباحثون سرعة تميز لغة (PROLOG) عن اللغات الأخرى المعروفة (بعضها لا يلائم الذكاء الاصطناعي) مثل (FORTRAN,BASIC,COBOL,PASCAL,C) و هي جميعها لغات إجرائية.

لغة (PROLOG) تعتمد المفسر (Interpreter) في نقل تعليماتها إلى الحاسب. و مفسر لغة (PROLOG) يختلف حسب الإصدار المستخدم.

البرنامج المستخدم للبرمجة بلغة البرولوج في هذا الفصل هو

Amzi , interactive Development Environment (IDE)

واجهة البرنامج



خطوة البداية تتم من خلال انشاء new project من قائمة file ثم انشاء new file والذي يجب أن يكون امتداده هو pro. ثم كتابة البرنامج في شاشة المحرر (Editor) .

و بعد الانتهاء من كتابة البرنامج يتم تخزين البرنامج من خلال الأمر (Save) الذي يندرج تحت قائمة (File) و التي تشمل العديد من الاوامر الاخرى. ثم تشغيل البرنامج من قائمة Run حيث يتم الانتقال مباشرة الى شاشة الاستفسار والتي يمكن الخروج منها باستخدام الخيار (quit.) .

مكونات برنامج بلغة البرولوج

البرنامج في Prolog هو ما يطلق عليه قاعدة المعرفة Knowledge base . ويتكون البرنامج في لغة البرولوج من حقائق وقواعد (facts + rules). وهناك ثلاث نقاط رئيسية يجب على المتعلم معرفتها

تعريف العلاقات بواسطة الحقائق (facts).

تعريف العلاقات بواسطة القواعد (rules).

كيف يجيب الprolog على الأسئلة والاستفسارات (Query).

هيكل البرنامج بلغة البرولوج

يحتوي هيكل برنامج (PROLOG) على ثلاثة أجزاء رئيسية :

- **الجزء الخاص بالتصريحات (Declarations)**
- **الوصف المعرفي للمشكلة (Clauses)**
- **الجزء الخاص بالاستفسار (Goal)**

أما الجزء الخاص بالتصريحات (Declarations) فانه يحوي حقل لتصريح العلاقات الثابتة (predicate) . بينما الجزء الخاص بالوصف المعرفي فتسجل فيه كل الحقائق و القواعد المعروفة عن المشكلة و التي تحوي على بدائل مختلفة و تكتب بشكل متتابع . أما جزء الاستفسار فمن خلاله يتم السؤال عن الحلول الممكنة للمشكلة الموصوفة.

العلاقات الثابتة (predicate):

يمكن التصريح عن العلاقات الثابتة (المعلنات أو الأسانيد) في بداية البرنامج. يبدأ هذا الحقل بكتابة (predicates) و بدون أي إضافات ثم يتم التصريح عن العلاقات الثابتة (والتي تشارك في وصف المشكلة) في السطر التالي مباشرة كما يلي :

<predicate_ name> (P1, P2, …, Pn)

حيث تمثل Pi المعاملات (arguments) وهي تسمى أيضاً (الحجج أو الوسطاء). و هي التي تؤثر عليها العلاقة (أو المعلِن).

لذا فإن التعبير الإعلاني (المعلِن) من أهم مصطلحات لغة البر ولوج (PROLOG) و هو يأتي متبوعاً (على الأغلب كما نلاحظ) بعدد من المعاملات (الوسطاء أو الحجج) مفصولة عن بعضها بفاصلة و موضوعة بين قوسين.

و أسماء المعلنات يمكن أن تتكون من مجموعة من الحروف و الأرقام. و يلاحظ هنا أن الأسماء الثابتة يجب أن تبدأ بأحرف صغيرة لأن الأحرف الكبيرة لها مدلول آخر (تدل على المتغيرات) في لغة (PROLOG). كما تعامل الشرطة السفلية (_) كحرف و غالباً ما تستخدم لتوضيح معاني الأسماء أو دمج معنيين ليستخدما معاً كاسم واحد. و كمثال على عدد من المعلنات المقبولة:

run

pi
parent(X, Y)
city(london,uk)

و تمثل هذه المعلنات بدورها حقائق عن المشكلات التي تواجهنا و نبحث لها عن حل. و هي تأخذ أشكالاً مختلفة حسب المشاكل العملية و طرق الحل المقترحة.

تأخذ المعلنات و الوسطاء أسماء كيفية و يفضل استخدام الأسماء التي يسهل تذكرها أثناء تصميم البرنامج و خلال عمله. و عند اختيار أسماء المعلنات أو معاملاتها ينصح بإتباع الإرشادات التالية:

- ☒ **تستخدم الأسماء الإنجليزية الواضحة (أو ما يكتب منها بحروف إنجليزية). إذا تطلب اسم المعلن (او معامله) أكثر من كلمة فمن الممكن الفصل بينها بشرطة سفلية (خط سفلي) مثل (day_of _week) أو بدون فاصل مثل (dayofweek).**
- ☒ **يفضل اختيار الأسماء التي تصف وظيفتها بدقة. كأن يختار (day_of _month) لوصف أي يوم في الشهر مثل (march_1_1999).**
- ☒ **من الأفضل تجنب الأسماء التي تحمل أكثر من معنى. فمثلاً إذا كان القائد (commander) يُدعى (salim) و إحدى السفن (ship) تدعى أيضاً (salim) فعليه يجب التمييز بينهما بإضافة حرف ما إلى اسم القائد ليصبح مثلاً (s_salim).**
- ☒ **الحذر من استخدام الأرقام إلا إذا كانت تعابيراً حسابية أو متغيرات مثل (x2, x1).**
- ☒ **تستخدم الأسماء المختزلة عند الضرورة فقط. و ذلك تجنباً لأي لبس بين الأسماء نظراً لتعدد الاختزالات في برامج الذكاء الاصطناعي.**
- ☒ **تعطى المعلنات أسماء أكثر شمولية من أسماء المعاملات (الوسطاء).**
- ☒ **يفضل استخدام الاسم حسب المعنى الذي يدل عليه كأن يستخدم الاسم (car) للسيارة بدلاً من (c أو x).**
- ☒ **ينصح بعدم استخدام الكلمات المحجوزة من قبل لغة (PROLOG) بتاتاً في موقع المعلنات أو المتغيرات (لأنها معلنات خاصة باللغة). و منها ما هو مبين في الجدول التالي:**

الكلمات المحجوزة من قبل مفسر (PROLOG)

and	clauses	Findall	If	Predicates

asserta	database	Free	Include	Readterm
assertz	domains	Global	Not	Retract
bound	Fail	Goal	Or	

حقل الوصف المعرفي للمشكلة (Clauses) :

في حقل الوصف المعرفي للمشكلة (Clauses) يتم إدراج كل الحقائق و القواعد التي يتم وصف المشكلة من خلالها. فمثلاً نكتب العلاقة الوالدية بين حمد و سعد (كمثال على حقيقة) كما يلي:

parent("7hamad", sa3ad).

تعتبر النقطة في نهاية العبارة مهمة لقواعد لغة (PROLOG).

ويمكن أن تكتب قاعدة التعرف على علاقة الأجداد (كقاعدة) كما يلي:

grandparent (A, C):- parent(A, B) and parent(B, C).

بعض الكلمات داخل القواعد يمكن اختصارها باستخدام الرموز التالية:

كلمة	الرمز المقابل
if	:-
and	,
or	;

نموذج لبرنامج مبسط بلغة (PROLOG)

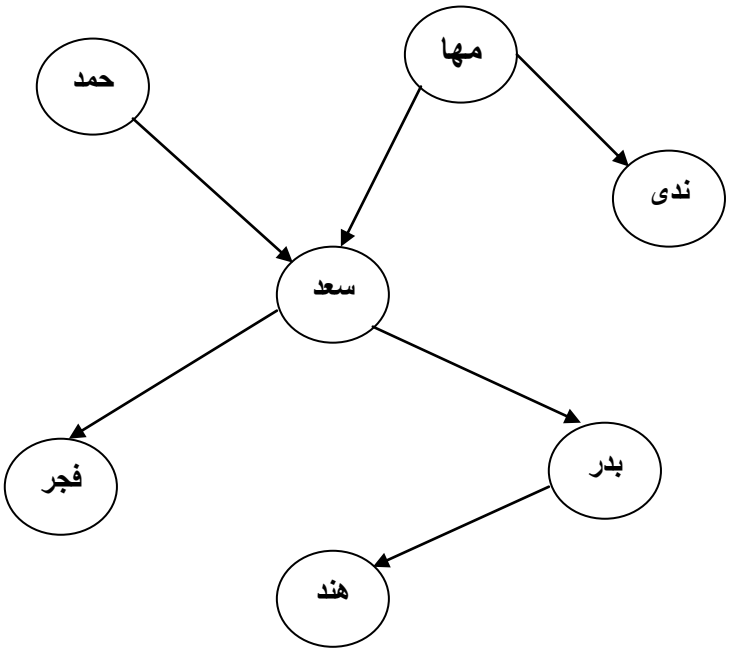
في هذا البرنامج نعرض كيفية تمثيل العلاقات الأسرية باستخدام برنامج (amze).

فمن المعروف أن لغة (PROLOG) هي عبارة عن لغة برمجة للمعالجة الرمزية و غير الرقمية. و

هذه اللغة مناسبة جداً و بصفة خاصة لحل المسائل المتعلقة بالعناصر و العلاقات التي توجد بين هذه العناصر. الشكل (1) يوضح مثال على العلاقة الأسرية أو العائلية "علاقة الوالدية". فالحقيقة أن "حمد" يعتبر "والداً" أو (أصلاً) " لسعد" و يمكن كتابتها بلغة (PROLOG) كما يلي:

```
parent("7amad", sa3ad).
```

حيث اختيرت (parent) كاسم للعلاقة (relation) بينما "حمد" و "سعد" عبارة عن معاملاتها أو "حججها" (argument).



شكل (1) : العلاقة الوالدية (parent) و معاملاتها

والبرنامج التالي يعرف لنا كل عناصر شجرة العائلة طبقاً للشكل (1) مع علاقة تحديد الجنس.

Clauses

```
male("7amad").
male(sa3ad).
```

```
male(bader).
female(maha).
female(nada).
female(fajer).
female(hind).
parent("7amad",sa3ad).
parent(maha,sa3ad).
parent(maha,nada).
parent(sa3ad,fajer).
parent(sa3ad,bader).
parent(bader,hind).
grandparent (A, C):- parent(A, B) ,  parent(B, C) .
```

الوصف المعرفي يتألف من عدد من العبارات (clauses). و كل عبارة من العبارات السابقة تعلن عن حقيقة واحدة. بينما العبارة الأخيرة في الوصف المعرفي هي عبارة عن قاعدة "علاقة الأجداد" و هي ليست حقيقة. و على سبيل المثال العبارة :

parent("7amad", sa3ad).

تعتبر مثالاً خاصاً من العلاقة الأصلية حيث يطلق عليها أيضاً "علاقة" (relationship).

و بصفة عامة فإنه يمكن تعريف العلاقة (relation) كما يلي :

"هي عبارة عن المجموعة (set) و المكونة من جميع أمثلتها (instance)"

و عند ترجمة هذا البرنامج بصورة صحيحة فإن شاشة الاستفسار تكون جاهزة لما يطرح من أسئلة عن العلاقة الأصلية (parent relation) ، مثل : هل حمد والد سعد ؟

Goal:- parent("7amad", sa3ad).

و سوف يجيب البرنامج كما يلي : yes و ذلك بعد التأكد من صحة الإجابة.

ويمكن أن يسأل السؤال التالي:

? – parent(sami, sa3ad)

و سوف يجيب البرنامج كما يلي : no و ذلك لأن البرنامج لم يذكر شيئاً عن أبوة سامي ل (سعد). ويمكن أيضاً الاستفسار عن أشياء أخرى مثل (من هو والد سعد ؟):

? – parent(X, sa3ad)

و سوف يجيب البرنامج عن قيمة المتغير (X) بحيث تكون الجملة السابقة (صواب) و ليس بمجرد القول (نعم) أو (لا) و تكون الإجابة كما يلي:

X = "7amad"

ويمكن أن يطرح السؤال : من هم أبناء سعد ؟ كما يلي :

?- parent(sa3ad, X).

في هذه الحالة يجيب البرنامج عن قيمة المتغير (X) بحيث تكون الجملة السابقة (صواب) و نجد أن هناك أكثر من إجابة واحدة ممكنة كما يلي:

X = fajer
X = bader

في هذا الاصدار من لغة (PROLOG)، يتم الحصول على حل واحد فقط ثم الحل الأخر بكتابة فاصلة منقوطة "semicolon".

ويمكن أن يطرح سؤالا أوسع مثل : من هم الاباء ومن الابناء ؟ وبطريقة أخرى " أوجد (X) و (Y) بحيث يكون (X) أباً لـ (Y). و يمكن التعبير عن ذلك بلغة (PROLOG) في شاشة الاستفسار كما يلي :

?- parent(X, Y).

وعليه فإن مترجم لغة (PROLOG) سوف يوجد كل الأزواج و المكونة من (الابن و والده) واحداً تلو الآخر. و سوف تعرض الحلول واحداً تلو الآخر لحظة بلحظة. فيما يلي نموذج الإجابات التي يمكن الحصول عليها اعتماداً على الشكل (1).

```
X = "7amad"
Y = sa3ad ;
```

```
X = maha
Y = sa3ad ;

X = maha
Y = nada ;

X = sa3ad
Y = fajer ;

X = sa3ad
Y = bader ;

X = bader
Y = hind ;
No
```

ويمكن أن يطرح سؤال أكثر تعقيداً للبرنامج البسيط السابق، مثال ذلك : من هو جد هند ؟ و حيث أن البرنامج السابق و الذي تم تبسيطه آنفا، يحتوي على تعريف مباشر "للعلاقة الجدية" في الوصف المعرفي. و هذا الاستفسار يجب أن ينطوي على خطوتين:

الخطوة الأولى: من هو والد هند (نفرض أنه Y)

الخطوة الثانية: من هو والد Y (نفرض أنه X)

و هذا الاستفهام المركب يتم كتابته بلغة (PROLOG) على هيئة متتابعة من استفهامين بسيطين كما يلي:

```
?- parent(X, Y), parent(Y, hind) .
```

وعليه فإن الإجابة سوف تكون كما يلي:

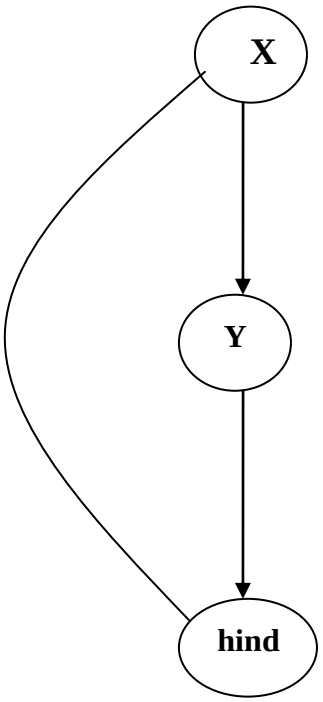
```
X = sa3ad Y=bader
```

و اعتماداً على العلاقة المعرفة في نهاية الوصف المعرفي يمكن كتابة الاستفسار التالي:

?- grandparent (A, hind)

وعليه تكون الإجابة كما يلي (حل واحد فقط):

A = sa3ad



شكل (2) : العلاقة الجدية و المعبر عنها بعلاقتين أصليتين من(parent)

وبطريقة مماثلة يمكن الاستفسار : من هم أحفاد حمد ؟

?- grandparent ("7amad", A)

وعليه تكون الإجابة كما يلي (حلين):

A=fajer
A=bader

الاستعلام عن الحقائق (Querying the Facts):

بعد كتابة برنامج وتصحيحه يقوم مفسر (مترجم) لغة (PROLOG) بمهمة تفسير و تنفيذ الحقائق المكتوبة بلغة (PROLOG). يتم الانتقال إلى شاشة الاستفسار بعد تشغيل البرنامج، حيث تظهر شاشة الاستفسار في حالة خلو البرنامج من الأخطاء اللغوية. و في هذا الإصدار تظهر علامة الاستفهام متبوعة بشرطة عادية (?-) في بداية التعبير معلنة عن الانتظار لأي استفسار.

ايضا من الاستفسارات التي يمكن تطبيقها على البرنامج السابق والاجابة عليها :

☒ **من هم الذكور؟**

```
?- male(X).
X = "7amad" ;

X = sa3ad ;

X = bader ;
No
```

☒ **من هم الاناث ؟**

```
?- female(X).
X = maha ;

X = nada ;

X = fajer ;

X = hind ;
No
```

☒ **هل لسعد أبناء ؟**

```
?- parent(sa3ad,_).
```

```
Yes
```

☒ هل فجر أخت لبدر ؟

```
?- parent(_,fajer),parent(_,bader).
```

```
Yes
```

☒ من من أحفاد حمد هو أب لهند ؟

```
?- parent("7amad",X),parent(X,Y),parent(Y,hind).
```

```
X = sa3ad
Y = bader ;
No
```

☒ من أبناء حمد ؟

```
?- parent("7amad",X).
```

```
X = sa3ad ;
No
```

☒ هل مها أم لندى ؟

```
?- parent(maha,nada).
```

```
Yes
```

☒ من أهل سعد ؟

```
?- parent(X,sa3ad).
```

```
X = "7amad" ;

X = maha ;
No
```

☒ من الوالد الذكر لسعد ؟

```
?- parent(X,sa3ad),male(X).
```

```
X = "7amad" ;
```

```
No
```

☒ من الوالدة الأنثى لسعد ؟

```
?- parent(X,sa3ad),female(X).
```

```
X = maha ;
No
```

☒ من الجد الذكر لبدر ؟

```
?- parent(X,bader),parent(Y,X),male(Y).
```

```
X = sa3ad
Y = "7amad" ;
No
```

☒ من الجدة الأنثى لبدر ؟

```
?- parent(X,bader),parent(Y,X),female(Y).
```

```
X = sa3ad
Y = maha ;
No
```

☒ من عمة هند ؟

```
?- parent(X,hind),parent(Y,X),parent(Y,Z),female(Z).
```

```
X = bader
Y = sa3ad
Z = fajer ;
no
```

للخروج نكتب في شاشة الاستفسار :

```
?- quit.
```

<u>في نهاية مناقشة هذا البرنامج يمكن إبراز بعض النقاط المهمة:</u>

- ✓ من السهولة بمكان على لغة (PROLOG) تعريف "علاقة" ما مثل علاقة الأبوة (relation parent) عن طريق سرد الأشياء (objects) ذات عدد من الأزواج (المعاملات) و التي تحقق هذه المعرفة سابقاً.
- ✓ يستطيع المستخدم بسهولة الاستفسار من لغة (PROLOG) عن العلاقات المعرفة في ثنايا البرنامج.
- ✓ يتألف برنامج لغة (PROLOG) من عبارات (جمل) للوصف المعرفي (clauses) و هي إما حقائق أو قواعد. و كل جملة منها لابد و أن تنتهي بوقفة كاملة (full stop).
- ✓ المعاملات "الحجج" (arguments) الخاصة بالعلاقات يمكن أن تكون أشياء:

1. محددة (concrete) أو ثابتة (constants) (مثل سعد و حمد).
2. أو متغيرة بوجه عام مثل (X) أو (Y).

ويطلق على النوع الأول في البرنامج السابق الذرات (atoms) وهي أصغر نوع . أما الأشياء من النوع الثاني فيطلق عليها المتغيرات (variables).

أنواع المعلنات (Predicates Types):

تشير المعلنات إلى علاقات مختلفة و يمكن إجمالها في الأنواع الأساسية التالية وهي:

- معلن النوع
- معلن الخاصة
- معلن العلاقة
- معلن قاعدة البيانات
- معلن الوظيفة
- معلن الاحتمال

معلن النوع (Type's Predicates):

أمثلة كثيرة على ذلك :

vehicle(car)

أي أن السيارة مركبة.

saudi(oil)

لتحديد نوع "الزيت" بأنه سعودي

day_of_week(monday)

و يوصف به يوم الأثنين على أنه من أيام الأسبوع.

معلن الخاصة (Property Predicates):

حيث يمكن من خلاله إخبار الحاسب الآلي بصفات عامة لشيء محدد مثل:

location(car, west)

تحدد خاصية للسيارة و هو موقعها.

color(book, white)

تبين لون الكتاب و هو الأبيض.

large(u_dammam) or size(u_dammam, large)

توضح أن حجم "جامعة الدمام" هو كبير.

ويفضل أن تمثل خواص الأشياء في تعبير بمعاملين (وسيطين) بحيث أن اسم المعلن هو اسم الخاصية (color, size, location) "لون أو حجم أو موقع" أما المعامل الأول فهو اسم الشيء و المعامل الثاني يوضح الخاصية نفسها.

و هنا تتجلى مميزات استخدام المعلنات الأكثر شمولية. كما أن العلاقة بين المعلن و المعامل الأول يوضحها المعامل الثاني (الذي يأخذ أحد حالات المعلن (مثل كبير لحجم الجامعة و الأبيض للون الكتاب). و المعلن يخبرنا بأنه اسم لخاصية تأخذ القيمة الموضحة في المعامل الثاني بالنسبة للشيء المعطى بالمعامل الأول.

معلن العلاقة (Relation Predicates):

يعتبر معلن العلاقة من أهم المعلنات التي تربط بين أشياء مختلفة. و ذلك لأن المطابقة بين العلاقات المختلفة تؤدي إلى نتائج جديدة كما هو الحال عند الإنسان. فعلى سبيل المثال يمكن استخدام المعلن (part_of) بوسيطين (معاملين) بحيث يدل الوسيط الأول على أنه عنصر من (ينتمي إلى) الوسيط الثاني مثل:

part_of(earth, solar_system)

أي أن الأرض جزء من النظام الشمسي

part_of(ksa, gccc)

أي أن المملكة العربية السعودية دولة من دول مجلس التعاون الخليجي

ويلاحظ أنه إذا وضعنا المعلن بين المعاملين (الوسيطين) فالمعنى يتطابق كما في قواعد اللغة الإنجليزية مثل (ksa part_of gccc) .
ويمكن معاملة العلاقات المختلفة الأخرى مثل:

(east, south, north, space, during, after, left, west,…)

حسب القاعدة المذكورة سابقاً. كما يمكن وصف العلاقات البشرية بنفس الطريقة مثل:

(parent, father, grandfather mother, uncle, …)

كما أنه يوجد معلِن علاقة هام و غالباً ما يستخدم في الذكاء الاصطناعي و هو معلن العلاقة (a_kind_of) و الذي يمكن أن يحل مكان جل المعلنات بحيث يكون معامله (وسيطه) الأول هو شيء أما المعامل الثاني فهو نوع الشيء مثل:

a_kind_of(mango, fruit)
a_kind_of(chair, furniture)
a_kind_of(girls_college, higher_education)

شبكات الدلالة (Semantic Networks):

تعتبر شبكات الدلالة إحدى طرق فهم التعابير الإعلانية و كذلك تستخدم في تمثيل المعرفة. و هي

تستخدم على وجه العموم مع المعلنات ذات الوسيطين. كما يلاحظ أنها تعتمد بدرجة كبيرة على معلنات العلاقة.

و شبكات الدلالة أحد أنواع المخططات. حيث ترمز الدائرة (او المستطيل كخلية) إلى الوسيط (الأول أو الثاني). أما المعلن فيشار إليه بسهم ينطلق من دائرة (مستطيل) الوسيط الأول و يستقر في دائرة (مستطيل) الوسيط الثاني و يعنون السهم باسم المعلن. أي أن التعبير الاعلاني [p(a,b)] يمثّل بسهم ينطلق من الدائرة (a) و يستقر في الدائرة المعنونة (b) أما عنوان السهم ذاته فهو (p).

معلن قاعدة البيانات (Data Base Predicate):

تأخذ المعلنات العدد المرغوب به من الوسطاء بحيث لا تزداد شبكة الدلالة تعقيدا. و بالتالي يمكن أن تحتوي على قيم لخواص متعددة بالإضافة إلى علاقة رابطة. و على سبيل المثال يمكن أن نضع كل ما يتعلق بأي طالب (يدرس في جامعة) في تعبير واحد:

student_info(name, dob, dept, address, city, tel)

ليكون لطالب معين كما يلي:

student_info(bader_akram, "May/3/1999", "computer", "P.O. box 12800", "industrial Jubail", 033469740)

وهذه يمكن قراءتها بكل سهولة (معلومات عن طالب معين). ولكي تفهم هذه الحقائق فهناك حاجة لحفظ مواصفات الوسطاء و صيغها و تسلسلها.

يحدد هذا النوع من المعلنات قاعدة بيانات علائقية للحقائق المعنية. و يعتبر من المعلنات متعددة الوسطاء (المعاملات).

معلن الوظيفة (Function Predicate):

معلن الوظيفة هو صنف آخر من المعلنات متعددة الوسطاء (المعاملات). و هو يستخدم غالباً في التوابع

الرياضية. فالوسيط الأخير يمثل ناتج عملية رياضية ما على بقية الوسطاء. مثال ذلك :

أ) استخدام معلن الجمع و الذي ينص على أن مجموع الوسيطين الأول و الثاني يساوي الوسيط الثالث:

$\mathbf{sum(X,Y, Z) \rightarrow Z = X + Y}$

$\mathbf{sum(2,3, Z) \rightarrow Z = 2 + 3}$

ب) استخدام معلن الضرب و الذي ينص على أن حاصل ضرب الوسيطين الأول و الثاني يساوي الوسيط الثالث:

$\mathbf{mult(X,Y, Z) \rightarrow Z = X * Y}$

$\mathbf{mult(9,11, Z) \rightarrow Z = 9 * 11}$

سوف تستخدم معلنات الوظيفة بشكل مستمر مع اعتبار أن الناتج هو قيمة الوسيط الأخير. مع ملاحظة أن معلن الوظيفة قد يكون متغيراً غير عددي.

معلن الإحتمال (Probability Predicate):

يستخدم هذا المعلن عند عدم التأكد من صحة حقيقة معينة و لكن يفترض صحتها باحتمال معين. حيث يستخدم المفهوم الرياضي للاحتمال و توضع قيمة الاحتمال في الوسيط الأخير داخل التعبير الإعلاني مثل:

color(car, grey, .75).

أي أن السيارة ذات لون رمادي باحتمال يصل إلى 75%.