



الاستدعاء العودي

د. عملي مشترك

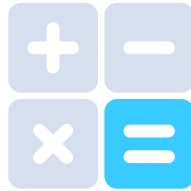
12/03/2019

RB Informatics; مبادئ الذكاء الصناعي

محتويات المحاضرة



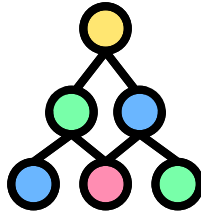
مقدمة

العمليات الرياضية في
برنامج prolog

مسائل



تعريف الاستدعاء العودي



شجرة الاستدعاءات العودية



مستويات الاستدعاء العودي

مقدمة

تكلّمنا في المحاضرة السابقة عن كيفية كتابة برنامج شجرة العائلة بلغة prolog وسنكمل بها في هذه المحاضرة مع الاستدعاء العودي .

الاستدعاء العودي

الاستدعاء العودي يعرف منطقيا في لغة prolog كباقي لغات البرمجة , فهو عملية استدعاء تابع ما لنفسه عدة مرات بوجود شروط وضوابط لهذا التكرار.

مثال (1)

علاقة السلف :

 $\text{ancestor}(X,Y) :- \text{parent}(X,Y).$ $\text{ancestor}(X,Y) :- \text{parent}(X,Z), \text{ancestor}(Z,Y).$ **قاعدة السلف :** وتعني أن X سلف لـ Y أي أن Y هو أب لـ X أو أحد أبائه .

عندما نقوم بأخذ الأب لـ X ونقوم بعمل استدعاء عودي عنده من أجل إيجاد جميع الأجداد الموجودين حتى الوصول إلى الجد الأعلى لـ X وهو Y والذي يعتبر بمثابة شرط توقف الاستدعاء العودي (يتمثل شرط التوقف في السطر الأول) .

ملاحظة

عندما يكون لدينا قاعدتين بنفس الاسم ونفس الـ parameters نستطيع اعتبارهما قاعدة واحدة تفصلهم علاقة or.

فالمثال السابق يكتب بالشكل:

 $\text{ancestor}(X,Y) :- \text{parent}(X,Y) ; \text{parent}(X,Z), \text{ancestor}(Z,Y).$

كيف يتحقق برنامج Prolog من هذه القاعدة للشجرة التالية ؟

 $A \rightarrow B \rightarrow C \rightarrow D$ Goal : $\text{ancestor}(A,D).$ وتعني هل A سلف لـ D ؟

يقوم برنامج الـ prolog باختبار شرط التوقف فوراً وهو يعني هل A سلف لـ D مباشرة فيقوم برد False ثم يقوم بالانتقال إلى الاستدعاء العودي لإيجاد جميع الأجداد الموجودين حتى الوصول إلى الجد D ويتم تحقق شرط التوقف ويرد True .

مستويات الاستدعاء العودي

الاستدعاء الأول:

1

يقوم البرنامج بتبديل Z إلى الخيار الثاني وهو الأب المباشر لـ A أي $B \rightarrow Z$ فيصبح الاستدعاء العودي : $\text{parent}(A,B), \text{ancestor}(B,D)$

يقوم البرنامج بتطبيق هذه القاعدة فيجد أن $\text{parent}(A,B)$ صحيحة ولكن $\text{ancestor}(B,D)$ لم تحقق الشرط فيقوم باستدعاء عودي آخر .

الاستدعاء الثاني:

2

يقوم البرنامج بالاستدعاء العودي للابن المباشر ل B أي $Z \rightarrow C$ فيقوم بتطبيق الاستدعاء العودي التالي :

$\text{parent}(B,C)$, $\text{ancestor}(C,D)$

عندما يقوم البرنامج باختبار القاعدة التالية يجد أن :

$\text{parent}(B,C) \rightarrow \text{True}$

$\text{ancestor}(C,D) \rightarrow \text{True}$

أي يحقق شرط توقف الاستدعاء العودي عند $\text{parent}(C,D)$ بالتالي يتوقف البرنامج ويرد true أي A سلف ل D .

شجرة الاستدعاءات العودية للمثال السابق

$\text{parent}(A,Z)$, $\text{ancestor}(Z,D)$ $Z \rightarrow B$

$\text{parent}(A,B)$, $\text{ancestor}(B,D)$

↓
true

$\text{parent}(B,C)$, $\text{ancestor}(C,D)$ $Z \rightarrow C$

↓
true

$\text{parent}(C,D) \rightarrow \text{True}$

مثال (2)

قاعدة الخلف :

 $\text{des}(X,Y) :- \text{child}(X,Y) ; \text{child}(X,Z) , \text{des}(Z,Y).$ **قاعدة الخلف:** وتعني أن X خلف Y أي أن X حفيد أو أحد أحفاد Y .

لتكن لدينا الشجرة التالية :

 $A \rightarrow B \rightarrow C \rightarrow D$ 

خلف

كيف يقوم برنامج prolog باختبار القاعدة التالية :

Goal : $\text{des}(D,B)$?تعني هل D حفيد أو من أحفاد B ؟

شجرة الاستدعاءات العودية للمثال السابق

 $\text{child}(X,Z) , \text{des}(Z,Y).$ $Z \rightarrow C$ $\text{parent}(D,C) , \text{des}(C,B)$

true
 $\text{parent}(C,A) , \text{des}(A,B)$ $Z \rightarrow A$

true
 $\text{des}(D,B) \rightarrow \text{True}$

ملاحظة

يمكن تعريف متحول عبارة عن (_) فقط .
 مثال : $\text{isfather}(X, _) :- \text{father}(X, Y)$:
 هنا تلقائياً يقوم بتعويض قيمة Y أيما كانت .

مثال (3)

تعريف مستقيمان horizontal و vertical بينهما نقطة مشتركة :

$\text{vertical}(\text{point}(X, Y), \text{point}(X, Y1))$

$\text{horizontal}(\text{point}(X, Y), \text{point}(X1, Y))$

حيث المستقيمان المتقاطعان في نقطة المستقيم العمودي له نفس ال X ويختلف في ال Y أما المستقيم الأفقي فله نفس ال Y ويختلف بال X .

مثال (4)

اكتب قاعدة لإيجاد فيبوناتشي لعدد ما ؟

الحل:

القاعدة العامة:

نعلم أن: $\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2)$ وأن $\text{fib}(1) = 1$, $\text{fib}(0) = 0$

```
fib.pl
File Edit Browse Compile Prolog Pce Help
fib.pl
fib(1,1) .
fib(0,0) .
fib(X,Y) :- X>1, X1 is X-1, X2 is X-2, fib(X1,Y1), fib(X2,Y2), Y is Y1+Y2.
```

حيث $Y1$, $Y2$ متحولات جديدة للتخزين .
 أول سطرين يعبران عن شروط التوقف.

نلاحظ أنه عند الوصول إلى إحدى شرطي التوقف يتوقف الاستدعاء العودي ويعود البرنامج ليأكم قيم المتحولات $Y1, Y2$ في المتحول Y والتي تعبر عن خرج كل استدعاء وصولاً إلى الاستدعاء الأول ونلاحظ أنه من الضروري وضع هذه التعليمة بعد الاستدعاءات العودية.

?- fib(5,Z).
Z = 5 .

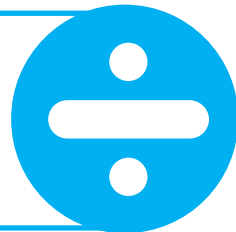
التنفيذ:

العمليات الرياضية في برنامج prolog



القسمة :

$$X \text{ is } \frac{5}{2} \rightarrow X = 2 \text{ (integer)}$$



باقي القسمة :

$$X \text{ is mood } (5,3) \rightarrow X = 1$$



الاسناد :

$$8 = 5 + 3 \rightarrow 8 \text{ is } 5 + 3$$



المقارنات

-1 لا تساوي :

$$= \setminus = \text{ or } = /$$



-2 المساواة :

$$= : = \text{ or } =$$



ملاحظة

(=) , (= /) هي فقط عملية وصول matching وليست مقارنة .

مثال (5)

لتكن لدينا الحقائق التالية :

Facts :

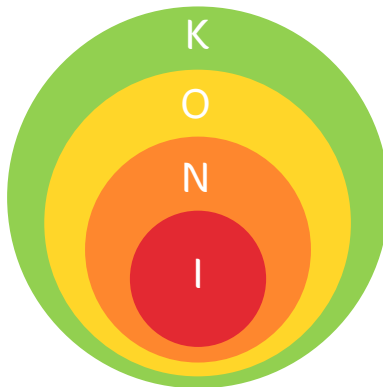
direction (I,N) تعني | مباشرة داخل N

direction (N,O)

direction (O,K)

Goal : in (X,Y)

تعني هل X داخل Y ؟



القاعدة :

in (X,Y) :- direction(X,Y).

in (X,Y) :- direction (X,Z) , in (Z,Y).

اختبر القاعدة التالية :

Goal :- in (I,K) ?

شجرة الاستدعاءات العودية للمثال السابق

direction (Z,K) , in(I,Z). $Z \rightarrow O$

direction (O,K) , in(I,O).

↓
true

direction (N,O) , in(I,N). $Z \rightarrow N$

↓
true

in(I,K) → True

نلاحظ أن N داخل O مباشرة و I داخل N مباشرة بالتالي العلاقة كاملة صحيحة و O داخل k .

مثال (6)

اكتب قاعدة لجمع عددين عودياً ؟

الحل :

القاعدة :

p1.pl [modified]

File Edit Browse Compile Prolog Pce Help

p1.pl [modified]

```
sum(0,Y,Z):-Z is Y.
sum(X,Y,Z):-X>0,Y>0,X1 is X-1,Y1 is Y+1,sum(X1,Y1,Z).
```

التنفيذ:

?- sum(1,2,Z).

Z = 3 .

?- sum(5,10,Z).

Z = 15 .

?- sum(9,1,K).

K = 10 .

- شرط التوقف : اذا كانت قيمة المتحول الأول صفر فإن القيمة النهائية ل Z هي المتحول الآخر غير المعدوم .
- شرط العودي : اذا لم يتحقق شرط التوقف يقوم البرنامج باختبار ايجابية المتحولين ومن ثم يتم انقاص قيمة المتحول الاول بمقدار واحد وجمع واحد للمتحول الثاني واستدعاء القاعدة عند هذه القيمة من جديد حتى يصل إلى شرط التوقف وتصبح قيمة X صفر .

مثال (7)

اكتب قاعدة لإيجاد العامل لعدد ما ؟

الحل:

القاعدة:

$$N! = N-1 * N-2 * N-3 * * 1$$

fac.pl

File Edit Browse Compile Prolog Pce Help

fac.pl

```
factor(0,Z):- Z is 1.
factor(1,Z):- Z is 1.
factor(X,Z):-X>0,X>1,X1 is X-1,factor(X1,Z1),Z is Z1*(X1+1).
```

التنفيذ:

```
?- factor(4,Z).
Z = 24 .

?- factor(2,A).
A = 2 .

?- factor(5,K).
K = 120 .

?- factor(0,Z).
Z = 1 .
```

توضيح القاعدة :

- شرط التوقف : العامل للعديدين صفر و واحد هو 1 .
- شرط العودي : يقوم البرنامج بانقاص قيمة X بمقدار واحد واسناد القيمة الجديدة لمتحول مؤقت واستدعاء التابع من جديد عند هذه القيمة الجديدة مع متغير مرحلي جديد لتخزين النتيجة Z1 ثم يقوم بعد انتهاء الاستدعاءات العودية بإعطاء النتيجة النهائية للبرنامج .

مثال (8)

اكتب قاعدة لإيجاد مجموع الاعداد بين 0 و N؟

$$X = \sum_{i=0}^N i = N + \sum_{i=0}^{N-1} i$$

الحل :

sum(0,0).

sum(N,X) :- N>0 , N1 is N-1 , sum(N1,Y) , X is Y+N.

Lecture(the end) :- ok

