

مقدمة:

Proof Search تشير إلى عملية البحث التي يقوم بها المحرك التنفيذي للعثور على حلاً للاستفسار أو الهدف المطلوب. تتيح لغة Prolog للمستخدم تحديد العلاقات والقواعد، ثم يقوم المحرك بالبحث عن حلول للاستفسارات باستخدام هذه القواعد.

الخوارزمية التي يستخدمها Prolog للبحث عن الحلول تسمى "Depth-First Search" (بحث أولاً في العمق). يتمثل فكرة البحث في استكشاف أولاً فرع واحد من شجرة البحث حتى نهايته، ثم التحول إلى فرع آخر، وهكذا. إذا واجه المحرك مشكلة أو تعارضاً أثناء البحث، يقوم بالتراجع (Backtracking) إلى الخطوة السابقة للبحث عن حل بديل أي يرجع إلى أقرب نقطة خيار.

في Prolog، يتم تحقيق "Proof Search" من خلال محاولة تطابق (unification or matching) الهدف المطلوب مع الحقائق ورؤوس القواعد المتاحة في البرنامج. يبدأ المحرك بمحاولة تطابق الهدف مع حقيقة أو مع رأس قاعدة. إذا كان هناك تطابق، يتم استخدام الجسم (جسم القاعدة) لتوليد أهداف فرعية جديدة. يتابع المحرك هذه العملية حتى يجد حلاً أو ينتهي البحث.

إذا فشل المحرك في العثور على حل أو نجح في إيجاد أول حل من مجموعة حلول، يعود إلى الوراء (Backtrack) للبحث عن حل آخر. يتكرر هذا البرنامج حتى يتم العثور على حلاً أو يتم استكشاف جميع الخيارات الممكنة.

المفهوم الأساسي لـ "Proof Search" في Prolog هو تكامل تحقيق الهدف بتحقيق القواعد والتحقق من الشروط.

مثال أول:

لتكن لدينا قاعدة المعرفة الآتية:

f(a).

f(b).

g(a).

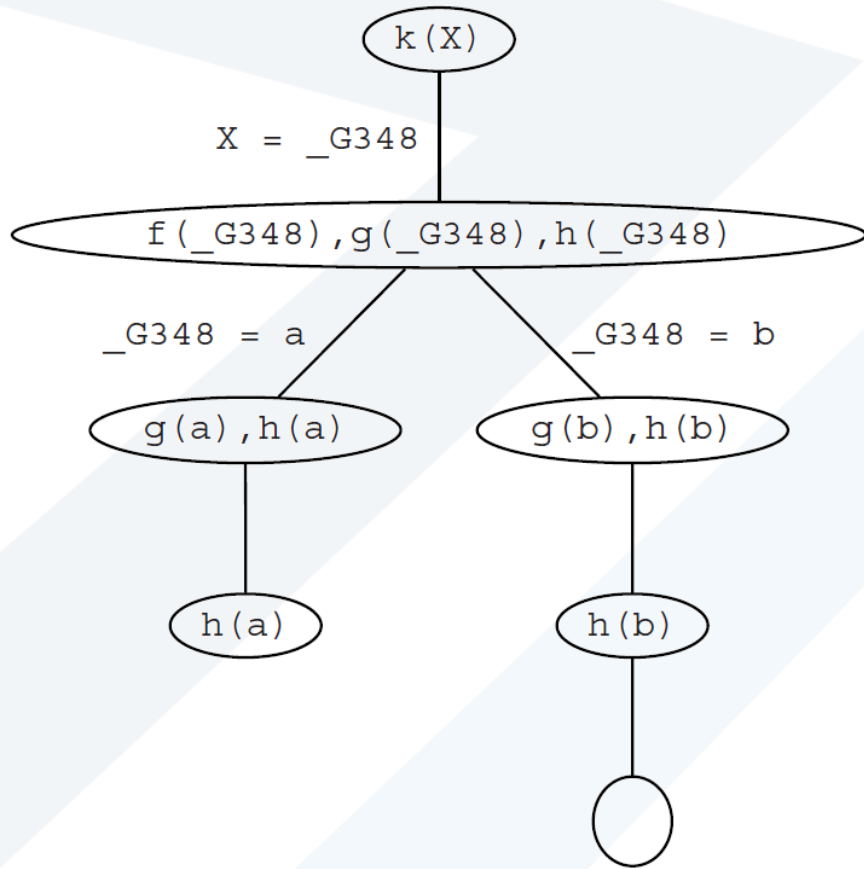
g(b).

h(b).

k(x) :- f(x),g(x),h(x).

ونريد ان نطرح الاستعلام الآتي: k(x).

الشكل الآتي يعبر عن شجرة البحث عن الحل.



ببساطة يقوم البرولوج بوضع متحول خاص به مثل $_G348$ مكان متحولنا X ثم يبحث عن ما يطابق الـ $f(_G348)$ في حقائق ورؤوس القواعد بالتالي يذهب إلى القاعدة ومن أجل أن يثبت أن الرأس محقق فيجب أن يكون الجسم محقق بالتالي نضع ضمن الشكل البيضوي مجموعة الأهداف التي نريد إثبات أنها محققة وهي $f(_G348), g(_G348), h(_G348)$.
يقوم بالتحقق منها هدف هدف.

يذهب إلى الهدف الأول $f(_G348)$ ويخبر إذا كان محققاً أم لا وذلك عن طريق البحث عن مطابق له في حقائق ورؤوس القواعد وعندئذ يجد أنه تحدث المطابقة في الحقيقة الأولى (وأيضاً الحقيقة الثانية ولكنه يؤجل الحقيقة الثانية حالياً) وناتج المطابقة بين هدفنا $f(_G348)$ (نحن الآن في الفرع اليساري) والحقيقة الأولى هو أن $_G348=a$ (لأن المتحول يطابق أي قيمة) بالتالي نحذف الهدف $f(_G348)$ من مجموعة الأهداف ويصبح لدينا فقط هدفان $g(a), h(a)$ حيث لاحظ أن قام بتعويض ناتج الهدف الأول $_G348=a$ في الهدف الثاني والثالث ويتابع في الهدفين المتبقين لكنه يكتشف أن $h(a)$ ليس هناك ما يطابقها بالتالي تبقى من دون مطابقة أي أنها عبارة عن هدف لم يتم تحقيقه بالتالي مجموعة الأهداف لم تصبح فارغة وإنما تحوي على هدف غير محقق $h(a)$ لهذا تعتبر هذه الحالة فشل fail لذلك يقوم الـ prolog بعملية تراجع أو backtracking من أجل أن يجرب الخيارات الأخرى أي أنه يعود إلى أقرب نقطة خيار وبالتالي يعود من أجل أن يطابق $f(_G348)$ مع الحقيقة الثانية (صرنا في الفرع اليميني) بالتالي يكون ناتج المطابقة $_G348=b$ وبالتالي يعوضها في الهدفين المتبقين ويكمل بنفس الطريقة السابقة تماماً لكنه يكتشف أن الهدف هذه المرة كلها محققة لأن $g(b)$ محققة (يطابقها مع الحقيقة $g(b)$) وأيضاً الهدف $h(b)$ أيضاً محقق وبالتالي يتصبح مجموعة الأهداف فارغة (لأن الأهداف تحققت كلها).
تذكر: كلما تحقق هدف قم بإزالته من مجموعة الأهداف ضمن الشكل البيضوي، وعوض ناتج مطابقه في باقي الأهداف.

مثال ثاني:

لدينا قاعدة المعرفة الآتية:

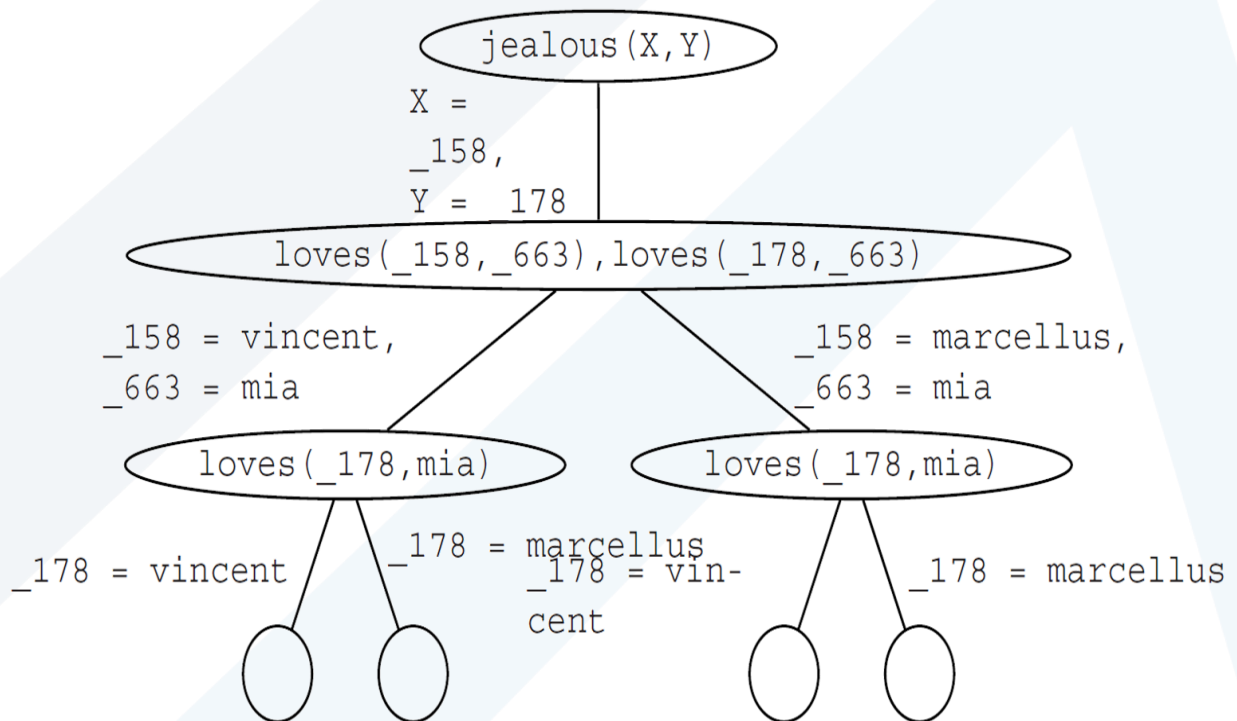
loves(vincent,mia).

loves(marcellus,mia).

jealous(X,Y) :- loves(X,Z),loves(Y,Z).

ولدينا السؤال: من يغار من من؟ jealous(X,Y).

الشكل الآتي يعبر عن شجرة البحث:



ملاحظة: كل رجوع إلى أقرب نقطة خيار أي كل عملية backtracking تعادل الضغط على زر الفاصلة المنقوطة من الاستعلام عن باقي الحلول.

الحلول في المثال السابق هي: vincent يغار من vincent و vincent يغار من marcellus و marcellus يغار من marcellus و marcellus يغار من vincent و marcellus يغار من marcellus.

تمرين خرج:

أوجد الخرج أولاً بدون استخدام البرولوج ثم تأكد باستخدام البرولوج، حيث أننا نستعلم فيما إذا كان عنصرين متطابقين أم لا:

- ?- bread = bread.
- ?- 'Bread' = bread.
- ?- 'bread' = bread.
- ?- Bread = bread.
- ?- bread = sausage.
- ?- food(bread) = bread.
- ?- food(bread) = X.
- ?- food(X) = food(bread).
- ?- food(bread,X) = food(Y,sausage).
- ?- food(bread,X,beer) = food(Y,sausage,X).
- ?- food(bread,X,beer) = food(Y,kahuna_burger).
- ?- food(X) = X.
- ?- meal(food(bread),drink(beer)) = meal(X,Y).
- ?- meal(food(bread),X) = meal(X,drink(beer)).

تمرين:

لدينا قاعدة المعرفة الآتية:

- house_elf(dobby).
- witch(hermione).
- witch('McGonagall').
- witch(rita_skeeter).
- magic(X):-house_elf(X).
- magic(X):-wizard(X).
- magic(X):-witch(X).

أوجد خرج الاستعلامات الآتية وارسم شجرة البحث في كل استعلام:

- ?- magic(house_elf).
- ?- wizard(harry).
- ?- magic(wizard).
- ?- magic('McGonagall').
- ?- magic(Hermione).

انتبه: وسيط الاستعلام الأخير هو متحول.

word(article,a).

word(article,every).

word(noun,criminal).

word(noun,'big kahuna burger').

word(verb,eats).

word(verb,likes).

sentence(Word1,Word2,Word3,Word4,Word5) :-

word(article,Word1),

word(noun,Word2),

word(verb,Word3),

word(article,Word4),

word(noun,Word5).

اكتب الاستعلام من أجل اكتشاف كل الجمل التي يمكن توليدها باستخدام قاعدة المعرفة السابقة.

الحل:

```
?- sentence(Word1,Word2,Word3,Word4,Word5), write(Word1),write(' '), write(Word2),write(' '), write(Word3),write(' '),  
write(Word4),write(' '), write(Word5),nl.
```

حيث ان write هي من أجل الطباعة أما nl فهي من أجل النزول سطر بعد كل حل.

انتهت الجلسة