

Logic Programming

Section #2

Eman Safwat

10/17/2009

Max between Two Numbers:

هنعمل برنامج بيحيب ال max بين رقمين.

لو بالـ C#:

```
max (x, y)
{
    if (x >= y)
        return x;
    else
        return y;
}
```

عاوزين نعمل البرنامج ده بالبرولوج، بس البرولوج مفهش If.. Else Statements فهنعمل Two Rules

بيكملو بعض، فلو عمل Matching مع واحدة ورجعت false الثانية ترجعل ال Max number.

كمان البرولوج مفهش Return Value فال Value بتاعت ال Max number نرجعها ازاي؟

كل Rule هنديها 3 arguments، اتنين Input وواحد Output.

Predicates

nondeterm max (integer, integer, integer).

وحطيت nondeterm عشان هعمل Max 2 فيدخل في الاتنين.

أول Rule:

Clauses

max (X, Y, Z):-

$X \geq Y, Z = X.$

دي كده بتحقق أول جزء من ال if, X, Y ال input دايماً بيكونو Binded. وال Z هترجع فيها قيمة

ال Output هتبقى Free في الأول لحد ما تاخذ قيمة.

هيعمل Comparison بين X و Y لو True (إن X أكبر من Y أو =) هيخش ينفذ اللي بعده $Z = X$.

Assignment, Comparison ال Operator ' = ' في البرولوج بيعمل حاجتين

Assignment لو ال Variables واحد free والتاني binded زي في حالتنا دي (ال Z هتاخذ قيمة ال X)،

أو لو الاتنين free مش هي Assign حاجة يعنى.

و Comparison لو الاتنين binded وفي الحالة دي هيرجع Yes or No.

ال Rule الثانية:

$\max(X, Y, Z)$:-

$$X < Y, Z = Y.$$

هنا نفس الكلام زى اللى فوق، لو ال Condition إتتحق هيدى ال Z قيمة ال Y.

Goal

$\max(5, 7, \text{What})$.

هيعمل Matching مع أول Max.

$$X = 5$$

$$Y = 7$$

$$Z = \text{What}$$

$(X \geq Y)$ مش هتتحقق (هترجع False) مش هيكمل و Backtrack وهيبدا يعمل Matching مع Max الثانية.

$$X = 5$$

$$Y = 7$$

$$Z = \text{What}$$

ملحوظة:

X, Y, Z فى تانى Max دول New Variables غير اللى كانوا فى أول Max (يعنى بيعمل Assignment

لكل Variable من جديد، عشان كده كتبتهم تانى).

$(X < Y)$ هتتحقق و هترجع True فهيكمل وينفذ $Z = Y$.

$$\text{So, } Z = 7$$

What = 7 (final solution).

Second Way to do The Max Program:

عايزين ن Time Decrease البرنامج اللي فات فهنشيل ال Comparison ل Max الثانية.

هتبقى:

max (X, Y, Z) :-

$Z = Y.$

Final Sol.:

What = 7

أول Max متحققتش فنفذ الثانية.

بس لو هيا كده:

Goal

max (7, 5, What).

Final Sol.:

What = 7

What = 5

جواب ال 2 ليه بقى؟

لما دخل في أول Max ال Condition إتحقق فرجع

What = 7

وبعدين Backtrack ودخل Max الثانية ورجع

What = 5

وده طبعاً مش صح، إحنا عاوزينه يشتغل كـ If.. else ويرجع قيمة واحدة!!

هزود حاجة اسمها (!) Cut كل اللي بتعملوه انها بتوقف ال Backtracking عندها فأى حاجة قبلها

مبيتعملهاش Backtracking.

لما أزود ال Cut في أول Max:

Max (X, Y, Z):-

$X \geq Y, !, Z = X.$

Final Sol.:

What = 7 (only)

ايه اللي حصل؟

بعد ما عمل Matching ولقى ال Condition صح وال $Z = 7$ جيه يعمل Backtracking لقي ال Cut
قدامه فوقف ومكملش في Max الثانية.

ال ! دائماً بترجع True فاللي بعدها دائماً بيتنفذ، بحطها أما أكون عاوزاه يحقق كل اللي بعدها وميعملش
Backtrack للي قبلها.

Third Way (Last) to do The Max Program:

max (X, Y, X) :-

$X \geq Y, !.$

ايه اللي حصل هنا؟

بدل ماکانوا ثلاثة Variables خلّتهم اثنين بس، أول X بتاخده ال input (Binded) وتاني X لل
output (Free).

فكده بقي ال Assignment (X input في X output) الأول في بداية ال Rule

وبعدين ال Checking لل Condition ولو رجع True هيلقى ال ! هيقف ومش هيعمل Backtracking
لل Max.

تاني Max:

max (X, Y, Y).

عملت نفس الكلام اللي فوق، أول ما هيدخل في ال Rule دي هي Assign ..Y input في Y output.

لو عملت ال Rule كده هيديني Error (Unused variable X) عشان ال X مش بعمل عليها أي

Body ال Operations (checking, assignment, ..etc) بتاع ال Rule أصلاً فاضي!

فخليها:

max (_, Y, Y).

ال underscore معناها إني مش مهتمة بال Variable ده قيمته كام وكده هيشغل تمام.

Goal

max (5, 7, What).

Final Sol.:

What = 7 (only)

هوا بيخش في Max الأولى ويعمل Assignment وبعدين ي Check لو رجع True هي Cut ويرجع ال X،
ولو ال Condition رجع False هي Backtrack للتانية ويرجع ال Y.. بس كده.

Example:

Predicates

nondeterm car (symbol, symbol, integer).

nondeterm color (symbol, symbol).

nondeterm buy (symbol, integer).

Clauses

car (opel, black, 30000).

car (fiat, green, 20000).

car (mercedes, green, 40000).

car (honda, red, 25000).

color (red, nice).

color (green, nice).

color (black, wonderful).

buy (M, P) :-

car (M, C, P),

color (C, nice),

P <= 25000.

buy (M, P) :-

car (M, C, P),

color (c, wonderful).

Goal

buy (What, Price).

هنا ال Goal عاوز كل ال Cars اللى فيهم Nice وال Price لكل واحدة أقل من أو يساوى 25000، أو

ال Car يكون فيها Wonderful.

هنبداً Tracing:

What = M

Price = P

هبخش فى Buy الأولى: (Matching مع كل ال Facts)

car (M, C, P)

M = opel

C = black

P = 3000

وبعدين ال Sub-goal:

color (C, nice)

Nice مش هت Match.

Backtrack (All variables become free again)

تانى Fact:

M = fiat

C = green

P = 20000

color (C, nice) (Matched)

P <= 25000 (Matched)

First Sol.:

What = fiat, Price = 20000

Backtrack (All variables become free)

M = Mercedes

C = green

P = 40000

color (C, nice) (Matched)

P <= 25000 (not Matched)

Backtrack (All variables become free)

M = honda

C = red

P = 25000

color (C, nice) (Matched)

P <= 25000 (Matched)

Second Sol.:

What = honda, Price = 25000

كده عمل Matching مع كل ال Facts وأول Buy.

نخش على Buy الثانية:

buy (M, P) :-

car (M, C, P), color (C, wonderful).

بعد منعزل Matching مش هنلاقى غير:

What = opel

Price = 30000

عشان مفيش Wonderful غير مع Black ومفيش Black غير مع Opel!

Final Sol.:

What = fiat, Price = 20000

What = honda, Price = 25000

What = opel, Price = 3000

Additional Explanation

هوا ببدا كل مرة من:

car (M, C, P)

و ي Match لحد:

$P \leq 25000$

وبعد ميخلص ال Fact بي Backtrack ويبدأ من أول:

car (M, C, P)

تاني وهكذا..

وأما يخلص كل ال Facts بي Backtrack على Buy ويروح ل Buy الثانية ويكمل.

Notes:

لو جيت غيرت في أول Buy وخليتها كده:

buy (M, P) :-

!,

car (M, C, P),

color (C, nice),

$P \leq 25000$.

ال ! دى هتغير في ايه؟

هتغير إنيو بعد ما يعمل Matching لكل ال Facts مع:

car (M, C, P),

color (C, nice),

$P \leq 25000$.

ويجى يعمل Backtracking عشان يروح لتانى Buy هيلاقى ال ! فـهيفصل ويقف وميروحش لتانى Buy.

فـهتبقى كالاتى:

Final Sol.:

What = fiat, Price = 20000

What = honda, Price = 25000

لو حظيتها كده:

buy (M, P) :-

car (M, C, P),

!,

color (C, nice),

$P \leq 25000$.

أول مرة:

M = opel, C = black, P = 30000

فال black مش هت Match مع Nice، هيعمل Backtracking هيلاقى ! هيقف. ولا هيكمل ال Facts

ولا هيروح ل Buy الثانية.

Final Sol.:

No solution.

لو Fact ال Fiat كانت أول واحدة فى ترتيب ال Facts كانت طلعت Solution، عشان كانت هت Match

مع Nice ومع ($P \leq 25000$) وأما يعمل Backtracking ويقف تكون هيا خلاص طلعت.

ولو جيت خليتها:

buy (M, P) :-

car (M, C, P),

color (C, nice),

!,

P <= 25000.

هنا هيعمل Matching مع Opel ومش هت Match عشان ال Nice فهيعمل Backtracking

من أول:

car (M, C, P)

Fiat ال Fact هت Match مع Nice و (P <= 25000) وترجلى Fiat (ملاحظين عدينا ال ! كده) هيعمل

Backtracking من عند (P <= 25000) هيلاقى ! فهيقف.

Final Sol.:

What = fiat, Price = 20000

ونفس الكلام لو هيا:

buy (M, P) :-

car (M, C, P),

color (C, nice),

P <= 25000, !.

لأنه بعد ميخلص Matching مع (P <= 25000) هيعدى ال Cut فمش هينفع يرجع تانى.

Example:

Predicates

nondeterm father (symbol, symbol).

Clauses

father (ali, ahmed).

father (ali, hossam).

father (ahmed, sami).

Goal

father (X, Y), write (X, " is the father of ", Y). //write() == console.write()

دی کده هتطلع:

ali is the father of ahmed X=ali, Y=ahmed

ali is the father of hossam X=ali, Y=hossam

ahmed is the father of sami X=ahmed, Y=sami

واحنا مش عايزين X=ali, Y=ahmed تطلع..

فهنعمل Rule اسمها Display (مبتختش Arguments)

فترجعلي ali is the father of ahmed بس.

display :-

father (X, Y),

write (X, " is the father of ", Y).

ده كده هيرجع One Solution بس،

وكمنا مشكلة ال Yes or No Rules حاجة، بترجع مع الحل Yes or No فها هترجع

واحنا برضو مش عايزينها تطلع!!

فهنزود حاجة جديدة اسمها "fail" دي بترجع دايماً False (No) فتهخليه بعد كل حل يرجع No فيعمل

Backtrack لحد ما يجيب كل ال Possible Solutions هتبقى كده:

display :-

father (X, Y),

write (X, " is the father of ", Y), fail.

هيطلع كل ال Solutions:

ali is the father of ahmed

ali is the father of hossam

ahmed is the father of sami

no

بعد ما طلع كل ده رجلى No! ودى احنا مش عاوزينها، هنزود حاجة عشان تبقى Yes بدالها.

هنعمل Rule تانية ل Display من غير Body كل فايدتها ترجع Yes بعد ما يعمل Backtracking عليها.

display :-

father (X, Y),

write (X, " is the father of ", Y), fail.

display.

Final Sol.:

ali is the father of ahmed

ali is the father of hossam

ahmed is the father of sami

yes

احنا ليه عملنا ده كله ما كنا نسيبها No وخلاص!؟

عشان لو جيت تستعمل Fail وحطيت بعدها Conditions or statements مش هتتفد!

Fail بعد مبرجع No البرنامج بي Skip اللي بعدها كله (عكس Cut).

Note

ال ',' معناها And زى ما احنا عارفين، كمان ال ';' Semi Colon معناها Or وال , ليها

Precedence أعلى من ; يعنى لو الاتنين موجودين مع بعض ال , بتتفد الأول.

بس احنا مبنستعملش ; كتير عشان بتعمل لغبطة فى البرنامج.

Recursion

عاوزين نعمل Factorial وال Factorial بيتعمل لإما Loop or Recursion والبرولوج مفهانش Loops
فهنعملها Recursion.

$0! = 1$ (Base step)

$n! = n (n-1)!$ (Recursive step)

Predicates

fact (integer, real).

Clauses

fact (0, 1) :- !.

fact (N, F) :-

$N1 = N - 1,$

fact (N1, F1),

$F = N * F1.$

Goal

fact (3, What).

نمسك ال Rules:

fact (0, 1) :- !.

دى ال Base Step كل ما ال Input يكون 0 ال Output يكون 1.

ال Recursive Step بجيبها إنى بضرب الرقم فى Fact الرقم الأصغر بـ 1.

فتبقى كده:

fact (N, F) :-

fact (N-1, F1), $F = N * F1.$

بس البرولوج مش هتفهم fact (N-1, F1) فخلناها كده:

fact (N, F) :-

N1 = N - 1,

fact (N1, F1),

F = N * F1.

هنبداً Tracing:

هوا عاوز..

fact (3, What).

Matching مع أول Fact مش هتدخل، هتخش فى الثانية:

N = 3, F = What

N1 = 2

هتخش فى ال Sub-goal:

fact (N1, F1),

N * F1.  Level 1

fact (2, F1)

Matching مع أول Fact مش هتدخل، هتخش فى الثانية:

N = 2, F = F1

N1 = 1

هتخش فى ال Sub-goal:

fact (N1, F1),

N * F1.  Level 2

fact (1, F1)

Matching مع أول Fact مش هتدخل، هتخش فى الثانية:

N = 1, F1 = F1

N1 = 0

Matching مع أول Fact هتدخل وترجع 1:

fact (0, 1) :- !.

N * F1.  Level 3

هناك النتائج من الآخر للأول:

Level 3:

$$N = 1, F1 = 1$$

$$\text{Product1} = N * F1 = 1 \quad (\text{Product1} == F1 \text{ to the upper-level})$$

Level 2:

$$N = 2, F1 = 1$$

$$\text{Product2} = N * \text{Product1} = 2 \quad (\text{Product2} == F1 \text{ to the upper-level})$$

Level 1:

$$N = 3, F1 = 2$$

$$\text{Product3} = N * \text{Product2} = 6 \quad (\text{Product3} == F)$$

Final Sol.:

$$\text{What} = 6$$

ايه اللي حصل؟

احنا كل مرة بننده على Function لحد منوصل لل Base step بتاعتنا، بعد كده بنحسب كل Function بتاخذ كام وبترجع كام بس بالمقلوب (من آخر Level وصلنا له - بتاع ال Base step - لأول Level اتدهت فيه ال Function لأول مرة).

لكل Level بشوف ال F1 اللي رجعله من ال Level اللي قبله بكام وال N بتاعته أصلن بكام وأضربهم في بعض والنتائج يبقى ال F1 بتاعت ال Level اللي بعده وهكذا..

لحد اما أوصل لآخر Level وال Product بتاعه هيكون ال F الكبيرة (Recursion Output).

Final Note

في ال Base Step:

$$\text{fact}(0, 1) :- !.$$

حطيت ال ! عشان لما يوصل لل Base Step هيعمل Backtracking ويخس في Fact الثانية (N=0).

في N1 = -1.. وبعدين تخس في نفسها تاني وهكذا infinite فال Cut بتمنعه يعمل Backtracking.

Assignment:

Use Recursion to get the power function, which takes 3 arguments:

X: Base

Y: Power

Z: Final Result

$X^0 = 1$ (Base Step)

$X^Y = X * (X^{Y-1})$ (Recursive Step)

Good Luck