

Міністерство освіти і науки України
Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Мультипарадигмненне програмування

ЗВІТ

до лабораторних робіт

Виконав
студент

ПІ-01 Ніколаєв Іван Романович

(№ групи, прізвище, ім'я, по батькові)

Прийняв

ас. Очеретяний О. К.

(посада, прізвище, ім'я, по батькові)

Київ 2022

Завдання :

Exercise 2.1 If a map has N regions, then estimate how many computations may have to be done in order to determine whether or not the coloring is in conflict. Argue using program clause trees.

Дано : мапа налічує N регіонів, підрахуйте максимальну кількість обчислень які необхідно зробити для визначення чи є конфлікт кольорів? Обґрунтуйте вашу відповідь із використанням дерева фактів програмного коду

Exercise 2.2.1 Using the first factorial program, show explicitly that there cannot possibly be a clause tree rooted at 'factorial(5,2)' having all true leaves.

Використовуючи першу програму факторіал покажіть що не існує дерева фактів що має корінь 'factorial(5,2)' що має всі листки true

Exercise 2.2.2 Draw a clause tree for the goal 'factorial(3,1,6)' having all true leaves, in a fashion similar to that done for factorial(3,6) previously. How do the two programs differ with regard to how they compute factorial? Also, trace the goal 'factorial(3,1,6)' using Prolog.

Напишіть дерево фактів для цілі 'factorial(3,1,6)' маючи всі листки true, в такому ж форматі як це зроблено для factorial(3,6) попередньо. Як ці дві програми відрізняються з точки зору підрахунку факторіалу? Також виконайте трасування цілі 'factorial(3,1,6)' використовуючи Prolog

Exercise 2.3.1 Draw a program clause tree for the goal 'move(3,left,right,center)' show that it is a consequence of the program. How is this clause tree related to the substitution process explained above?

Напишіть дерево фактів для цілі 'move(3,left,right,center)' , покажіть що це програмна послідовність.

Яким чином це дерево фактів стосується процесу підстановки описаного вище.

Exercise 2.3.2 Try the Prolog goal ?-move(3,left,right,left). What's wrong? Suggest a way to fix this and follow through to see that the fix works.

Виконайте ціль Prolog move(3,left,right,left). У чому помилка? Запропонуйте шлях вирішення та перевірте що виправлення спрацювало.

Виконання:

Вправа 2.1

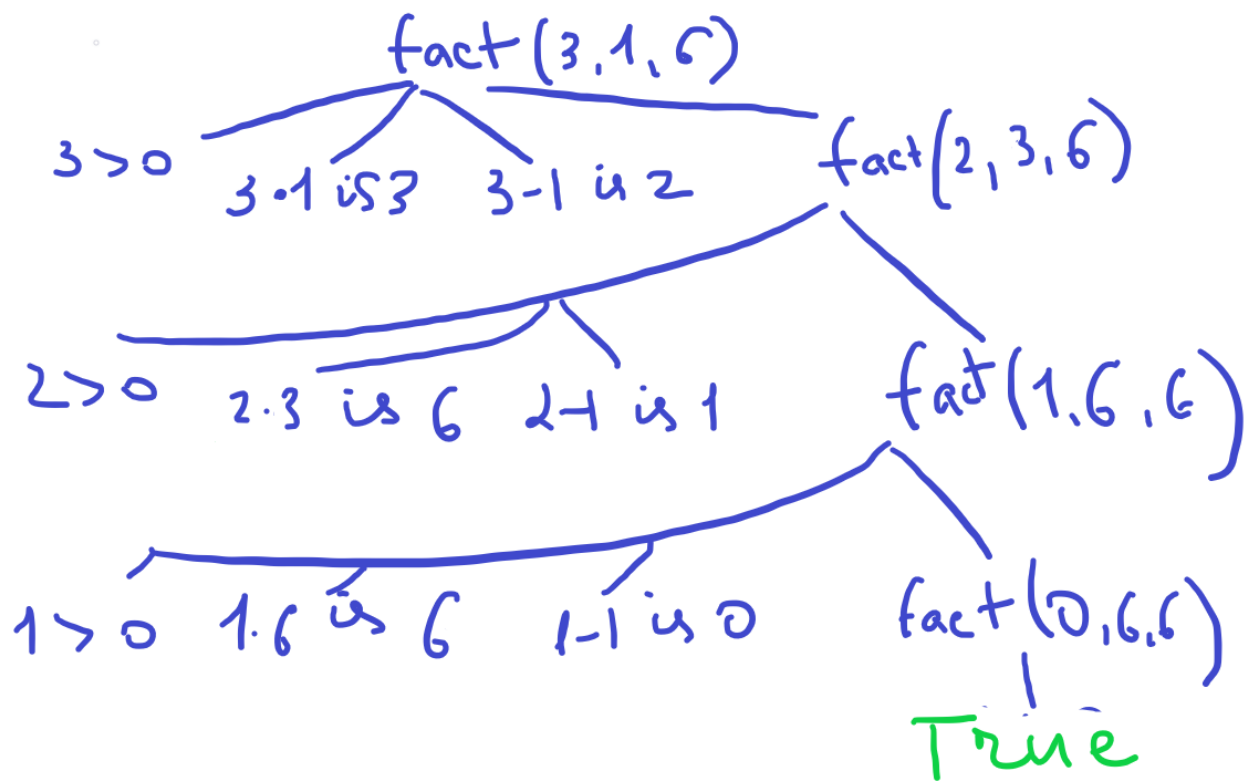
Якщо мапа містить N регіонів, то для кожної пари цих регіонів необхідно виконати наступні перевірки: перевірити, чи є ці регіони сусідніми, і для кожного регіона перевірити, чи має він шуканий колір, тобто складеість таких перевірок складе константу 3. Знайдемо кількість таких пар, а саме за формулою комбінацій: $C_n^2 = \frac{n!}{2!(n-2)!} = \frac{n(n-1)}{2}$. Тоді остаточна складність становитиме: $\frac{3n(n-1)}{2}$.

Вправа 2.2.1

Виконуємо трасування для $\text{factorial}(5, 2)$. Отримали наступний результат:

```
Call: factorial(5,2)
Call: 5>0
Exit: 5>0
Call: _780 is 5 + -1
Exit: 4 is 5 + -1
Call: factorial(4,_782)
Call: 4>0
Exit: 4>0
Call: _790 is 4 + -1
Exit: 3 is 4 + -1
Call: factorial(3,_792)
Call: 3>0
Exit: 3>0
Call: _800 is 3 + -1
Exit: 2 is 3 + -1
Call: factorial(2,_802)
Call: 2>0
Exit: 2>0
Call: _810 is 2 + -1
Exit: 1 is 2 + -1
Call: factorial(1,_812)
Call: 1>0
Exit: 1>0
Call: _820 is 1 + -1
Exit: 0 is 1 + -1
Call: factorial(0,_822)
Exit: factorial(0,1)
Call: _812 is 1*1
Exit: 1 is 1*1
Exit: factorial(1,1)
Call: _802 is 2*1
Exit: 2 is 2*1
Exit: factorial(2,2)
Call: _792 is 3*2
Exit: 6 is 3*2
Exit: factorial(3,6)
Call: _782 is 4*6
Exit: 24 is 4*6
Exit: factorial(4,24)
Call: 2 is 5*24
Fail: 2 is 5*24
```

Бачимо, що хибний результат отримали у листі, в якому перевіряється, чи дорівнює дане число знайденому факторіалу: $2 == 5 * 24$? – отримали тут False, і вся функція повернула False.



Ці дві програми відрізняються наступним чином: перша програма порівнювала даний факторіал із заданим аж біля самого кореня, перед цим рекурсивно заглиблюючись до листків, де проводила необхідні обчислення, натомість друга програма виконувала всі обчислення до заглиблення до рекурсії, та перевіряла заданий факторіал та даний у листку, а наверх повертала вже True або False.

Вправа 2.3.1

Трасування:

Вправа 2.3.2

 `move(3,left,right,left).`
Move top disk from left to right
Move top disk from left to left
Move top disk from right to left
Move top disk from left to right
Move top disk from left to left
Move top disk from left to right
Move top disk from left to right
true

Результат виконання `move(3, left, right, left)`.

Програма відпрацювала без помилок, проте має не дуже прийнятний результат, бо вона сказала передвинути диск з лівої осі до лівої, тобто ми не зможемо відрізнити, яке “ліве” є яким (бо на вході ми даємо цілих два “ліва”), через що розв’язок може бути неточним. Тому необхідно якимось перевіряти, чи різні назви осей дали на вхід. Це можна зробити наступним чином, модифікувавши написану функцію:

```
move(N,X,Y,Z) :-  
    N>1,  
    X\=Y, Y\=Z, X\=Z,  
    M is N-1,  
    move(M,X,Z,Y),  
    move(1,X,Y,_),  
    move(M,Z,Y,X).
```

Просто будемо порівнювати осі. Якщо є однакові, то будемо повертати `False`.

Результат виконання функції після модифікації:

 `move(3,left,right,left).`
false