

ELFE TP3: Résolution - Récursivité

François LEPAN
Benjamin VAN RYSEGHEM

24 janvier 2012

Première partie Généalogie

Question 1.2

22 étapes sont franchies pour obtenir un résultat lors de l'exécution du but `descendant_de(anne, emily)`.

Question 1.3

Nous obtenons 10 solutions :

1. X = anne
Y = bridget
2. X = bridget
Y = caroline
3. X = caroline
Y = donna
4. X = donna
Y = emily
5. X = anne
Y = caroline
6. X = anne
Y = donna
7. X = anne
Y = emily
8. X = bridget
Y = donna
9. X = bridget
Y = emily
10. X = caroline
Y = emily

Question 2.2

Il y a une boucle infinie.

Question 2.3

Nous obtenons 10 solutions :

1. X = anne
Y = emily
2. X = anne
Y = donna
3. X = anne
Y = caroline
4. X = bridget
Y = emily
5. X = bridget
Y = donna
6. X = caroline
Y = emily
7. X = anne
Y = bridget
8. X = bridget
Y = caroline
9. X = caroline
Y = donna
10. X = donna
Y = emily

Question 3.2

Fatal Error: local stack overflow.

Question 3.3

Fatal Error: local stack overflow.

Question 4.2

6 étapes puis :

```
true ? ;
```

Fatal Error: local stack overflow

Question 4.3

Nous obtenons 10 solutions :

1. X = anne
Y = bridget
2. X = bridget
Y = caroline
3. X = caroline
Y = donna
4. X = donna
Y = emily
5. X = anne
Y = caroline
6. X = bridget
Y = donna
7. X = caroline
Y = emily
8. X = anne
Y = donna
9. X = bridget
Y = emily
10. X = anne
Y = emily

puis : Fatal Error: local stack overflow.

Deuxième partie

Entiers de Peano

Question 6.1

```
entier(zero).
entier(succ(X)) :- entier(X).
```

Question 6.2

Il y a une infinité de successeurs de `zero`.

subsection*Question 7.1

```
inf_ou_egal(zero, X) :- entier(X).
inf_ou_egal(succ(X),succ(Y)) :- inf_ou_egal(X,Y).
```

subsection*Question 7.2

1. true
2. false
3.
 - X = zero ;
 - X = succ(zero) ;
 - X = succ(succ(zero)) ;
4.
 - X = succ(zero) ;
 - X = succ(succ(zero)) ;
 - X = succ(succ(succ(zero))) ;
 - X = succ(succ(succ(succ(zero)))) ;
 - ...

Toutes les prédicats à partir de `succ(zero)` satisfont le but `inf_ou_egal(succ(zero),Z)`.

Question 8.1

```
add(zero, X, X) :- entier(X).
add(X, zero, X) :- entier(X).
add(succ(X),succ(Y),succ(succ(Z))) :- add(X,Y,Z).
```

Question 8.2

1. false
2. true
3. R= succ(succ(succ(zero)))
4. Z= succ(succ(zero))
5.
 - X = zero
 - Y = succ(zero)
 - X = succ(zero)
 - Y = succ(succ(zero))
 - X = succ(succ(zero))
 - Y = succ(succ(succ(zero)))
 - ...

Question 8.3

```
sub(X, zero, X) :- entier(X).  
sub(succ(X),succ(Y),Z) :- sub(X,Y,Z).
```

Question 9

```
mult(zero,X,zero) :- entier(X).  
mult(X,zero,zero) :- entier(X).  
mult(succ(X),Y,T) :- mult(X,Y,Z), add(Y,Z,T).  
mult(X,succ(Y),T) :- mult(X,Y,Z), add(X,Z,T).
```