

2 Arbres de dérivation

Exercice 1

Pour définir une fonction $f(X_1, \dots, X_n) = Y$, on définit un prédicat $p(X_1, \dots, X_n, Y)$ qui est vrai si $f(X_1, \dots, X_n) = Y$. Par exemple, pour la fonction $X+1 = Y$, on définit un prédicat `a_pour_successeur(X,Y)`.

1. Comment exprimer qu'on veut calculer la somme de deux nombres ?
2. Comment exprimer qu'on veut connaître le plus grand parmi 2 nombres ? parmi 3 ?
Que produit `max2(X,1,3)` ? Pourquoi ? Même question avec `max2(1,X,3)`.

Exercice 2

Pour chacun des cas suivants, donner la réponse de l'interpréteur Prolog, puis vérifier.

1. `7<9`.
2. `X < 12`.
3. `X = Y+1`.
4. `X is 10+2, Y=X*2`.
5. `Y = X*2, X is 10+2`.
6. `X is Y+1, Y is 4`.
7. `X is 2, Y is X+1, X+Y < 6`.

Exercice 3

1. Afficher les N premiers entiers.
2. Ecrire un prédicat `envers(+X)` qui affiche le nombre X à l'envers. Par exemple `envers(1234)` affichera 4321.
3. Ecrire un prédicat pour trouver la somme des N premiers entiers. (Ou : `somme(+N,?X)` est vrai si X est la somme des entiers de 1 à N).
4. Ecrire un prédicat `fact(+N,?X)` pour trouver la factorielle d'un nombre N .
5. Ecrire un prédicat `fibonacci(+N,?X)` qui est vrai si X est la valeur de la suite de Fibonacci au rang N .
6. L'énoncé suivant spécifie un algorithme récursif bien connu :
Un entier n divise un entier m si $n \leq m$ et si ou bien $n = m$, ou bien n divise $m - n$.
Ecrire un prédicat `div(+diviseur,+divident)` qui implante cet algorithme.

Exercice 4

Pour chacun des cas suivants, donner la réponse de l'interpréteur Prolog, puis vérifier.

1. `n(X,n(X,a,b),f(Y)) = n(Y,n(a,a,b),f(Z))`.
2. `g(X,f(Y,X),A) = g(f(A),A,f(B))`.
3. `g(X,f(Y,X),A) = g(f(A),C,f(B))`.
4. `f(X,X) = f(g(a,Y),g(Z,b))`.
5. `f(g(X,a),Y) = f(Y,g(b,Z))`.
6. `p(Y,X,h(Y)) = p(g(a)+Z,f(V),V)`.
7. `p(X+Z,Y+5) = p(Y+1+Z,2+Z)`.

Exercice 5

Alice aime les pommes. Bruno aime les carottes. Clara aime les oranges. Les pommes et les oranges sont des fruits. Les carottes sont des légumes. Ceux qui aime les fruits sont en bonne santé.

1. Formaliser ces faits et règles en Prolog.
2. Quelle est la requête pour savoir qui est en bonne santé?
3. Quelle est la requête pour “Qui aime les pommes?” ?
4. Comment savoir les fruits que connaît le programme?
5. Donner l’arbre de dérivation pour la requête “Qui est en bonne santé?”.

Exercice 6

Considérons le programme suivant :

```
pere(paul,luc).    pere(guy,aude).    pere(luc,anne).    pere(luc,eric).
mere(rose,luc).    mere(ines,aude).    mere(aude,anne).    mere(aude,eric).
parent(X,Y) :-    pere(X,Y).
parent(X,Y) :-    mere(X,Y).
grand_pere(X,Y) :-    parent(Z,Y), pere(X,Z).
ascendant1(A,P) :-    parent(A,P).
ascendant1(A,P) :-    parent(X,P), ascendant1(A,X).
ascendant2(A,P) :-    parent(A,P).
ascendant2(A,P) :-    ascendant2(A,X), parent(X,P), .
```

Donner l’arbre de dérivation pour les requêtes suivantes :

```
?- grand_pere(X, anne).
?- ascendant1(A, aude).
?- ascendant2(A, aude).
```

Exercice 7

Donner l’arbre de dérivation pour les requêtes suivantes pour les prédicats de l’exercice 3 :

```
?- somme(2,X).
?- fact(3,X).
```