




TP3: PROLOG

Réalisatrice:
Fatma Laribi GL3/2



PLAN

- I. Exercice 1
- II. Exercice 2
- III. Exercice 3

Exercice 1

1. Soit le prédicat `adjacent(Region1,Region2)` permettant de tester si 2 régions sont adjacentes. Définir la carte à l'aide de ce prédicat.
2. Soit `color(Region,Couleur,Coloriage)` qui, pour une certain Coloriage, attribue à chaque région une couleur. Définir à l'aide de ce prédicat, un exemple de `coloriage1` sans conflit, et un autre exemple de `coloriage2` avec conflit.
3. Définir un prédicat `conflit(X,Y,Coloriage)` qui permet de savoir quelles régions adjacentes ont la même couleur.
4. Définir un prédicat `conflit(Coloriage)` qui permet de voir si un coloriage des régions respecte les contraintes que nous nous sommes fixées. Tester sur `coloriage1` et `coloriage2`.

■ Base de faits

```
base4 - Bloc-notes

Fichier  Modifier  Affichage

adjacent(a,b).
adjacent(a,d).
adjacent(a,c).
adjacent(b,e).
adjacent(b,a).
adjacent(b,c).
adjacent(e,d).
adjacent(e,b).
adjacent(e,c).
adjacent(d,a).
adjacent(d,e).
adjacent(d,c).

color(a,bleu,coloriage1).
color(b,vert,coloriage1).
color(d,vert,coloriage1).
color(c,rouge,coloriage1).
color(e,bleu,coloriage1).

color(a,rouge,coloriage2).
color(b,vert,coloriage2).
color(d,vert,coloriage2).
color(c,rouge,coloriage2).

color(e,bleu,coloriage2).

conflit(X,Y,Coloriage) :- adjacent(X,Y), color(X,Z,Coloriage), color(Y,Z,Coloriage).
conflit(Coloriage) :- conflit(_,_,Coloriage).
```

■ Test

```
?- conflit(coloriage1).
false.

?- conflit(coloriage2).
true .
```

Exercice 2

Ecrire un programme Prolog pour calculer les nombres de Fibonacci en utilisant la définition réursive suivante : $U_0 = 1$, $U_1 = 1$, $\forall n > 1 : U_n = U_{n-1} + U_{n-2}$

```
fib(0,1).  
fib(1,1).  
fib(N,X) :- N>1, N1 is N-1, N2 is N-2, fib(N1,X1), fib(N2, X2), X is X1+X2 .
```

■ Test

```
?- fib(6,X).  
X = 13 ,
```

Exercice 3

Ecrire un programme Prolog qui lit N entiers de l'utilisateur et affiche leur maximum ainsi que leur somme

```
boucle(0,0,0).  
boucle(N,S,MAX):- N>0, write('saisir entier '),nl, read(X) , A is N-1, boucle(A,B,MAX2), S is X+B , MAX is max(X,MAX2).
```

■ Test

```
?- boucle(3,S,MAX).  
saisir entier  
|: 15.  
saisir entier  
|: 6.  
saisir entier  
|: 4.  
  
S = 25,  
MAX = 15 .
```

Merci pour votre
attention!