Rapport Travaux Pratiques : Programmation par Contraintes - TP 4 : **Contraintes Logiques**

Nicolas Desfeux
Aurélien Texier

30 mars 2011

Dans ce T.P., nous allons utiliser la programmation par contraintes pour faire un planning pour organiser une régate, planning qui respecte certaines contraintes.

Question 4.1 Nous définissons ici un prédicat *getData(?TailleEquipes, ?NbEquipes, ?CapaBateaux, ?NbBateaux, ?* qui unifie les variables passées en paramètres avec les données du problème.

Listing 1 – "getData"

```
getData(TailleEquipes,NbEquipes,CapaBateaux,NbBateaux,NbConf):-
        TailleEquipes = [](5,5,2,1),
        NbEquipes = 4,
        CapaBateaux = [](7,6,5),
        NbBateaux = 3,
        NbConf = 3.

/* Tests
[eclipse 7]: getData(TailleEq,NbEq,CapaBat,NbBat,NbConf).

TailleEq = [](5, 5, 2, 1)
NbEq = 4
CapaBat = [](7, 6, 5)
NbBat = 3
NbConf = 3
Yes (0.00s cpu)
*/
```

Question 4.2 Nous définissons ici un prédicat *defineVars(?T,+NbEquipes,+NbConf,+NbBateaux)* qui unifie T au tableau des variables et contraint le domaine des variables.

Listing 2 – "defineVars"

```
defineVars(T,NbEquipes,NbConf,NbBateaux):-
```

```
        dim(T,[NbEquipes,NbConf]),
        ( for(Ind1,1,NbEquipes),param(T,NbBateaux,NbConf)
        do
                ( for(Ind2,1,NbConf),param(T,Ind1,NbBateaux)
                do
                        T[Ind1,Ind2] #:: 1..NbBateaux
                )
        ).

/* Tests
[eclipse 8]: getData(TailleEq,NbEq,CapaBat,NbBat,NbConf),defineVars(T,NbEq,NbConf,NbBat).

TailleEq = [](5, 5, 2, 1)
NbEq = 4
CapaBat = [](7, 6, 5)
NbBat = 3
NbConf = 3
T = []([](_419{1 .. 3}, _488{1 .. 3}, _557{1 .. 3}), [](_628{1 .. 3}, _697{1 .. 3}, _766{1 .. 3}), [](_837{1 .. 3}, _906{1 .. 3}, _975{1 .. 3}), [](_1046{1 .. 3}, _1115{1 .. 3}, _1184{1 .. 3}))
Yes (0.00s cpu)
*/
```

Question 4.3 Nous définissons ici un prédicat *getVarList(+T, ?L)* qui construit la liste L des variables contenues dans le tableau T. La liste des variables contient les variables de la première colonne suivies de celles de la seconde colonne, etc.

Listing 3 – "getVarList"

```
getVarList(T,L):-
        dim(T,[NbEquipes,NbConf]),
        ( for(Indice1,1,NbConf),fromto([],In,Out,L),param(T,NbEquipes)
        do
                ( for(Indice2,1,NbEquipes),fromto([],In2,Out2,L2),param(T,Indice1)
                do
                        Var is T[Indice2,Indice1],
                        append(In2,[Var],Out2)
                ),
                append(In,L2,Out)
        ).

/* Tests
[eclipse 9]: getData(TailleEq,NbEq,CapaBat,NbBat,NbConf),defineVars(T,NbEq,NbConf,NbBat),getVarList(T,L).

TailleEq = [](5, 5, 2, 1)
NbEq = 4
CapaBat = [](7, 6, 5)
NbBat = 3
NbConf = 3
```

```
T = []([](_484{1 .. 3}, _553{1 .. 3}, _622{1 .. 3}), [](_693{1 .. 3}, _762{1 .. 3}, _831{1 .. 3}), [](_902{1 .. 3}, _971{1 .. 3}, _1040{1 .. 3}), [](_1111{1 .. 3}, _1180{1 .. 3}, _1249{1 .. 3}))
L = [_484{1 .. 3}, _693{1 .. 3}, _902{1 .. 3}, _1111{1 .. 3}, _553{1 .. 3}, _762{1 .. 3}, _971{1 .. 3}, _1180{1 .. 3}, _622{1 .. 3}, _831{1 .. 3}, _1040{1 .. 3}, _1249{1 .. 3}]
Yes (0.00s cpu)
*/
```

Question 4.4 Nous définissons ici un prédicat *solve(?T)* qui résoud le problème des régates où seules les contraintes de domaines sont posées.

Listing 4 – "solve1"

```
solve1(T) :-
        getData(_TailleEquipes,NbEquipes,_CapaBateaux,NbBateaux,NbConf),
        defineVars(T,NbEquipes,NbConf,NbBateaux),
        getVarList(T,L),
        labeling(L).

/* Tests
[eclipse 10]: solve1(T).

T = []([](1, 1, 1), [](1, 1, 1), [](1, 1, 1), [](1, 1, 1))
Yes (0.00s cpu, solution 1, maybe more) ? ;

T = []([](1, 1, 1), [](1, 1, 1), [](1, 1, 1), [](1, 1, 2))
Yes (0.00s cpu, solution 2, maybe more) ? ;

T = []([](1, 1, 1), [](1, 1, 1), [](1, 1, 1), [](1, 1, 3))
Yes (0.00s cpu, solution 3, maybe more) ?
*/
```

Question 4.5 Nous définissons ici un prédicat *pasMemeBateaux(+T,+NbEquipes,+NbConf)* qui impose qu'une même équipe ne retourne pas deux fois sur le même bateau. On modifie ensuite le prédicat *solve* pour qu'il prenne en compte cette nouvelle contrainte.

Listing 5 – "pasMemeBateaux"

```
pasMemeBateaux(T,NbEquipes,NbConf):-
        dim(T,[NbEquipes,NbConf]),
        ( for(Indice1,1,NbEquipes),param(T,NbConf)
        do
                ( for(Indice2,1,NbConf), fromto([],In,Out,L), param(T,Indice1)
                )
                do
                        Bat is T[Indice1,Indice2],
                        append(In,[Bat],Out)
                ),
                alldifferent(L)
        ).

```

```
solve2(T) :-
        getData(_TailleEquipes ,NbEquipes ,_CapaBateaux ,NbBateaux ,NbConf) ,
        defineVars(T,NbEquipes ,NbConf,NbBateaux) ,
        pasMemeBateaux(T,NbEquipes ,NbConf) ,
        getVarList(T,L) ,
        labeling(L).

/* Tests
[eclipse 11]: solve2(T).

T = []([](1, 2, 3), [](1, 2, 3), [](1, 2, 3), [](1, 2, 3))
Yes (0.00s cpu, solution 1, maybe more) ? ;

T = []([](1, 2, 3), [](1, 2, 3), [](1, 2, 3), [](1, 3, 2))
Yes (0.00s cpu, solution 2, maybe more) ? ;

T = []([](1, 2, 3), [](1, 2, 3), [](1, 3, 2), [](1, 2, 3))
Yes (0.00s cpu, solution 3, maybe more) ?
*/
```

Question 4.6 Nous définissons ici un prédicat *pasMemePartenaires(+T,+NbEquipes,+NbConf)* qui impose qu'une même équipe ne se retrouve pas deux fois avec la même équipe. On modifie une nouvelle fois le prédicat *solve* pour qu'il prenne en compte cette nouvelle contrainte.

Listing 6 – "pasMemePartenaires"

```
pasMemePartenaires(T,NbEquipes ,NbConf):-
        dim(T,[NbEquipes ,NbConf]) ,
        ( for(Equipe1 ,1 ,NbEquipes) ,param(T,NbConf,NbEquipes)
        do
                Indice is Equipe1+1,
                ( for(Equipe2 ,Indice ,NbEquipes) , param(T,Equipe1 ,NbConf)
                do

                        ( for(Conf ,1 ,NbConf) ,param(T,Equipe1 ,Equipe2) ,fromto
                            (0 ,In ,Out ,Tot)
                        do
                                Bateau1 is T[Equipe1 ,Conf] ,
                                Bateau2 is T[Equipe2 ,Conf] ,
                                #=(Bateau1 ,Bateau2 ,Ans) ,
                                Out #= In + Ans
                        ) ,
                        Tot #=< 1
                )
        ).

solve3(T) :-
        getData(_TailleEquipes ,NbEquipes ,_CapaBateaux ,NbBateaux ,NbConf) ,
        defineVars(T,NbEquipes ,NbConf,NbBateaux) ,
        pasMemeBateaux(T,NbEquipes ,NbConf) ,
        pasMemePartenaires(T,NbEquipes ,NbConf) ,
        getVarList(T,L) ,
        labeling(L).
```

```

/* Tests
[eclipse 12]: solve3(T).

T = []([](1, 2, 3), [](1, 3, 2), [](2, 1, 3), [](2, 3, 1))
Yes (0.00s cpu, solution 1, maybe more) ? ;

T = []([](1, 2, 3), [](1, 3, 2), [](2, 3, 1), [](2, 1, 3))
Yes (0.00s cpu, solution 2, maybe more) ? ;

T = []([](1, 3, 2), [](1, 2, 3), [](2, 1, 3), [](2, 3, 1))
Yes (0.00s cpu, solution 3, maybe more) ?
*/
```

Question 4.7 Nous définissons ici un prédicat *capaBateaux(+T,+TailleEquipes,+NbEquipes,+CapaBateaux,+NbB* qui vérifie que les capacités des bateaux sont respectées lors de chaque confrontation. On modifie une nouvelle fois le prédicat *solve* pour qu'il prenne en compte cette nouvelle contrainte.

Listing 7 – "capaBateaux"

```
capaBateaux(T,TailleEquipes,NbEquipes,CapaBateaux,NbBateaux,NbConf):-
        dim(T,[NbEquipes,NbConf]),
        ( for(Bateau,1,NbBateaux), param(T,NbEquipes,NbConf,CapaBateaux,
            TailleEquipes)
        do
                ( for(Conf,1,NbConf), param(T,NbEquipes,Bateau,CapaBateaux,
                    TailleEquipes)
                do
                        ( for(Equipe,1,NbEquipes),param(T,Bateau,Conf,
                            TailleEquipes),fromto(0,In,Out,Total)
                        do
                                BateauI is T[Equipe,Conf],
                                #=(Bateau,BateauI,Cond),
                                Inc #= TailleEquipes[Equipe] * Cond,
                                Out #= In + Inc
                        ),
                        Capacite is CapaBateaux[Bateau],
                        Total #=< Capacite
                )
        ).

solve4(T) :-
        getData(TailleEquipes,NbEquipes,CapaBateaux,NbBateaux,NbConf),
        defineVars(T,NbEquipes,NbConf,NbBateaux),
        pasMemeBateaux(T,NbEquipes,NbConf),
        pasMemePartenaires(T,NbEquipes,NbConf),
        capaBateaux(T,TailleEquipes,NbEquipes,CapaBateaux,NbBateaux,NbConf),
        getVarList(T,L),
        labeling(L).

/* Tests
[eclipse 13]: solve4(T).

```

```
T = []([](1, 2, 3), [](2, 3, 1), [](3, 1, 2), [](3, 2, 1))
Yes (0.01s cpu, solution 1, maybe more) ? ;

T = []([](1, 3, 2), [](2, 1, 3), [](3, 2, 1), [](3, 1, 2))
Yes (0.01s cpu, solution 2, maybe more) ? ;

T = []([](1, 2, 3), [](3, 1, 2), [](2, 3, 1), [](1, 3, 2))
Yes (0.01s cpu, solution 3, maybe more) ?
*/
```

Question 4.8 On passe ici à un problème de taille réelle. On dispose dorénavant de 13 voiliers et de 29 équipes qui doivent effectuer la régate comportant 7 confrontations.
Le temps d'exécution étant relativement long, il nous a été proposé d'améliorer le labeling. Pour cela, nous avons mélanger la liste des varaibles obtenu après *getVarList*, en alternant simplement les grosses et les petites équipes. Le gain sur le temps d'exécution est relativement important, puisqu'il est quasiment de 10 !

Listing 8 – "Problème de taille réelle et labeling"

```
getData2(TailleEquipes,NbEquipes,CapaBateaux,NbBateaux,NbConf):-
        TailleEquipes =
            [](7,6,5,5,5,4,4,4,4,4,4,4,4,3,3,2,2,2,2,2,2,2,2,2,2,2,2,2,2),
        NbEquipes = 29,
        CapaBateaux = [](10,10,9,8,8,8,8,8,8,7,6,4,4),
        NbBateaux = 13,
        NbConf = 7.

solve5(T):-
        getData2(TailleEq,NbEq,CapaBat,NbBat,NbConf),
        defineVars(T,NbEq,NbConf,NbBat),
        pasMemeBateaux(T,NbEq,NbConf),
        pasMemePartenaires(T,NbEq,NbConf),
        capaBateaux(T,TailleEq,NbEq,CapaBat,NbBat,NbConf),
        getVarList(T,L),
        labeling(L).

/* Tests
[eclipse 14]: solve5(T).

T = [](](1, 2, 3, 4, 5, 6, 7), [](2, 1, 4, 3, 6, 5, 8), [](3, 4, 1, 2, 7, 8,
     5), [](4, 3, 1, 5, 2, 7, 6), [](5, 6, 2, 1, 3, 4, 9), [](2, 3, 5, 1, 4,
    9, 10), [](3, 1, 2, 6, 4, 10, 11), [](6, 5, 7, 2, 1, 3, 4), [](6, 7, 5,
    8, 2, 1, 3), [](7, 5, 6, 8, 3, 2, 1), [](7, 8, 9, 6, 1, 11, 2), [](8, 7,
    6, 9, 10, 12, 2), [](8, 9, 7, 10, 11, 1, 12), [](1, 4, 8, 3, 9, 7, 10),
    [](4, 2, 8, 10, 7, 3, 9), [](5, 8, 3, 11, 6, 9, 1), [](9, 6, 4, 5, 8, 11,
     13), [](9, 8, 10, 12, 13, 2, 11), [](9, 10, 8, 11, 12, 13, 2), [](9, 11,
     12, 13, 1, 10, 3), [](10, 9, 11, 7, 12, 3, 13), [](10, 11, 9, 12, 8, 1,
    4), [](10, 12, 13, 11, 9, 2, 3), [](11, 9, 10, 13, 8, 4, 5), [](11, 10,
    12, 9, 13, 5, 1), [](11, 12, 9, 7, 10, 13, 8), [](12, 10, 13, 7, 11, 9,
    4), [](12, 13, 11, 9, 8, 2, 6), [](13, 11, 10, 7, 9, 8, 1))
Yes (55.23s cpu, solution 1, maybe more) ?
*/
```

```

% Amelioration du labeling pour gagner du temps

getLast([A],A,[]).
getLast([A|R],B,[A|L]):- getLast(R,B,L).

debutfin([A|R],A,L,B) :- getLast(R,B,L).
debutfin([A,B,C],A,[A,C,B],C).

melangListe([],[]).
melangListe([A,B,C],[A,C,B]):-!.
melangListe(L,[A,B|L2]) :- debutfin(L,A,L1,B),melangListe(L1,L2).

getVarList2(T,L):-
        dim(T,[NbEquipes,NbConf]),
        ( for(Indice1,1,NbConf),fromto([],In,Out,L),param(T,NbEquipes)
        do
                ( for(Indice2,1,NbEquipes),fromto([],In2,Out2,L2),param(T, Indice1)
                do
                        Var is T[Indice2,Indice1],
                        append(In2,[Var],Out2)
                ),
                melangListe(L2,L3),
                append(In,L3,Out)
        ).

solve6(T):-
        getData2(TailleEq,NbEq,CapaBat,NbBat,NbConf),
        defineVars(T,NbEq,NbConf,NbBat),
        pasMemeBateaux(T,NbEq,NbConf),
        pasMemePartenaires(T,NbEq,NbConf),
        capaBateaux(T,TailleEq,NbEq,CapaBat,NbBat,NbConf),
        getVarList2(T,L),
        labeling(L).

/* Tests
[eclipse 15]: solve6(T).

T = []([](1, 2, 3, 4, 5, 6, 7), [](2, 1, 4, 3, 6, 5, 8), [](3, 4, 1, 2, 8, 7, 9), [](4, 3, 5, 1, 2, 9, 10), [](5, 6, 2, 7, 1, 3, 4), [](6, 5, 7, 2, 1, 4, 3), [](6, 7, 8, 5, 2, 1, 11), [](7, 5, 6, 8, 9, 1, 2), [](8, 9, 7, 10, 4, 11, 5), [](8, 10, 9, 6, 11, 3, 2), [](9, 8, 12, 13, 7, 2, 6), [](10, 9, 8, 11, 13, 12, 1), [](12, 13, 11, 9, 10, 8, 1), [](11, 10, 13, 3, 8, 9, 4), [](11, 12, 10, 7, 3, 13, 9), [](13, 11, 10, 9, 6, 1, 5), [](13, 8, 11, 12, 4, 10, 3), [](10, 11, 9, 8, 12, 2, 3), [](9, 11, 6, 12, 10, 5, 13), [](9, 7, 10, 11, 12, 8, 2), [](7, 8, 9, 10, 3, 4, 1), [](7, 6, 5, 9, 11, 2, 8), [](5, 7, 1, 8, 3, 10, 13), [](4, 1, 6, 5, 3, 2, 12), [](3, 2, 4, 1, 7, 11, 12), [](3, 1, 2, 6, 9, 10, 11), [](2, 4, 3, 1, 9, 8, 6), [](2, 3, 1, 6, 7, 4, 5), [](1, 3, 2, 5, 4, 7, 6))
Yes (6.98s cpu, solution 1, maybe more) ? ;

```

```
T = []([](1, 2, 3, 4, 5, 6, 7), [](2, 1, 4, 3, 6, 5, 8), [](3, 4, 1, 2, 8, 7,
     9), [](4, 3, 5, 1, 2, 9, 10), [](5, 6, 2, 7, 1, 3, 4), [](6, 5, 7, 2, 1,
     4, 11), [](6, 7, 8, 5, 2, 1, 3), [](7, 5, 6, 8, 9, 1, 2), [](8, 9, 7,
    10, 4, 11, 5), [](8, 10, 9, 6, 11, 3, 2), [](9, 8, 12, 13, 7, 2, 6),
    [](10, 9, 8, 11, 13, 12, 1), [](12, 13, 11, 9, 10, 8, 1), [](11, 10, 13,
    3, 8, 9, 4), [](11, 12, 10, 7, 3, 13, 9), [](13, 11, 10, 9, 6, 1, 5),
    [](13, 8, 11, 12, 4, 10, 3), [](10, 11, 9, 8, 12, 2, 3), [](9, 11, 6, 12,
     10, 5, 13), [](9, 7, 10, 11, 12, 8, 2), [](7, 8, 9, 10, 3, 4, 1), [](7,
    6, 5, 9, 11, 2, 8), [](5, 7, 1, 8, 3, 10, 13), [](4, 1, 6, 5, 3, 2, 12),
    [](3, 2, 4, 1, 7, 11, 12), [](3, 1, 2, 6, 9, 10, 11), [](2, 4, 3, 1, 9,
    8, 6), [](2, 3, 1, 6, 7, 4, 5), [](1, 3, 2, 5, 4, 7, 6))
Yes (7.31s cpu, solution 2, maybe more) ?
*/
```

1 Code Complet, avec l'ensemble des tests

Listing 9 – "TP4"

```
:- lib(ic).

% Q4.1
getData(TailleEquipes,NbEquipes,CapaBateaux,NbBateaux,NbConf):-
        TailleEquipes = [](5,5,2,1),
        NbEquipes = 4,
        CapaBateaux = [](7,6,5),
        NbBateaux = 3,
        NbConf = 3.

/* Tests
[eclipse 7]: getData(TailleEq,NbEq,CapaBat,NbBat,NbConf).

TailleEq = [](5, 5, 2, 1)
NbEq = 4
CapaBat = [](7, 6, 5)
NbBat = 3
NbConf = 3
Yes (0.00s cpu)
*/

% Q4.2
defineVars(T,NbEquipes,NbConf,NbBateaux):-
        dim(T,[NbEquipes,NbConf]),
        ( for(Ind1,1,NbEquipes),param(T,NbBateaux,NbConf)
        do
                ( for(Ind2,1,NbConf),param(T,Ind1,NbBateaux)
                do
                        T[Ind1,Ind2] #:: 1..NbBateaux
                )
        ).

/* Tests
[eclipse 8]: getData(TailleEq,NbEq,CapaBat,NbBat,NbConf),defineVars(T,NbEq,NbConf,NbBat).

TailleEq = [](5, 5, 2, 1)
NbEq = 4
CapaBat = [](7, 6, 5)
NbBat = 3
NbConf = 3
T = []([](_419{1 .. 3}, _488{1 .. 3}, _557{1 .. 3}), [](_628{1 .. 3}, _697{1 .. 3}, _766{1 .. 3}), [](_837{1 .. 3}, _906{1 .. 3}, _975{1 .. 3}), [](_1046{1 .. 3}, _1115{1 .. 3}, _1184{1 .. 3}))
Yes (0.00s cpu)
*/

% Q4.3
getVarList(T,L):-
```

```
        dim(T,[NbEquipes,NbConf]),
        ( for(Indice1,1,NbConf),fromto([],In,Out,L),param(T,NbEquipes)
        do
                ( for(Indice2,1,NbEquipes),fromto([],In2,Out2,L2),param(T,Indice1)
                do
                        Var is T[Indice2,Indice1],
                        append(In2,[Var],Out2)
                ),
                append(In,L2,Out)
        ).

/* Tests
[eclipse 9]: getData(TailleEq,NbEq,CapaBat,NbBat,NbConf),defineVars(T,NbEq,NbConf,NbBat),getVarList(T,L).

TailleEq = [](5, 5, 2, 1)
NbEq = 4
CapaBat = [](7, 6, 5)
NbBat = 3
NbConf = 3
T = []([](_484{1 .. 3}, _553{1 .. 3}, _622{1 .. 3}), [](_693{1 .. 3}, _762{1 .. 3}, _831{1 .. 3}), [](_902{1 .. 3}, _971{1 .. 3}, _1040{1 .. 3}), [](_1111{1 .. 3}, _1180{1 .. 3}, _1249{1 .. 3}))
L = [_484{1 .. 3}, _693{1 .. 3}, _902{1 .. 3}, _1111{1 .. 3}, _553{1 .. 3}, _762{1 .. 3}, _971{1 .. 3}, _1180{1 .. 3}, _622{1 .. 3}, _831{1 .. 3}, _1040{1 .. 3}, _1249{1 .. 3}]
Yes (0.00s cpu)
*/

% Q4.4

solve1(T) :-
        getData(_TailleEquipes,NbEquipes,_CapaBateaux,NbBateaux,NbConf),
        defineVars(T,NbEquipes,NbConf,NbBateaux),
        getVarList(T,L),
        labeling(L).

/* Tests
[eclipse 10]: solve1(T).

T = []([](1, 1, 1), [](1, 1, 1), [](1, 1, 1), [](1, 1, 1))
Yes (0.00s cpu, solution 1, maybe more) ? ;

T = []([](1, 1, 1), [](1, 1, 1), [](1, 1, 1), [](1, 1, 2))
Yes (0.00s cpu, solution 2, maybe more) ? ;

T = []([](1, 1, 1), [](1, 1, 1), [](1, 1, 1), [](1, 1, 3))
Yes (0.00s cpu, solution 3, maybe more) ?
*/

% Q4.5
pasMemeBateaux(T,NbEquipes,NbConf):-
```

```
        dim(T,[NbEquipes,NbConf]),
        ( for(Indice1,1,NbEquipes),param(T,NbConf)
        do
                ( for(Indice2,1,NbConf), fromto([],In,Out,L), param(T,Indice1)
                )
                do
                        Bat is T[Indice1,Indice2],
                        append(In,[Bat],Out)
                ),
                alldifferent(L)
        ).

solve2(T) :-
        getData(_TailleEquipes,NbEquipes,_CapaBateaux,NbBateaux,NbConf),
        defineVars(T,NbEquipes,NbConf,NbBateaux),
        pasMemeBateaux(T,NbEquipes,NbConf),
        getVarList(T,L),
        labeling(L).

/* Tests
[eclipse 11]: solve2(T).

T = []([](1, 2, 3), [](1, 2, 3), [](1, 2, 3), [](1, 2, 3))
Yes (0.00s cpu, solution 1, maybe more) ? ;

T = []([](1, 2, 3), [](1, 2, 3), [](1, 2, 3), [](1, 3, 2))
Yes (0.00s cpu, solution 2, maybe more) ? ;

T = []([](1, 2, 3), [](1, 2, 3), [](1, 3, 2), [](1, 2, 3))
Yes (0.00s cpu, solution 3, maybe more) ?
*/

% Q4.6
pasMemePartenaires(T,NbEquipes,NbConf):-
        dim(T,[NbEquipes,NbConf]),
        ( for(Equipe1,1,NbEquipes),param(T,NbConf,NbEquipes)
        do
                Indice is Equipe1+1,
                ( for(Equipe2,Indice,NbEquipes), param(T,Equipe1,NbConf)
                do

                        ( for(Conf,1,NbConf),param(T,Equipe1,Equipe2),fromto(0,In,Out,Tot)
                        do
                                Bateau1 is T[Equipe1,Conf],
                                Bateau2 is T[Equipe2,Conf],
                                #=(Bateau1,Bateau2,Ans),
                                Out #= In + Ans
                        ),
                        Tot #=< 1
                )
        ).

```

```
solve3(T) :-
        getData(_TailleEquipes,NbEquipes,_CapaBateaux,NbBateaux,NbConf),
        defineVars(T,NbEquipes,NbConf,NbBateaux),
        pasMemeBateaux(T,NbEquipes,NbConf),
        pasMemePartenaires(T,NbEquipes,NbConf),
        getVarList(T,L),
        labeling(L).

/* Tests
[eclipse 12]: solve3(T).

T = []([](1, 2, 3), [](1, 3, 2), [](2, 1, 3), [](2, 3, 1))
Yes (0.00s cpu, solution 1, maybe more) ? ;

T = []([](1, 2, 3), [](1, 3, 2), [](2, 3, 1), [](2, 1, 3))
Yes (0.00s cpu, solution 2, maybe more) ? ;

T = []([](1, 3, 2), [](1, 2, 3), [](2, 1, 3), [](2, 3, 1))
Yes (0.00s cpu, solution 3, maybe more) ?
*/

% Q4.7
capaBateaux(T,TailleEquipes,NbEquipes,CapaBateaux,NbBateaux,NbConf):-
        dim(T,[NbEquipes,NbConf]),
        ( for(Bateau,1,NbBateaux), param(T,NbEquipes,NbConf,CapaBateaux,TailleEquipes)
        do
                ( for(Conf,1,NbConf), param(T,NbEquipes,Bateau,CapaBateaux,TailleEquipes)
                do
                        ( for(Equipe,1,NbEquipes),param(T,Bateau,Conf,TailleEquipes),fromto(0,In,Out,Total)
                        do
                                BateauI is T[Equipe,Conf],
                                #=(Bateau,BateauI,Cond),
                                Inc #= TailleEquipes[Equipe] * Cond,
                                Out #= In + Inc
                        ),
                        Capacite is CapaBateaux[Bateau],
                        Total #=< Capacite
                )
        ).

solve4(T) :-
        getData(TailleEquipes,NbEquipes,CapaBateaux,NbBateaux,NbConf),
        defineVars(T,NbEquipes,NbConf,NbBateaux),
        pasMemeBateaux(T,NbEquipes,NbConf),
        pasMemePartenaires(T,NbEquipes,NbConf),
        capaBateaux(T,TailleEquipes,NbEquipes,CapaBateaux,NbBateaux,NbConf),
        getVarList(T,L),
        labeling(L).

/* Tests
```

```
[eclipse 13]: solve4(T).

T = []([](1, 2, 3), [](2, 3, 1), [](3, 1, 2), [](3, 2, 1))
Yes (0.01s cpu, solution 1, maybe more) ? ;

T = []([](1, 3, 2), [](2, 1, 3), [](3, 2, 1), [](3, 1, 2))
Yes (0.01s cpu, solution 2, maybe more) ? ;

T = []([](1, 2, 3), [](3, 1, 2), [](2, 3, 1), [](1, 3, 2))
Yes (0.01s cpu, solution 3, maybe more) ?
*/

% Q4.8

getData2(TailleEquipes,NbEquipes,CapaBateaux,NbBateaux,NbConf):-
          TailleEquipes = [](7,6,5,5,5,4,4,4,4,4,4,4,4,3,3,2,2,2,2,2,2,2,2,2,2,2,2,2,2),
          NbEquipes = 29,
          CapaBateaux = [](10,10,9,8,8,8,8,8,8,7,6,4,4),
          NbBateaux = 13,
          NbConf = 7.

solve5(T):-
          getData2(TailleEq,NbEq,CapaBat,NbBat,NbConf),
          defineVars(T,NbEq,NbConf,NbBat),
          pasMemeBateaux(T,NbEq,NbConf),
          pasMemePartenaires(T,NbEq,NbConf),
          capaBateaux(T,TailleEq,NbEq,CapaBat,NbBat,NbConf),
          getVarList(T,L),
          labeling(L).

/* Tests
[eclipse 14]: solve5(T).

T = []([](1, 2, 3, 4, 5, 6, 7), [](2, 1, 4, 3, 6, 5, 8), [](3, 4, 1, 2, 7, 8, 5), [](4, 3, 1, 5, 2, 7, 6), [](5, 6, 2, 1, 3, 4, 9), [](2, 3, 5, 1, 4, 9, 10), [](3, 1, 2, 6, 4, 10, 11), [](6, 5, 7, 2, 1, 3, 4), [](6, 7, 5, 8, 2, 1, 3), [](7, 5, 6, 8, 3, 2, 1), [](7, 8, 9, 6, 1, 11, 2), [](8, 7, 6, 9, 10, 12, 2), [](8, 9, 7, 10, 11, 1, 12), [](1, 4, 8, 3, 9, 7, 10), [](4, 2, 8, 10, 7, 3, 9), [](5, 8, 3, 11, 6, 9, 1), [](9, 6, 4, 5, 8, 11, 13), [](9, 8, 10, 12, 13, 2, 11), [](9, 10, 8, 11, 12, 13, 2), [](9, 11, 12, 13, 1, 10, 3), [](10, 9, 11, 7, 12, 3, 13), [](10, 11, 9, 12, 8, 1, 4), [](10, 12, 13, 11, 9, 2, 3), [](11, 9, 10, 13, 8, 4, 5), [](11, 10, 12, 9, 13, 5, 1), [](11, 12, 9, 7, 10, 13, 8), [](12, 10, 13, 7, 11, 9, 4), [](12, 13, 11, 9, 8, 2, 6), [](13, 11, 10, 7, 9, 8, 1))
Yes (55.23s cpu, solution 1, maybe more) ?
*/

% Amelioration du labeling pour gagner du temps

getLast([A],A,[]).
getLast([A|R],B,[A|L]):- getLast(R,B,L).

```

```
debutfin([A|R],A,L,B) :- getLast(R,B,L).
debutfin([A,B,C],A,[A,C,B],C).

melangListe([],[]).
melangListe([A,B,C],[A,C,B]):-!.
melangListe(L,[A,B|L2]) :- debutfin(L,A,L1,B),melangListe(L1,L2).

getVarList2(T,L):-
        dim(T,[NbEquipes,NbConf]),
        ( for(Indice1,1,NbConf),fromto([],In,Out,L),param(T,NbEquipes)
        do
                ( for(Indice2,1,NbEquipes),fromto([],In2,Out2,L2),param(T,
                    Indice1)
                do
                        Var is T[Indice2,Indice1],
                        append(In2,[Var],Out2)
                ),
                melangListe(L2,L3),
                append(In,L3,Out)
        ).

solve6(T):-
        getData2(TailleEq,NbEq,CapaBat,NbBat,NbConf),
        defineVars(T,NbEq,NbConf,NbBat),
        pasMemeBateaux(T,NbEq,NbConf),
        pasMemePartenaires(T,NbEq,NbConf),
        capaBateaux(T,TailleEq,NbEq,CapaBat,NbBat,NbConf),
        getVarList2(T,L),
        labeling(L).

/* Tests
[eclipse 15]: solve6(T).

T = []([](1, 2, 3, 4, 5, 6, 7), [](2, 1, 4, 3, 6, 5, 8), [](3, 4, 1, 2, 8, 7,
     9), [](4, 3, 5, 1, 2, 9, 10), [](5, 6, 2, 7, 1, 3, 4), [](6, 5, 7, 2, 1,
     4, 3), [](6, 7, 8, 5, 2, 1, 11), [](7, 5, 6, 8, 9, 1, 2), [](8, 9, 7,
    10, 4, 11, 5), [](8, 10, 9, 6, 11, 3, 2), [](9, 8, 12, 13, 7, 2, 6),
    [](10, 9, 8, 11, 13, 12, 1), [](12, 13, 11, 9, 10, 8, 1), [](11, 10, 13,
    3, 8, 9, 4), [](11, 12, 10, 7, 3, 13, 9), [](13, 11, 10, 9, 6, 1, 5),
    [](13, 8, 11, 12, 4, 10, 3), [](10, 11, 9, 8, 12, 2, 3), [](9, 11, 6, 12,
     10, 5, 13), [](9, 7, 10, 11, 12, 8, 2), [](7, 8, 9, 10, 3, 4, 1), [](7,
    6, 5, 9, 11, 2, 8), [](5, 7, 1, 8, 3, 10, 13), [](4, 1, 6, 5, 3, 2, 12),
    [](3, 2, 4, 1, 7, 11, 12), [](3, 1, 2, 6, 9, 10, 11), [](2, 4, 3, 1, 9,
    8, 6), [](2, 3, 1, 6, 7, 4, 5), [](1, 3, 2, 5, 4, 7, 6))
Yes (6.98s cpu, solution 1, maybe more) ? ;

T = []([](1, 2, 3, 4, 5, 6, 7), [](2, 1, 4, 3, 6, 5, 8), [](3, 4, 1, 2, 8, 7,
     9), [](4, 3, 5, 1, 2, 9, 10), [](5, 6, 2, 7, 1, 3, 4), [](6, 5, 7, 2, 1,
     4, 11), [](6, 7, 8, 5, 2, 1, 3), [](7, 5, 6, 8, 9, 1, 2), [](8, 9, 7,
    10, 4, 11, 5), [](8, 10, 9, 6, 11, 3, 2), [](9, 8, 12, 13, 7, 2, 6),
    [](10, 9, 8, 11, 13, 12, 1), [](12, 13, 11, 9, 10, 8, 1), [](11, 10, 13,
    3, 8, 9, 4), [](11, 12, 10, 7, 3, 13, 9), [](13, 11, 10, 9, 6, 1, 5),
    [](13, 8, 11, 12, 4, 10, 3), [](10, 11, 9, 8, 12, 2, 3), [](9, 11, 6, 12,
        10, 5, 13), [](9, 7, 10, 11, 12, 8, 2), [](7, 8, 9, 10, 3, 4, 1), [](7,
       6, 5, 9, 11, 2, 8), [](5, 7, 1, 8, 3, 10, 13), [](4, 1, 6, 5, 3, 2, 12),
       [](3, 2, 4, 1, 7, 11, 12), [](3, 1, 2, 6, 9, 10, 11), [](2, 4, 3, 1, 9,
       8, 6), [](2, 3, 1, 6, 7, 4, 5), [](1, 3, 2, 5, 4, 7, 6))
Yes (7.31s cpu, solution 2, maybe more) ?
*/
```