

Très bon travail

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*, *?NbConf*) qui unifie les variables passées en paramètres avec les données du problème.

Listing 1 – "getData"

```
1  getData( TailleEquipes , NbEquipes , CapaBateaux , NbBateaux , NbConf ):-  
2      TailleEquipes = [(5,5,2,1) ,  
3      NbEquipes = 4,  
4      CapaBateaux = [(7,6,5) ,  
5      NbBateaux = 3,  
6      NbConf = 3.  
7  
8  /* Tests  
9  [ eclipse 7]: getData( TailleEq , NbEq , CapaBat , NbBat , NbConf ).  
10  
11  TailleEq = [(5, 5, 2, 1)  
12  NbEq = 4  
13  CapaBat = [(7, 6, 5)  
14  NbBat = 3  
15  NbConf = 3  
16  Yes (0.00 s cpu)  
17  */
```

OK

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"

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

```

2      dim(T,[NbEquipes,NbConf]),
3      ( for(Ind1,1,NbEquipes),param(T,NbBateaux,NbConf)
4      do
5          ( for(Ind2,1,NbConf),param(T,Ind1,NbBateaux)
6          do
7              T[Ind1,Ind2] #:: 1..NbBateaux
8          )
9      ).
10
11  /* Tests
12  [eclipse 8]: getData(TailleEq,NbEq,CapaBat,NbBat,NbConf),defineVars(T,NbEq,
13              NbConf,NbBat).
14
15  TailleEq = [(5, 5, 2, 1)
16  NbEq = 4
17  CapaBat = [(7, 6, 5)
18  NbBat = 3
19  NbConf = 3
20  T = [([(1, 419, 3), (1, 488, 3), (1, 557, 3)], [(1, 628, 3), (1, 697, 3),
21              (1, 766, 3)], [(1, 837, 3), (1, 906, 3), (1, 975, 3)], [(1,
22              1046, 3), (1, 1115, 3), (1, 1184, 3)])]
23  Yes (0.00s cpu)
24  */

```

Bien

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"

```

1  getVarList(T,L):-
2      dim(T,[NbEquipes,NbConf]),
3      ( for(Indice1,1,NbConf),fromto([],In,Out,L),param(T,NbEquipes)
4      do
5          ( for(Indice2,1,NbEquipes),fromto([],In2,Out2,L2),param(T,
6              Indice1)
7          do
8              Var is T[Indice2,Indice1],
9              append(In2,[Var],Out2)
10             ),
11             append(In,L2,Out)
12         ).
13
14  /* Tests
15  [eclipse 9]: getData(TailleEq,NbEq,CapaBat,NbBat,NbConf),defineVars(T,NbEq,
16              NbConf,NbBat),getVarList(T,L).
17
18  TailleEq = [(5, 5, 2, 1)
19  NbEq = 4
20  CapaBat = [(7, 6, 5)
21  NbBat = 3
22  NbConf = 3

```

Bien

```

21  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}))
22  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}]
23  Yes (0.00s cpu)
24  */

```

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"

```

1  solve1(T) :-
2      getData(_TailleEquipes, NbEquipes, _CapaBateaux, NbBateaux, NbConf),
3      defineVars(T, NbEquipes, NbConf, NbBateaux),
4      getVarList(T, L),
5      labeling(L).
6
7  /* Tests
8  [eclipse 10]: solve1(T).
9
10 T = [ ]([ ](1, 1, 1), [ ](1, 1, 1), [ ](1, 1, 1), [ ](1, 1, 1))
11 Yes (0.00s cpu, solution 1, maybe more) ? ;
12
13 T = [ ]([ ](1, 1, 1), [ ](1, 1, 1), [ ](1, 1, 1), [ ](1, 1, 2))
14 Yes (0.00s cpu, solution 2, maybe more) ? ;
15
16 T = [ ]([ ](1, 1, 1), [ ](1, 1, 1), [ ](1, 1, 1), [ ](1, 1, 3))
17 Yes (0.00s cpu, solution 3, maybe more) ?
18 */

```

Bien

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"

```

1  pasMemeBateaux(T, NbEquipes, NbConf) :-
2      dim(T, [NbEquipes, NbConf]),
3      ( for (Indice1, 1, NbEquipes), param(T, NbConf)
4      do
5          ( for (Indice2, 1, NbConf), fromto([], In, Out, L), param(T, Indice1
6              )
7              do
8                  Bat is T[Indice1, Indice2],
9                  append(In, [Bat], Out)
10             ),
11             alldifferent(L)
12         ).

```

Bien

```

13 solve2(T) :-
14     getData(_TailleEquipes ,NbEquipes ,_CapaBateaux ,NbBateaux ,NbConf) ,
15     defineVars (T,NbEquipes ,NbConf,NbBateaux) ,
16     pasMemeBateaux(T,NbEquipes ,NbConf) ,
17     getVarList(T,L) ,
18     labeling (L) .
19
20 /* Tests
21 [eclipse 11]: solve2(T).
22
23 T = [[[(1, 2, 3), [(1, 2, 3), [(1, 2, 3), [(1, 2, 3))
24 Yes (0.00s cpu, solution 1, maybe more) ? ;
25
26 T = [[[(1, 2, 3), [(1, 2, 3), [(1, 2, 3), [(1, 3, 2))
27 Yes (0.00s cpu, solution 2, maybe more) ? ;
28
29 T = [[[(1, 2, 3), [(1, 2, 3), [(1, 3, 2), [(1, 2, 3))
30 Yes (0.00s cpu, solution 3, maybe more) ?
31 */

```

OK

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"

```

1 pasMemePartenaires (T,NbEquipes ,NbConf):-
2     dim(T,[NbEquipes ,NbConf]) ,
3     ( for (Equipe1,1,NbEquipes) ,param (T,NbConf ,NbEquipes)
4     do
5         Indice is Equipe1+1,
6         ( for (Equipe2,Indice ,NbEquipes) , param (T,Equipe1 ,NbConf)
7         do
8
9             ( for (Conf,1,NbConf) ,param (T,Equipe1 ,Equipe2) ,fromto
10             (0,In ,Out ,Tot)
11             do
12                 Bateau1 is T[Equipe1 ,Conf] ,
13                 Bateau2 is T[Equipe2 ,Conf] ,
14                 #=(Bateau1 ,Bateau2 ,Ans) ,
15                 Out #= In + Ans
16                 Tot #=< 1
17             ) ,
18         ) .
19
20 solve3(T) :-
21     getData(_TailleEquipes ,NbEquipes ,_CapaBateaux ,NbBateaux ,NbConf) ,
22     defineVars (T,NbEquipes ,NbConf,NbBateaux) ,
23     pasMemeBateaux (T,NbEquipes ,NbConf) ,
24     pasMemePartenaires (T,NbEquipes ,NbConf) ,
25     getVarList (T,L) ,
26     labeling (L) .

```

Bien

```

27
28 /* Tests
29 [eclipse 12]: solve3(T).
30
31 T = [[[(1, 2, 3), [(1, 3, 2), [(2, 1, 3), [(2, 3, 1))
32 Yes (0.00s cpu, solution 1, maybe more) ? ;
33
34 T = [[[(1, 2, 3), [(1, 3, 2), [(2, 3, 1), [(2, 1, 3))
35 Yes (0.00s cpu, solution 2, maybe more) ? ;
36
37 T = [[[(1, 3, 2), [(1, 2, 3), [(2, 1, 3), [(2, 3, 1))
38 Yes (0.00s cpu, solution 3, maybe more) ?
39 */

```

Question 4.7 Nous définissons ici un prédicat *capaBateaux*(+T,+TailleEquipes,+NbEquipes,+CapaBateaux,+NbBateaux) 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"

```

1 capaBateaux(T, TailleEquipes, NbEquipes, CapaBateaux, NbBateaux, NbConf):-
2     dim(T,[NbEquipes,NbConf]),
3     ( for(Bateau,1,NbBateaux), param(T,NbEquipes,NbConf,CapaBateaux,
4         TailleEquipes)
5         do
6             ( for(Conf,1,NbConf), param(T,NbEquipes,Bateau,CapaBateaux,
7                 TailleEquipes)
8                 do
9                     BateauI is T[Equipe,Conf],
10                    #=(Bateau,BateauI,Cond),
11                    Inc #= TailleEquipes[Equipe] * Cond,
12                    Out #= In + Inc
13                ),
14                Capacite is CapaBateaux[Bateau],
15                Total #=< Capacite
16            )
17        ).
18
19 solve4(T):-
20     getData(TailleEquipes,NbEquipes,CapaBateaux,NbBateaux,NbConf),
21     defineVars(T,NbEquipes,NbConf,NbBateaux),
22     pasMemeBateaux(T,NbEquipes,NbConf),
23     pasMemePartenaires(T,NbEquipes,NbConf),
24     capaBateaux(T,TailleEquipes,NbEquipes,CapaBateaux,NbBateaux,NbConf),
25     getVarList(T,L),
26     labeling(L).
27
28 /* Tests
29 [eclipse 13]: solve4(T).
30

```

Bien

```

31 T = [ ]([ ](1, 2, 3), [ ](2, 3, 1), [ ](3, 1, 2), [ ](3, 2, 1))
32 Yes (0.01s cpu, solution 1, maybe more) ? ;
33
34 T = [ ]([ ](1, 3, 2), [ ](2, 1, 3), [ ](3, 2, 1), [ ](3, 1, 2))
35 Yes (0.01s cpu, solution 2, maybe more) ? ;
36
37 T = [ ]([ ](1, 2, 3), [ ](3, 1, 2), [ ](2, 3, 1), [ ](1, 3, 2))
38 Yes (0.01s cpu, solution 3, maybe more) ?
39 */

```

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 mélangé la liste des variables 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"

```

1  getData2 ( TailleEquipes , NbEquipes , CapaBateaux , NbBateaux , NbConf ) : -
2      TailleEquipes =
3          [ ](7,6,5,5,5,4,4,4,4,4,4,4,3,3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2) ,
4      NbEquipes = 29 ,
5      CapaBateaux = [ ](10,10,9,8,8,8,8,8,8,8,7,6,4,4) ,
6      NbBateaux = 13 ,
7      NbConf = 7 .
8  solve5 ( T ) : -
9      getData2 ( TailleEq , NbEq , CapaBat , NbBat , NbConf ) ,
10     defineVars ( T , NbEq , NbConf , NbBat ) ,
11     pasMemeBateaux ( T , NbEq , NbConf ) ,
12     pasMemePartenaires ( T , NbEq , NbConf ) ,
13     capaBateaux ( T , TailleEq , NbEq , CapaBat , NbBat , NbConf ) ,
14     getVarList ( T , L ) ,
15     labeling ( L ) .
16
17 /* Tests
18 [ eclipse 14 ] : solve5 ( T ) .
19
20 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))
21 Yes (55.23s cpu, solution 1, maybe more) ?
22 */

```

```

23
24 % Amelioration du labeling pour gagner du temps
25
26 getLast([A],A,[ ]).
27 getLast([A|R],B,[A|L]):- getLast(R,B,L). = append(L,[B],R)
28
29 debutfin([A|R],A,L,B):- getLast(R,B,L).
30 debutfin([A,B,C],A,[A,C,B],C).
31
32 melangListe([ ],[ ]).
33 melangListe([A,B,C],[A,C,B]):-!.
34 melangListe(L,[A,B|L2]):- debutfin(L,A,L1,B),melangListe(L1,L2).
35
36 getVarList2(T,L):-
37     dim(T,[NbEquipes,NbConf]),
38     ( for(Indice1,1,NbConf),fromto([ ],In,Out,L),param(T,NbEquipes)
39     do
40         ( for(Indice2,1,NbEquipes),fromto([ ],In2,Out2,L2),param(T,
41             Indice1)
42         do
43             Var is T[Indice2,Indice1],
44             append(In2,[Var],Out2)
45         ),
46         melangListe(L2,L3),
47         append(In,L3,Out)
48     ).
49
50 solve6(T):-
51     getData2(TailleEq,NbEq,CapaBat,NbBat,NbConf),
52     defineVars(T,NbEq,NbConf,NbBat),
53     pasMemeBateaux(T,NbEq,NbConf),
54     pasMemePartenaires(T,NbEq,NbConf),
55     capaBateaux(T,TailleEq,NbEq,CapaBat,NbBat,NbConf),
56     getVarList2(T,L),
57     labeling(L).
58
59 /* Tests
60 [eclipse 15]: solve6(T).
61
62 T = [[](1, 2, 3, 4, 5, 6, 7), [(2, 1, 4, 3, 6, 5, 8), [(3, 4, 1, 2, 8, 7,
63     9), [(4, 3, 5, 1, 2, 9, 10), [(5, 6, 2, 7, 1, 3, 4), [(6, 5, 7, 2, 1,
64     4, 3), [(6, 7, 8, 5, 2, 1, 11), [(7, 5, 6, 8, 9, 1, 2), [(8, 9, 7,
65     10, 4, 11, 5), [(8, 10, 9, 6, 11, 3, 2), [(9, 8, 12, 13, 7, 2, 6),
66     [(10, 9, 8, 11, 13, 12, 1), [(12, 13, 11, 9, 10, 8, 1), [(11, 10, 13,
67     3, 8, 9, 4), [(11, 12, 10, 7, 3, 13, 9), [(13, 11, 10, 9, 6, 1, 5),
68     [(13, 8, 11, 12, 4, 10, 3), [(10, 11, 9, 8, 12, 2, 3), [(9, 11, 6, 12,
69     10, 5, 13), [(9, 7, 10, 11, 12, 8, 2), [(7, 8, 9, 10, 3, 4, 1), [(7,
70     6, 5, 9, 11, 2, 8), [(5, 7, 1, 8, 3, 10, 13), [(4, 1, 6, 5, 3, 2, 12),
71     [(3, 2, 4, 1, 7, 11, 12), [(3, 1, 2, 6, 9, 10, 11), [(2, 4, 3, 1, 9,
72     8, 6), [(2, 3, 1, 6, 7, 4, 5), [(1, 3, 2, 5, 4, 7, 6))
73
74 Yes (6.98s cpu, solution 1, maybe more) ? ;

```

```

64  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))
65  Yes (7.31s cpu, solution 2, maybe more) ?
66  */

```


1 Code Complet, avec l'ensemble des tests

Listing 9 – "TP4"

```
1 :-lib(ic).
2
3 % Q4.1
4 getData(TailleEquipes,NbEquipes,CapaBateaux,NbBateaux,NbConf):-
5     TailleEquipes = [(5,5,2,1),
6     NbEquipes = 4,
7     CapaBateaux = [(7,6,5),
8     NbBateaux = 3,
9     NbConf = 3.
10
11 /* Tests
12 [eclipse 7]: getData(TailleEq,NbEq,CapaBat,NbBat,NbConf).
13
14 TailleEq = [(5, 5, 2, 1)
15 NbEq = 4
16 CapaBat = [(7, 6, 5)
17 NbBat = 3
18 NbConf = 3
19 Yes (0.00s cpu)
20 */
21
22 % Q4.2
23 defineVars(T,NbEquipes,NbConf,NbBateaux):-
24     dim(T,[NbEquipes,NbConf]),
25     ( for(Ind1,1,NbEquipes),param(T,NbBateaux,NbConf)
26     do
27         ( for(Ind2,1,NbConf),param(T,Ind1,NbBateaux)
28         do
29             T[Ind1,Ind2] #:: 1..NbBateaux
30         )
31     ).
32
33 /* Tests
34 [eclipse 8]: getData(TailleEq,NbEq,CapaBat,NbBat,NbConf),defineVars(T,NbEq,
35     NbConf,NbBat).
36
37 TailleEq = [(5, 5, 2, 1)
38 NbEq = 4
39 CapaBat = [(7, 6, 5)
40 NbBat = 3
41 NbConf = 3
42 T = [([](_419{1..3},_488{1..3},_557{1..3}),[( _628{1..3},_697{1
43     ..3},_766{1..3}),[( _837{1..3},_906{1..3},_975{1..3}),[(
44     _1046{1..3},_1115{1..3},_1184{1..3})]
45 Yes (0.00s cpu)
46 */
47
48 % Q4.3
49 getVarList(T,L):-
```

```

47     dim(T,[NbEquipes,NbConf]),
48     ( for (Indice1,1,NbConf),fromto([],In,Out,L),param(T,NbEquipes)
49     do
50         ( for (Indice2,1,NbEquipes),fromto([],In2,Out2,L2),param(T,
51             Indice1)
52         do
53             Var is T[Indice2,Indice1],
54             append(In2,[Var],Out2)
55         ),
56         append(In,L2,Out)
57     ).
58 /* Tests
59 [eclipse 9]: getData(TailleEq,NbEq,CapaBat,NbBat,NbConf),defineVars(T,NbEq,
60 NbConf,NbBat),getVarList(T,L).
61 TailleEq = [(5, 5, 2, 1)
62 NbEq = 4
63 CapaBat = [(7, 6, 5)
64 NbBat = 3
65 NbConf = 3
66 T = [( [(484{1 .. 3}, _553{1 .. 3}, _622{1 .. 3}), [(693{1 .. 3}, _762{1
67 .. 3}, _831{1 .. 3}), [(902{1 .. 3}, _971{1 .. 3}, _1040{1 .. 3}), [(
68 _1111{1 .. 3}, _1180{1 .. 3}, _1249{1 .. 3})]
69 L = [_484{1 .. 3}, _693{1 .. 3}, _902{1 .. 3}, _1111{1 .. 3}, _553{1 .. 3},
70 _762{1 .. 3}, _971{1 .. 3}, _1180{1 .. 3}, _622{1 .. 3}, _831{1 .. 3},
71 _1040{1 .. 3}, _1249{1 .. 3}]
72 Yes (0.00s cpu)
73 */
74 % Q4.4
75 solve1(T) :-
76     getData(_TailleEquipes,NbEquipes,_CapaBateaux,NbBateaux,NbConf),
77     defineVars(T,NbEquipes,NbConf,NbBateaux),
78     getVarList(T,L),
79     labeling(L).
80 /* Tests
81 [eclipse 10]: solve1(T).
82 T = [( [(1, 1, 1), [(1, 1, 1), [(1, 1, 1), [(1, 1, 1)
83 Yes (0.00s cpu, solution 1, maybe more) ? ;
84
85 T = [( [(1, 1, 1), [(1, 1, 1), [(1, 1, 1), [(1, 1, 2)
86 Yes (0.00s cpu, solution 2, maybe more) ? ;
87
88 T = [( [(1, 1, 1), [(1, 1, 1), [(1, 1, 1), [(1, 1, 3)
89 Yes (0.00s cpu, solution 3, maybe more) ?
90 */
91
92 % Q4.5
93 pasMemeBateaux(T,NbEquipes,NbConf):-

```

```

94     dim(T,[NbEquipes,NbConf]),
95     ( for(Indice1,1,NbEquipes),param(T,NbConf)
96     do
97         ( for(Indice2,1,NbConf), fromto([],In,Out,L), param(T,Indice1
98             )
99         do
100             Bat is T[Indice1,Indice2],
101             append(In,[Bat],Out)
102         ),
103         alldifferent(L)
104     ).
105 solve2(T) :-
106     getData(_TailleEquipes,NbEquipes,_CapaBateaux,NbBateaux,NbConf),
107     defineVars(T,NbEquipes,NbConf,NbBateaux),
108     pasMemeBateaux(T,NbEquipes,NbConf),
109     getVarList(T,L),
110     labeling(L).
111
112 /* Tests
113 [eclipse 11]: solve2(T).
114
115 T = [[[(1, 2, 3), [(1, 2, 3), [(1, 2, 3), [(1, 2, 3)]]
116 Yes (0.00s cpu, solution 1, maybe more) ? ;
117
118 T = [[[(1, 2, 3), [(1, 2, 3), [(1, 2, 3), [(1, 3, 2)]]
119 Yes (0.00s cpu, solution 2, maybe more) ? ;
120
121 T = [[[(1, 2, 3), [(1, 2, 3), [(1, 3, 2), [(1, 2, 3)]]
122 Yes (0.00s cpu, solution 3, maybe more) ?
123 */
124
125 % Q4.6
126 pasMemePartenaires(T,NbEquipes,NbConf):-
127     dim(T,[NbEquipes,NbConf]),
128     ( for(Equipe1,1,NbEquipes),param(T,NbConf,NbEquipes)
129     do
130         Indice is Equipe1+1,
131         ( for(Equipe2,Indice,NbEquipes), param(T,Equipe1,NbConf)
132         do
133
134             ( for(Conf,1,NbConf),param(T,Equipe1,Equipe2),fromto
135                 (0,In,Out,Tot)
136             do
137                 Bateau1 is T[Equipe1,Conf],
138                 Bateau2 is T[Equipe2,Conf],
139                 #=(Bateau1,Bateau2,Ans),
140                 Out #= In + Ans
141             ),
142             Tot #=< 1
143         )
144     ).

```

```

145 solve3(T) :-
146     getData(_TailleEquipes, NbEquipes, _CapaBateaux, NbBateaux, NbConf),
147     defineVars(T, NbEquipes, NbConf, NbBateaux),
148     pasMemeBateaux(T, NbEquipes, NbConf),
149     pasMemePartenaires(T, NbEquipes, NbConf),
150     getVarList(T, L),
151     labeling(L).
152
153 /* Tests
154 [eclipse 12]: solve3(T).
155
156 T = []([](1, 2, 3), [](1, 3, 2), [](2, 1, 3), [](2, 3, 1))
157 Yes (0.00s cpu, solution 1, maybe more) ? ;
158
159 T = []([](1, 2, 3), [](1, 3, 2), [](2, 3, 1), [](2, 1, 3))
160 Yes (0.00s cpu, solution 2, maybe more) ? ;
161
162 T = []([](1, 3, 2), [](1, 2, 3), [](2, 1, 3), [](2, 3, 1))
163 Yes (0.00s cpu, solution 3, maybe more) ?
164 */
165
166 % Q4.7
167 capaBateaux(T, TailleEquipes, NbEquipes, CapaBateaux, NbBateaux, NbConf):-
168     dim(T, [NbEquipes, NbConf]),
169     ( for(Bateau, 1, NbBateaux), param(T, NbEquipes, NbConf, CapaBateaux,
170         TailleEquipes)
171     do
172         ( for(Conf, 1, NbConf), param(T, NbEquipes, Bateau, CapaBateaux,
173             TailleEquipes)
174         do
175             ( for(Equipe, 1, NbEquipes), param(T, Bateau, Conf,
176                 TailleEquipes), fromto(0, In, Out, Total)
177             do
178                 BateauI is T[Equipe, Conf],
179                 #=(Bateau, BateauI, Cond),
180                 Inc #= TailleEquipes[Equipe] * Cond,
181                 Out #= In + Inc
182             ),
183             Capacite is CapaBateaux[Bateau],
184             Total #=< Capacite
185         )
186     ).
187
188 solve4(T) :-
189     getData(TailleEquipes, NbEquipes, CapaBateaux, NbBateaux, NbConf),
190     defineVars(T, NbEquipes, NbConf, NbBateaux),
191     pasMemeBateaux(T, NbEquipes, NbConf),
192     pasMemePartenaires(T, NbEquipes, NbConf),
193     capaBateaux(T, TailleEquipes, NbEquipes, CapaBateaux, NbBateaux, NbConf),
194     getVarList(T, L),
195     labeling(L).
196
197 /* Tests

```

```

195 [eclipse 13]: solve4(T).
196
197 T = []([](1, 2, 3), [](2, 3, 1), [](3, 1, 2), [](3, 2, 1))
198 Yes (0.01s cpu, solution 1, maybe more) ? ;
199
200 T = []([](1, 3, 2), [](2, 1, 3), [](3, 2, 1), [](3, 1, 2))
201 Yes (0.01s cpu, solution 2, maybe more) ? ;
202
203 T = []([](1, 2, 3), [](3, 1, 2), [](2, 3, 1), [](1, 3, 2))
204 Yes (0.01s cpu, solution 3, maybe more) ?
205 */
206
207 % Q4.8
208
209 getData2(TailleEquipes ,NbEquipes ,CapaBateaux ,NbBateaux ,NbConf):-
210     TailleEquipes =
211         [(7,6,5,5,5,4,4,4,4,4,4,4,3,3,2,2,2,2,2,2,2,2,2,2,2,2,2,2)],
212     NbEquipes = 29,
213     CapaBateaux = [(10,10,9,8,8,8,8,8,8,7,6,4,4)],
214     NbBateaux = 13,
215     NbConf = 7.
216
217 solve5(T):-
218     getData2(TailleEq,NbEq,CapaBat,NbBat,NbConf),
219     defineVars(T,NbEq,NbConf,NbBat),
220     pasMemeBateaux(T,NbEq,NbConf),
221     pasMemePartenaires(T,NbEq,NbConf),
222     capaBateaux(T,TailleEq,NbEq,CapaBat,NbBat,NbConf),
223     getVarList(T,L),
224     labeling(L).
225
226 /* Tests
227 [eclipse 14]: solve5(T).
228
229 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))
230 Yes (55.23s cpu, solution 1, maybe more) ?
231 */
232
233 % Amelioration du labeling pour gagner du temps
234
235 getLast([A],A,[ ]).
236 getLast([A|R],B,[A|L]):- getLast(R,B,L).
```

```

237 debutfin ([A|R],A,L,B) :- getLast(R,B,L).
238 debutfin ([A,B,C],A,[A,C,B],C).
239
240 melangListe ([],[]).
241 melangListe ([A,B,C],[A,C,B]) :- !.
242 melangListe (L,[A,B|L2]) :- debutfin (L,A,L1,B), melangListe (L1,L2).
243
244 getVarList2 (T,L):-
245     dim(T,[NbEquipes,NbConf]),
246     ( for (Indice1,1,NbConf), fromto ([],In,Out,L), param (T,NbEquipes)
247     do
248         ( for (Indice2,1,NbEquipes), fromto ([],In2,Out2,L2), param (T,
249             Indice1)
250         do
251             Var is T[Indice2,Indice1],
252             append (In2,[Var],Out2)
253         ),
254         melangListe (L2,L3),
255         append (In,L3,Out)
256     ).
257 solve6 (T):-
258     getData2 (TailleEq,NbEq,CapaBat,NbBat,NbConf),
259     defineVars (T,NbEq,NbConf,NbBat),
260     pasMemeBateaux (T,NbEq,NbConf),
261     pasMemePartenaires (T,NbEq,NbConf),
262     capaBateaux (T,TailleEq,NbEq,CapaBat,NbBat,NbConf),
263     getVarList2 (T,L),
264     labeling (L).
265
266 /* Tests
267 [eclipse 15]: solve6(T).
268
269 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))
270 Yes (6.98s cpu, solution 1, maybe more) ? ;
271
272 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))
273  Yes (7.31s cpu, solution 2, maybe more) ?
274  */

```