TP4 - Les régates

Paul Chaignon - Ulysse Goarant

21 février 2014

Listing 1 – regates.ecl

```
:- lib(ic).

/**
 * Question 4.1
 * getData(?TailleEquipes, ?NbEquipes, ?CapaBateaux, ?NbBateaux,
    ?NbConf)
 */
getData(TailleEquipes, NbEquipes, CapaBateaux, NbBateaux, 7):-
  %TailleEquipes = [](5, 5, 2, 1),
  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),
  dim(TailleEquipes, [NbEquipes]),
  %CapaBateaux = [](7, 6, 5),
  CapaBateaux = [](10, 10, 9, 8, 8, 8, 8, 8, 8, 7, 6, 4, 4),
  dim(CapaBateaux, [NbBateaux]).

/**
 * Question 4.2
 * defineVars(?T, +NbEquipes, +NbConf, +NbBateaux)
 */
defineVars(T, NbEquipes, NbConf, NbBateaux):-
  dim(T, [NbEquipes, NbConf]),
  T #:: 1..NbBateaux.

/**
 * Question 4.3
 * getVarList(+T, ?L)
 */
getVarList(T, L):-
  dim(T, [NbEquipes, NbConf]),
  (for(J, 0, NbConf-1), fromto([], In, Out, L), param(T, NbEquipes,
     NbConf) do
    (for(I, 1, NbEquipes), foreach(Elem, SubL), param(T, J, NbConf) do
      JInv is NbConf - J,
      Elem is T[I, JInv]
    ),
    append(SubL, In, Out)
  ).

/**
 * Question 4.4
 * solve(?T)
 */
```

```
41 solve(T):-
42   getData(TailleEquipes, NbEquipes, CapaBateaux, NbBateaux, NbConf),
43   defineVars(T, NbEquipes, NbConf, NbBateaux),
44
45   pasMemeBateaux(T, NbEquipes, NbConf),
46   pasMemePartenaires(T, NbEquipes, NbConf),
47   capaBateaux(T, TailleEquipes, NbEquipes, CapaBateaux, NbBateaux,
       NbConf),
48
49   getVarList(T, L),
50   labeling(L).
51
52 /**
53  * Question 4.5
54  * pasMemeBateaux(+T, +NbEquipes, +NbConf)
55  * Verifie qu'une equipe ne concours pas deux fois avec le meme bateau.
56  * Il faut donc qu'il n'y ait pas deux fois la meme valeur sur une
       ligne.
57  */
58 pasMemeBateaux(T, NbEquipes, NbConf):-
59   (for(K, 1, NbEquipes), param(T, NbConf) do
60     (for(I, 1, NbConf), param(T, NbConf, K) do
61       (for(J, I+1, NbConf), param(T, I, K) do
62         T[K, I] #\= T[K, J]
63       )
64     )
65   ).
66
67 /**
68  * Question 4.6
69  * Verifie qu'une equipe ne se retrouve pas partenaires deux fois avec
       la meme equipe.
70  * pasMemePartenaires(+T, +NbEquipes, +NbConf)
71  * T[i][k] == T[j][k] => T[i][x] != T[j][x] pour tout x \ x != k.
72  */
73 pasMemePartenaires(T, NbEquipes, NbConf):-
74   (for(I, 1, NbEquipes), param(T, NbEquipes, NbConf) do
75     (for(J, I+1, NbEquipes), param(T, NbConf, I) do
76       (for(K, 1, NbConf), param(T, NbConf, I, J) do
77         (for(X, K+1, NbConf), param(T, I, J, K) do
78           (T[I, K] #= T[J, K]) => (T[I, X] #\= T[J, X])
79         )
80       )
81     )
82   ).
83
84 /**
85  * Question 4.7
86  * capaBateaux(+T, +TailleEquipes, +NbEquipes, +CapaBateaux,
       +NbBateaux, +NbConf)
87  * Verifie que les capacites des bateaux sont respectees.
88  */
89 capaBateaux(T, TailleEquipes, NbEquipes, CapaBateaux, NbBateaux,
       NbConf):-
90   (for(Conf, 1, NbConf), param(T, CapaBateaux, TailleEquipes,
       NbBateaux, NbEquipes) do
91     (for(Bateau, 1, NbBateaux), param(T, CapaBateaux, TailleEquipes,
         Conf, NbEquipes) do
```

```
92        (for(Equipe, 1, NbEquipes), fromto(0, TailleTotale,
              NewTailleTotale, TailleFinale),
93        param(T, TailleEquipes, Bateau, Conf) do
94          NewTailleTotale #= TailleTotale + (T[Equipe, Conf] #= Bateau) *
                TailleEquipes[Equipe]
95        ),
96        CapaBateaux[Bateau] #>= TailleFinale
97      )
98    ).
99
100 /**
101  * Question 4.8
102  * getVarListAlt(+T, ?List)
103  * Alterne une petite et une grande equipe par rapport a getVarList.
104  * Utilise le fait que les equipes sont donnes dans un tableau ordonne.
105  * L'ajout a la liste est donc realise en partant des deux extremites
         et en allant vers le milieu.
106  */
107 getVarListAlt(T, List):-
108   dim(T, [NbEquipes, NbConf]),
109   (for(J, 0, NbConf-1), fromto([], In, Out, List), param(T, NbEquipes,
          NbConf) do
110     MoitieNbEquipes is div(NbEquipes, 2),
111     (for(I, 0, MoitieNbEquipes-1), fromto([], SubIn, SubOut, SubList),
            param(MoitieNbEquipes, T, J, NbConf, NbEquipes) do
112       % Les indices sont inverses car fromto inverse les listes:
113       JInv is NbConf - J,
114       IInv is MoitieNbEquipes - I,
115       Elem1 is T[IInv, JInv], % Une grande equipe.
116       Elem2 is T[NbEquipes-IInv+1, JInv], % Une petite equipe.
117       SubOut = [Elem1, Elem2|SubIn]
118     ),
119     append(SubList, In, Out)
120   ).
121
122
123 /**
124  * Tests
125  */
126 getVarList([]([](3, 8), [](4, 9), [](1, 5), [](7, 10)), L).
127   L = [3, 4, 1, 7, 8, 9, 5, 10]
128
129 getVarListAlt([]([](3, 8), [](4, 9), [](1, 5), [](7, 10)), L).
130   L = [3, 7, 4, 1, 8, 10, 9, 5]
131
132 solve(T).
133   T = [][](1, 2, 3), [](2, 3, 1), [](3, 1, 2), [](3, 2, 1))
134   Yes (0.00s cpu, solution 1, maybe more)
135   T = [][](1, 3, 2), [](2, 1, 3), [](3, 2, 1), [](3, 1, 2))
136   Yes (0.00s cpu, solution 2, maybe more)
137   T = [][](1, 2, 3), [](3, 1, 2), [](2, 3, 1), [](1, 3, 2))
138 solve(T).
139   T = [][](1, 2, 3, 4, 5, 6, 7), [](2, 1, 4, 3, 6, 5, 8),
140     [](3, 4, 1, 2, 7, 8, 5), [](4, 3, 1, 5, 2, 7, 6),
141     [](5, 6, 2, 1, 3, 4, 9), [](2, 3, 5, 1, 4, 9, 10),
142     [](3, 1, 2, 6, 4, 10, 11), [](6, 5, 7, 2, 1, 3, 4),
143     [](6, 7, 5, 8, 2, 1, 3), [](7, 5, 6, 8, 3, 2, 1),
144     [](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 (204.77s cpu, solution 1, maybe more)
```