TP1 - Découverte de la bibliothèque de contraintes à domaines finis

Paul Chaignon - Ulysse Goarant

February 2, 2014

1 De Prolog à Prolog+ic

Listing 1: prolog-ic.ecl

```
1  :- lib(ic).
2
3  voiture(rouge).
4  voiture(vert(clair)).
5  voiture(gris).
6  voiture(blanc).
7  bateau(vert).
8  bateau(blanc).
9  bateau(noir).
10
11 /**
12  * Question 1.1
13  * choixCouleur(?CouleurBateau, ?CouleurVoiture)
14  */
15 choixCouleur(Couleur, Couleur):-
16   voiture(Couleur),
17   bateau(Couleur).
18
19
20 minResistance(5000).
21 maxResistance(10000).
22 minCondensateur(9000).
23 maxCondensateur(20000).
24
25 /**
26  * Question 1.3
27  * isBetween(?Var, +Min, -Max)
28  */
29 isBetween(Var, Min, Max):-
30   not(free(Var)),
31   Var >= Min,
32   Var =< Max.
33 isBetween(Var, Min, Max):-
34   free(Var),
35   isBetweenIncremental(Var, Min, Max).
36
37 /**
38  * isBetweenIncremental(-Var, +Min, +Max)
```

```
 */
isBetweenIncremental(Min, Min, Max).
isBetweenIncremental(Var, Min, Max):-
  NextMin is Min + 1,
  NextMin =< Max,
  isBetweenIncremental(Var, NextMin, Max).

/**
 * Question 1.4
 * commande(-NbResistances, -NbCondensateurs)
 */
commande(NbResistances, NbCondensateurs):-
  minResistance(MinResistance),
  maxResistance(MaxResistance),
  minCondensateur(MinCondensateur),
  maxCondensateur(MaxCondensateur),
  isBetween(NbResistances, MinResistance, MaxResistance),
  isBetween(NbCondensateurs, MinCondensateur, MaxCondensateur),
  NbResistances > NbCondensateurs.

/**
 * Question 1.7
 * commandeIC(-NbResistances, -NbCondensateurs)
 */
commandeIC(NbResistances, NbCondensateurs):-
  minResistance(MinResistance),
  maxResistance(MaxResistance),
  minCondensateur(MinCondensateur),
  maxCondensateur(MaxCondensateur),
  NbResistances #:: MinResistance..MaxResistance,
  NbCondensateurs #:: MinCondensateur..MaxCondensateur,
  NbResistances #> NbCondensateurs.

/**
 * Question 1.8
 * commandeLabeling(-NbResistances, -NbCondensateurs)
 */
commandeLabeling(NbResistances, NbCondensateurs):-
  commandeIC(NbResistances, NbCondensateurs),
  labeling([NbResistances, NbCondensateurs]).


/**
 * Tests
 */
/*
choixCouleur(blanc, blanc). => Yes
choixCouleur(noir, vert(clair)). => No
choixCouleur(vert, vert(clair)). => No
choixCouleur(CouleurBateau, CouleurVoiture). => 1 solution

isBetween(4000000, 1000000, 8000000). => Yes
isBetween(10000000, 1000000, 8000000). => No
isBetween(X, 1, 5). => 5 solutions

commande(NbResistances, NbCondensateurs).
findall((NbResistances, NbCondensateurs), commande(NbResistances,
   NbCondensateurs), Results), length(Results, NbResults). => 500500
```

```
commandeIC(NbResistances, NbCondensateurs). => Delayed goal.
commandeLabeling(NbResistances, NbCondensateurs).
findall((NbResistances, NbCondensateurs),
    commandeLabeling(NbResistances, NbCondensateurs), Results),
    length(Results, NbResults). => 500500
*/
```

Question 1.2

Pourquoi Prolog peut être considéré comme un solveur de contraintes sur le domaine des arbres ?

Question 1.5

Question 1.6

Pourquoi peut-on dire que Prolog ne comprend pas les Maths ?

Question 1.7

Question 1.8

2 Zoologie

Listing 2: zoologie.ecl

```
:- lib(ic).

/**
 * chapieBug(-Chats, -Pies, -Pattes, -Tetes)
 * chapie tourne dans le vide a partir de la seconde solution pour la
    question 1.10.
 */
chapieBug(Chats, Pies, Pattes, Tetes):-
  Chats #:: 0..inf,
  Pies #:: 0..inf,
  Pattes #:: 0..inf,
  Tetes #:: 0..inf,
  Pattes #= Chats * 4 + Pies * 2,
  Tetes #= Chats + Pies.

/**
 * chapie(-Chats, -Pies, -Pattes, -Tetes)
 */
chapie(Chats, Pies, Pattes, Tetes):-
  Chats #:: 0..1000,
  Pies #:: 0..1000,
  Pattes #:: 0..1000,
  Tetes #:: 0..1000,
  Pattes #= Chats * 4 + Pies * 2,
  Tetes #= Chats + Pies.
```

```
25
26 /**
27  * Question 1.9
28  */
29 % chapie(2, Pies, Pattes, 5).
30 /*
31   Pies = 3
32   Pattes = 14
33   Yes (0.00s cpu)
34 */
35
36 /**
37  * Question 1.10
38  */
39 % chapie(Chats, Pies, Pattes, Tetes), Pattes #= 3 * Tetes,
    labeling([Chats, Pies, Pattes, Tetes]).
```

3 Le OU en contraintes

Listing 3: ou-constraint.ecl

```
1 :- lib(ic).
2
3 /**
4  * Question 1.11
5  * vabs(?Val, ?AbsVal)
6  */
7 vabs(Val, AbsVal):-
8   AbsVal #> 0,
9   (
10    Val #= AbsVal
11  ;
12    Val #= -AbsVal
13  ),
14  labeling([Val, AbsVal]).
15
16 /**
17  * vabsIC(?Val, ?AbsVal)
18  */
19 vabsIC(Val, AbsVal):-
20  AbsVal #> 0,
21  Val #= AbsVal or Val #= -AbsVal,
22  labeling([Val, AbsVal]).
23
24 /**
25  * Question 1.12
26  */
27 % X #:: -10..10, vabs(X, Y).
28 % X #:: -10..10, vabsIC(X, Y).
29
30 /**
31  * Question 1.13
32  * faitListeBug(?ListVar, ?Taille, +Min, +Max)
```

```
33  * faitListeBug(ListVar, 2, 1, 3) tourne dans le vide a partir de la
        seconde solution.
34  * Avec un cut c'est faitListeBug(ListVar, _, 1, 3) qui retourne une
        seule solution.
35  */
36 faitListeBug([], 0, _, _).
37 faitListeBug([First|Rest], Taille, Min, Max):-
38   First #:: Min..Max,
39   Taille1 #= Taille - 1,
40   faitListeBug(Rest, Taille1, Min, Max).
41
42 /**
43  * faitListe(?ListVar, ?Taille, +Min, +Max)
44  */
45 faitListe(ListVar, Taille, Min, Max):-
46   length(ListVar, Taille),
47   ListVar #:: Min..Max.
48
49 /**
50  * Question 1.14
51  * suite(?ListVar)
52  */
53 suite([Xi, Xi1, Xi2]):-
54   checkRelation(Xi, Xi1, Xi2).
55 suite([Xi, Xi1, Xi2|Rest]):-
56   checkRelation(Xi, Xi1, Xi2),
57   suite([Xi1, Xi2|Rest]).
58
59 /**
60  * checkRelation(?Xi, ?Xi1, ?Xi2)
61  */
62 checkRelation(Xi, Xi1, Xi2):-
63   vabs(Xi1, VabsXi1),
64   Xi2 #= VabsXi1 - Xi.
65
66 /**
67  * Question 1.15
68  * checkPeriode(+ListVar).
69  */
70 % faitListe(ListVar, 18, -9, 9), suite(ListVar), ListVar = [X1, X2, X3,
      X4, X5, X6, X7, X8, X9, X10, X11|Rest], X1 #\= X10 or X2 #\= X11,
      labeling(ListVar).
71
72
73 /**
74  * Tests
75  */
76 /*
77 vabs(5, 5). => Yes
78 vabs(5, -5). => No
79 vabs(-5, 5). => Yes
80 vabs(X, 5).
81 vabs(X, AbsX).
82 vabsIC(5, 5). => Yes
83 vabsIC(5, -5). => No
84 vabsIC(-5, 5). => Yes
85 vabsIC(X, 5).
86 vabsIC(X, AbsX).
```

```
87
88 faitListe(ListVar, 5, 1, 3). => 243 solutions
89 faitListe(ListVar, _, 1, 3). !!!!!!!!!!!!
90 faitListe([_, _, _, _, _], Taille, 1, 3). => Taille = 5
91
92 faitListe(ListVar, 18, -9, 9), suite(ListVar). => 99 solutions
93 */
```

Question 1.12