

# CR TP7 Programmation par contraintes

---

Histoire de menteurs

Julien LETOILE, Romain HUBERT

le 18/03/2021

# Table des matières

---

## Table des matières

### I. [Réponses rédigées](#)

Question 7.1

Question 7.2

Question 7.3

Question 7.4

### I. [Code source](#)

# I. Réponses rédigées

## Question 7.1

```
1  affirme(S, A):-  
2      (S &= femme) => A.
```

Test:

```
X &:: personnes, X &= femme, affirme(X, A).  
X = femme  
A = 1  
Yes (0.00s cpu)
```

```
X &:: personnes, X &= homme, affirme(X, A).  
X = homme  
A = A{[0, 1]}  
Yes (0.00s cpu)
```

## Question 7.2

```
1  affirme(S, A1, A2):-  
2      (S &= homme) => ((A1 and neg(A2)) or (neg(A1) and A2)).
```

Test:

```
X &:: personnes, X &= homme, affirme(X, A, B).  
  
X = homme  
A = A{[0, 1]}  
B = B{[0, 1]}
```

Delayed goals:

```
1711{[0, 1]} + B{[0, 1]} #= 1
#=(1711{[0, 1]} + A{[0, 1]}, 2, 1681{[0, 1]})
_1869{[0, 1]} + A{[0, 1]} #= 1
#=(1869{[0, 1]} + B{[0, 1]}, 2, _1854{[0, 1]})
- _1854{[0, 1]} - _1681{[0, 1]} #=< -1
```

Yes (0.00s cpu)

## Question 7.3

```
1
2  ?- local domain(sexe(femme, homme)).
3
4  solve(Parent1, Parent2, Enfant):-
5      pose_domaines(Parent1, Parent2, Enfant),
6      pose_contraintes(Parent1, Parent2, Enfant),
7      labeling_symbolic([Parent1, Parent2, Enfant]).
8
9  pose_domaines(Parent1, Parent2, Enfant):-
10     Parent1 &:: sexe,
11     Parent2 &:: sexe,
12     Enfant &:: sexe.
13
```

Test:

```
pose_domaines(Parent1, Parent2, Enfant).
```

```
Parent1 = Parent1{[femme, homme]}
```

```
Parent2 = Parent2{[femme, homme]}
```

```
Enfant = Enfant{[femme, homme]}
```

## Question 7.4

```
1
2  pose_contraintes(Parent1, Parent2, Enfant):-
```

```

3      /* Enfant affirme : Arrheu, arrheu ! */
4      AffE #:: 0..1,
5
6      /* Parent1 affirme : Enfant vous dit qu'elle est une femme.
       */
7      AffP1 #= (Enfant &= femme),
8
9      /* Parent2 affirme : Enfant est un homme puis . . . */
10     Aff1P2 #= (Enfant &= homme),
11
12     /* Parent2 affirme : Enfant ment. */
13     Aff2P2 #= (AffE #= 0),
14
15     /* Liaisons entre personnes et affirmations */
16     affirme(Enfant, AffE),
17     affirme(Parent1, AffP1),
18     affirme(Parent2, Aff1P2),
19     affirme(Parent2, Aff2P2),
20     affirme(Parent2, Aff1P2, Aff2P2),
21
22     /* Les 2 parents sont du sexe opposé */
23     Parent1 &\= Parent2.
24

```

Résultat final:

```
solve(P1, P2, Enfant).
```

P1 = homme

P2 = femme

Enfant = homme

Yes (0.00s cpu)

## I. Code source

```
1  :-lib(ic).
2  :-lib(ic_symbolic).
3  :-lib(branch_and_bound).
4
5
6  ?- local domain(sexe(femme, homme)).
7
8
9  solve(Parent1, Parent2, Enfant):-
10     pose_domaines(Parent1, Parent2, Enfant),
11     pose_contraintes(Parent1, Parent2, Enfant),
12     labeling_symbolic([Parent1, Parent2, Enfant]).
13
14
15  pose_domaines(Parent1, Parent2, Enfant):-
16     Parent1 &:: sexe,
17     Parent2 &:: sexe,
18     Enfant &:: sexe.
19
20
21  pose_contraintes(Parent1, Parent2, Enfant):-
22     /* Enfant affirme : Arrheu, arrheu ! */
23     AffE #:: 0..1,
24
25     /* Parent1 affirme : Enfant vous dit qu'elle est une femme.
26     */
27     AffP1 #= (Enfant &= femme),
28
29     /* Parent2 affirme : Enfant est un homme puis . . . */
30     Aff1P2 #= (Enfant &= homme),
31
32     /* Parent2 affirme : Enfant ment. */
33     Aff2P2 #= (AffE #= 0),
34
35     /* Liaisons entre personnes et affirmations */
36     affirme(Enfant, AffE),
```

```
36     affirme(Parent1, AffP1),
37     affirme(Parent2, Aff1P2),
38     affirme(Parent2, Aff2P2),
39     affirme(Parent2, Aff1P2, Aff2P2),
40
41     /* Les 2 parents sont du sexe opposé */
42     Parent1 &\= Parent2.
43
44
45     /* Les femmes disent toujours la vérité */
46     affirme(S, A):-
47         (S &= femme) => A.
48
49
50     /* Les hommes alternent systématiquement entre vérité et
51     mensonge */
51     affirme(S, A1, A2):-
52         (S &= homme) => ((A1 and neg(A2)) or (neg(A1) and A2)).
53
54
55     /* Labeling symbolique */
56     labeling_symbolic([]).
57     labeling_symbolic([X | Liste]):-
58         ic_symbolic:indomain(X),
59         labeling_symbolic(Liste).
```