

# Programmation Logique et Par Contraintes Avancée

## Cours 10 – Exemple Réification : le QCM auto-référentiel

Ralf Treinen

Université Paris Cité  
UFR Informatique  
Institut de Recherche en Informatique Fondamentale



treinen@irif.fr

13 mars 2024

### Le quiz

1. La première question avec réponse b est  
(a) 2; (b) 3; (c) 4; (d) 5; (e) 6.
2. Les deux seules questions consécutives avec la même réponse sont  
(a) 2 et 3; (b) 3 et 4; (c) 4 et 5; (d) 5 et 6; (e) 6 et 7.
3. La réponse à cette question est la même qu'à la question  
(a) 1; (b) 2; (c) 4; (d) 7; (e) 6.
4. Le nombre de questions avec réponse a est  
(a) 0; (b) 1; (c) 2; (d) 3; (e) 4.
5. La réponse à cette question est la même qu'à la question  
(a) 10; (b) 9; (c) 8; (d) 7; (e) 6.

### Exemple : le test auto-référentiel

- ▶ Quiz à choix multiple
- ▶ Il y a 10 questions (1 - 10)
- ▶ Pour chaque question 5 réponses possibles (a - e)
- ▶ On a besoin d'aucune connaissance pour répondre! Mais ...
- ▶ ... pour toute question et réponse proposée il y a une condition qui fait référence aux réponses aux autres questions (dont "auto-référentiel").
- ▶ Il faut trouver une solution (une réponse par question) telle que pour toute question, seulement les conditions pour la réponse choisie sont vraies, et les conditions pour les autres réponses sont fausses.

- 6 Le nombre de questions avec réponse a est égal au nombre de questions avec réponse  
(a) b; (b) c; (c) d; (d) e; (e) aucune.
- 7 La distance (dans l'alphabet) entre la réponse à cette question et la réponse à la question suivante est  
(a) 4; (b) 3; (c) 2; (d) 1; (e) 0.
- 8 Le nombre de questions dont la réponse est une voyelle est  
(a) 2; (b) 3; (c) 4; (d) 5; (e) 6.
- 9 Le nombre de questions dont la réponse est une consonne est  
(a) un premier; (b) une factorielle; (c) un carré; (d) un cube; (e) divisible par 5.
- 10 La réponse à cette question est  
(a) a; (b) b; (c) c; (d) d; (e) e.

## Modélisation

- ▶ Il convient d'utiliser une modélisation redondante.
- ▶ Variables  $Q.I$ , pour  $I \in \{1..10\}$ , avec domaine  $[1..5]$ , contient la réponse à la question  $I$ .
- ▶ Variables booléennes  $A.I$ , pour  $I \in \{1..10\}$ , exprime que la réponse à la  $I$ -ème question est a. Pareil pour  $B.I$ ,  $C.I$ ,  $D.I$ ,  $E.I$ .
- ▶ Les variables  $Q.I$  servent à
  - ▶ exprimer plus simplement certaines contraintes (par exemple : la réponse aux questions  $n$  et  $m$  est la même)
  - ▶ imposer que les réponses à la même question sont mutuellement exclusives.

## Exemples (aptitude.oz) II

```

{FD.decl S} {FD.sum V '=' S}
end
SumA = {Sum A}      % nombre de questions avec reponse a
SumB = {Sum B}      % nombre de questions avec reponse b
SumC = {Sum C}      % nombre de questions avec reponse c
SumD = {Sum D}      % nombre de questions avec reponse d
SumE = {Sum E}      % nombre de questions avec reponse e
SumAE = {Sum [SumA SumE]} % nmbr qu avec reponse a ou e
SumBCD = {Sum [SumB SumC SumD]} % reponse b ou c ou d

% {Assert I IsA IsB IsC IsD IsE} launches propagators for
% the fact that the answer to question number I is the X
% such that the corresponding IsX is true. The Boolean
% variables IsX are usually obtained by reification of a
% constraint from the problem description.
% This implies that precisely one of the five variables
% IsA..IsE is true.

```

## Exemples (aptitude.oz) I

```

% The self-referential aptitude test

declare
proc {SRAT S}
    % initial domains for the redundant vectors representing
    % the answers to the questions
    Q = {FD.tuple q 10 1#5}
    A = {FD.tuple v 10 0#1}
    B = {FD.tuple v 10 0#1}
    C = {FD.tuple v 10 0#1}
    D = {FD.tuple v 10 0#1}
    E = {FD.tuple v 10 0#1}

    % declare and initialize local variables to count, using
    % a finite-domain propagator, the number of questions
    % with a certain answer.
    proc {Sum V S} % S is the sum over vector V

```

## Exemples (aptitude.oz) III

```

proc {Assert I IsA IsB IsC IsD IsE}
    A.I=IsA B.I=IsB C.I=IsC D.I=IsD E.I=IsE
end

in
S=Q
% link the vector Q to the vectors A..E
for I in 1..10 do
    {Assert I Q.I=:1 Q.I=:2 Q.I=:3 Q.I=:4 Q.I=:5}
end

% for instance: A.1=1 iff Q.1=1, B.1=1 iff Q.1=2, etc.

% Question 1
{Assert 1
    B.2 % (a)
    0 % (b)
    {FD.conj B.4 (B.2+B.3=:0)} % (c)

```

## Exemples (aptitude.oz) IV

```

{FD.conj B.5 (B.2+B.3+B.4=:0)}           % (d)
{FD.conj B.6 (B.2+B.3+B.4+B.5=:0)}}      % (e)
% if the answer to 1 is a,c,d or e, then B.1=0
% if the answer to 1 is b then contradiction.

% Question 2
{Assert 2
  Q.2=:Q.3 % (a)
  Q.3=:Q.4 % (b)
  Q.4=:Q.5 % (c)
  Q.5=:Q.6 % (d)
  Q.6=:Q.7} % (e)
% only one of the above equations can be true.
% we have to express that these equalities cannot
% be true:
Q.1\=:Q.2  Q.7\=:Q.8  Q.8\=:Q.9  Q.9\=:Q.10
    
```

## Exemples (aptitude.oz) VI

```

Q.8=:Q.5 % (c)
Q.7=:Q.5 % (d)
Q.6=:Q.5} % (e)

% Question 6
{Assert 6
  SumA=:SumB % (a)
  SumA=:SumC % (b)
  SumA=:SumD % (c)
  SumA=:SumE % (d)
  _} % (e)
% the only information for (e) is that the others
% are false, which is redundant.

% Question 7
local D in
  D::0#5
    
```

## Exemples (aptitude.oz) V

```

% Question 3
{Assert 3
  Q.1=:Q.3 % (a)
  Q.2=:Q.3 % (b)
  Q.4=:Q.3 % (c)
  Q.7=:Q.3 % (d)
  Q.6=:Q.3} % (e)

% Question 4
{FD.element Q.4 [0 1 2 3 4] SumA}
% that means that SumA is the Q.4-th element
% of the list [0 1 2 3 4]

% Question 5
{Assert 5
  Q.10=:Q.5 % (a)
  Q.9=:Q.5 % (b)
    
```

## Exemples (aptitude.oz) VII

```

{FD.distance Q.7 Q.8 '=: D}
{FD.element Q.7 [4 3 2 1 0] D}
end

% Question 8
{FD.element Q.8 [2 3 4 5 6] SumAE}
% the vowels are a and e
% SumAE must be the Q.8-th element of the list

% Question 9
{Assert 9
  SumBCD::[2 3 5 7] % (a)
  SumBCD::[1 2 6] % (b)
  SumBCD::[0 1 4 9] % (c)
  SumBCD::[0 1 8] % (d)
  SumBCD::[0 5 10]} % (e)
% the consonants are b, c, d.
    
```

## Exemples (aptitude.oz) VIII

```
% the possible numbers are calculated "by hand"  
  
% Question 10  
% nothing to do  
  
{FD.distribute ff Q}  
end  
  
{Browse {SearchAll SRAT}}
```