DMCommunity Challenge Sep-2019

James Delonay jamesdelonay@gmail.com
MinZinc

```
int: max_value ; % largest integer value that can be contained in
trials matrix.
int: rows ;
int: cols ;
array[1..cols] of int: answer ; % set to all zeros if actual code
is unknown.
array[1..rows,1..cols] of int: trials ; % matrix of rows trials.
array[1..rows] of int: contained ; % cows
array[1..rows] of int: correct ; % bulls

array[1..cols] of var 0..max_value: values ;


include "alldifferent.mzn";

/* recursive combination function to generate all of the various
bull and cow possibilites */
function array[int] of set of int: comb(int: m, array[int] of
int: lst) =
  if m == 0 then [{}]
  else [{lst[i]} union suffix| i in index_set(lst),
  suffix in comb(m - 1, lst[i+1..])] endif ;


predicate bulls_and_cows(array[int] of var int: values,
                         array[int,int] of 0..max_value: trials,
                         array[int] of int: contained,
                         array[int] of int: correct) =
  alldifferent(values) /\
  forall(r in 1..rows)
    (exists(cows in comb(contained[r], index_set(values)),
            bulls in comb(correct[r],cows))

      (forall(bull in bulls)(values[bull] = trials[r,bull]) /\
         forall(cow in cows where not (cow in bulls))
           (values[cow] != trials[r,cow] /\
```

```
              exists(c1 in index_set(values) where not (c1=cow \/
(c1 in bulls)))
                 (values[cow]=trials[r,c1])) /\
                  forall(c2 in index_set(values) where not (c2 in
cows),
                         c3 in index_set(values))
                    (values[c2]!=trials[r,c3])) ;


constraint bulls_and_cows(values, trials, contained, correct) ;
solve satisfy ;

var 0..2: code_correct = bool2int(forall([values[i] = answer[i]|i
in 1..cols])) + bool2int(sum(answer) > 0) ;
array[0..2] of string: results = array1d(0..2,
["unknown","incorrect","correct"]) ;
output ["The code is \(values), which is " ++
results[fix(code_correct)] ++ "."] ;

/* (results)
Finished in 290msec
Running bulls_and_cows.mzn
The code is [0, 5, 2], which is unknown.

( this is the result of the larger set below.)
Finished in 7s 945msec
Running bulls_and_cows.mzn
The code is [28, 8, 3, 4, 6, 19, 0, 12, 11, 17], which is
correct.
*/

%/* <- remove leading percent sign to run larger test.
max_value = 9 ;
rows = 5 ;
cols = 3 ;
answer = [0,0,0] ;
trials = [|6,8,2,|6,4,5,|2,0,6,|7,3,8,|7,8,0|] ;
contained = [1,1,2,0,1] ;
correct = [1,0,0,0,0] ;
% */
```

```
/* No thought was given to the theoretical limits of bulls and
cows, required rows vs max_value vs columns.
   It is more about just playing with MiniZinc.
   A 'code' vector was generated of cols unique integers less
than or equal to max_value,
   then similar vectors were generated and the correct and
contained vectors obtained.
*/

/* <- add leading percent sign to run larger test.
max_value = 32 ;
rows = 30 ;
cols = 10 ;
answer = [28, 8, 3, 4, 6, 19, 0, 12, 11, 17] ;

trials = [|31,18,21,8,24,6,27,9,13,1,
|13,21,20,26,1,17,3,16,25,9,
|21,11,22,8,30,18,23,3,4,1,
|12,7,14,30,25,13,4,15,29,24,
|29,5,20,28,17,7,8,18,14,21,
|22,10,12,27,2,24,16,3,11,7,
|8,2,3,16,19,13,18,11,23,22,
|1,28,6,24,23,0,2,11,18,21,
|2,16,30,23,31,6,11,15,20,10,
|19,21,14,18,24,26,17,9,20,28,
|26,13,15,1,10,30,18,16,28,20,
|13,18,25,30,22,1,12,4,14,23,
|12,0,6,5,7,9,28,8,2,17,
|3,2,5,27,19,7,25,0,15,24,
|18,11,28,15,2,3,12,10,0,7,
|26,27,0,15,4,25,6,30,1,19,
|18,31,9,25,16,2,19,20,11,26,
|13,8,24,19,31,27,20,11,29,15,
|28,20,10,13,6,5,11,4,14,9,
|4,7,2,1,17,11,23,27,24,29,
|30,22,8,16,19,18,10,1,21,17,
|21,1,31,17,25,8,15,26,20,11,
|24,27,21,0,13,15,12,16,5,28,
|7,6,20,9,17,19,12,14,30,11,
|11,10,0,14,13,3,5,16,2,21,
|9,3,0,15,17,29,7,12,8,30,
|23,9,22,7,16,15,27,30,21,26,
```

```
|30,5,27,7,28,22,6,1,9,10,
|6,19,0,10,14,22,2,30,7,8,
|28,22,30,31,15,16,11,13,6,9|] ;

contained = [2, 2, 4, 2, 3, 3, 4, 4, 2, 3, 1, 2, 6, 3, 5,
             4, 2, 3, 4, 3, 3, 3, 3, 5, 3, 5, 0, 2, 4, 3] ;

correct = [0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0,
           0, 1, 1, 2, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1] ;
% */
```