

5. tutorial in Prolog

May 10, 2016

Task 1: Prepare your toolbox. Create a predicate `all_diff(List)` which succeeds if all items in the list are different. Assume all items are non-variables.

```
?- all_diff([3,1,2,4]).  
true.  
?- all_diff([1,2,3,1]).  
false.
```

Task 2: Create the predicate `inst_four(List)` which instantiates all items in the `List` to numbers between 1 and 4.

```
?- inst_four([A,B]).  
A = 1, B = 1 ;  
A = 1, B = 2 ;  
A = 1, B = 3 ;  
A = 1, B = 4 ;  
A = 2, B = 1 ;  
A = 2, B = 2 ;  
A = 2, B = 3 ;  
...
```

Task 3: Create a sudoku solver for small 4×4 matrices. The matrix can be represented as a list-of-lists: `[[A1,A2,A3,A4], [B1,B2,B3,B4], [C1,C2,C3,C4], [D1,D2,D3,D4]]`. For a start, feel free to start with a simple *generate-and-test* scheme:

1. Instantiate all variables A1 to D4 to values between 1 and 4. Use `inst_four` predicate.
2. Make sure all variables in each row are different, in each column are different and in each block are different. Use `all_diff` predicate.

In the end, your solver should work as follows:

```
?- sudoku(X).  
X = [[1,2,3,4], [3,4,1,2], [2,1,4,3], [4,3,2,1]] ;  
X = [[1,2,3,4], [3,4,1,2], [2,3,4,1], [4,1,2,3]] ;  
X = [[1,2,3,4], [3,4,1,2], [4,1,2,3], [2,3,4,1]] ;  
...
```

Does it take too long? Go to the next task!

Task 4: Make the sudoku solver faster by breaking the generate-and-test. Call the `all_diff` constraints as soon as possible by moving them up in the code. Just not too soon: Check that the variables are instantiated, because `all_diff(X,Y)` (probably) fails.

You can measure the time to compute all solutions using

```
?- time(findall(X,sudoku(X),_)).  
% 38,298,381 inferences, 2.097 CPU in 2.097 seconds
```

Feel free to feel challenged. You can surely do better than my 2.1 s!

Task 5: Convert your code to use the *constraint satisfaction* library:

1. Load the library by adding “:- use_module(library(clpfd)).” into your code.
2. Use `all_different` instead of `all_diff` and
“`[A1,A2,A3,A4] ins 1..4`” instead of `inst_four(...)`.
3. Call `label([A1,A2,...D4])` at the end of the `sudoku` predicate.

Want to learn more? Try a very good tutorial or SWI Prolog manual:

<http://www.pathwayslms.com/swipltuts/clpfd/clpfd.html>

<http://www.swi-prolog.org/man/clpfd.html>