

# AI Prolog Assignment 2

## IIT2016128

### Ques 2: Combinatorial Alphabet Problem

#### Problem Statement:

Design a game considering following points.

- Objective is to fill a 9×9 grid with alphabets.
- Each column, each row, and each of the nine 3×3 subgrids that compose the grid (also called "boxes", "blocks", or "regions") contains all of the alphabets from A to I.
- Entries in 9×9 grid not more than 18.

#### Solution Approach:

The given problem is solved by first solving a 9x9 grid of sudoku which consists of numbers from 1-9. The integers 1-9 are then mapped with alphabets from A-I with 1 being A, 2 being B and so on upto 9 being I. When the solved sudoku grid is generated then while printing this mapping is used to generate grid of alphabets rather than the integers.

The code mainly consist of 2 functions:

1. combinatorial
2. Blocks
3. printproblem

The program make use of the inbuilt prolog library clpfd.

Function description:

1. blocks

This function takes a list and convert it into list of list so that the 3x3 blocks are generated.

```
25 combinatorial(Matrix) :-
26     flatten(Matrix, Numbers), Numbers ins 1..9,
27     Rows = Matrix,
28     transpose(Rows, Columns),
29     blocks(Rows, Blocks),
30     maplist(all_distinct, Rows),
31     maplist(all_distinct, Columns),
32     maplist(all_distinct, Blocks),
33     maplist(label, Rows),
34     printproblem(Matrix).
```

## 2. combinatorial

This function is the main driving function for the problem. It first take a 2d array as argument, it first converts it into a single list using flatten function. After that columns are retrieved by transposing the input matrix. Then this function create blocks and make use of all\_distinct function to make sure the elements in a row, column and block is distinct. This solves the 9x9 combinatorial integer problem.

## 3. printproblem

This function takes the solved integer board of the game and prints it in alphabetical form by looking up for the corresponding alphabet for the integer from the defined set of rules.

```
3  convert_int_to_char(1):-
4      write('A').
5  convert_int_to_char(2):-
6      write('B').
7  convert_int_to_char(3):-
8      write('C').
9  convert_int_to_char(4):-
10     write('D').
11 convert_int_to_char(5):-
12     write('E').
13 convert_int_to_char(6):-
14     write('F').
15 convert_int_to_char(7):-
16     write('G').
17 convert_int_to_char(8):-
18     write('H').
19 convert_int_to_char(9):-
20     write('I').
21
22 printproblem([]).
23 printproblem([H|T]) :-
24     printList(H),
25     printproblem(T).
26
27 printList([]) :-
28     nl.
29 printList([H|T]) :-
30     convert_int_to_char(H),
31     write(' | '),
32     printList(T).
```

## Algorithm:

The game board is represented as a matrix simulated with a list of lists representing rows. The field values are represented by integers.

1. Generate a game board which is a list of lists with some of the values filled. Values will be integers.

2. Compute the blocks of the puzzle by splitting the game board into groups of 3 rows, then compute blocks for each of the groups and finally combine all the answers into a single response by appending the generated rows to a list.
3. Compute the rows, columns and blocks of the game so that all the values in rows, columns and blocks is distinct.
4. Verify that each field has been assigned exactly one value.
5. Write facts which map integer with corresponding alphabets.
6. While printing, replace the integer with the corresponding alphabet.

## Input Format:

A 2d array with some entries filled with integers, labelled array for same 2d array.

Query the program using `combinatorial(Puzzle)` where `Puzzle` is the created 2d array.

## Code Output:

```

ai:swipl — Konsole
File Edit View Bookmarks Settings Help

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- [ques2].
true.

?- Puzzle = [
|      [2,_,_,_,9,_,_,_,1],
|      [3,_,9,_,_,7,_,_,_],
|      [_,_,1,_,4,_,_,7,_],
|      [_,6,_,_,_,_,_,_,_],
|      [_,_,_,_,_,3,_,_,_],
|      [_,_,8,6,_,_,7,9,_],
|      [6,_,_,7,_,_,8,_,_],
|      [1,2,3,_,_,8,_,_,_],
|      [_,8,7,_,_,4,3,_,_]], Puzzle = [A,B,C,D,E,F,G,H,I], combinatorial(Puzzle).
B | G | F | H | I | E | D | C | A |
C | D | I | A | B | G | E | F | H |
H | E | A | C | D | F | B | G | I |
G | F | B | D | H | I | A | E | C |
I | A | E | B | G | C | F | H | D |
D | C | H | F | E | A | G | I | B |
F | I | D | G | C | B | H | A | E |
A | B | C | E | F | H | I | D | G |
E | H | G | I | A | D | C | B | F |

```