CSL765: Introduction to (Logic and Functional) Programming
I semester 2011-12
Assignment

A Rikudo solver

Puzzles in general involve trying out various possibilities and therefore most algorithms for solving them involve backtracking. Prolog is a suitable programming language to solve logic puzzles because you get back-tracking for free! For this assignment, you have to solve such a puzzle using Prolog.

Problem Statement

Develop a Rikudo puzzle solver in Prolog. The puzzle Rikudo was created in 2015. Check this website to enjoy the puzzle and get a clearer idea about the rules.

The rules of the game are as follows:

1. The hexagonal cells (empty or filled) tile a large hexagon.
2. The number of cells n can be 37, 61 or 91 depending upon the size.
3. The numbers in empty cells should be filled to create a hamiltonian path of consecutive numbers from 1 to $n-1$.
4. The centre cell cannot contain a number, hence, no path can go through it.
5. The numbers cannot be repeated.
6. In some cases, a *link* will be given. The link indicates that the path is crossing though that edge i.e., if cell A and B are linked they should contain the consecutive numbers making a path from A to B or B to A.
7. A correct solution must fill all the cells each having a unique number.

The coordinate system for naming the individual cells (see fig. 1 and fig. 2) may be briefly described as follows:

1. (0,0) is the origin of the coordinate system (a red dot)
2. All the cells (blue dots) in the odd rows and odd columns have both coordinates odd.
3. All cells (blue dots) in even rows and even columns have both coordinates even.
4. A typical cell (x, y) has at most 6 neighbours drawn from $(x-2, y)$, $(x+2, y)$, $(x-1, y+1)$, $(x+1, y+1)$, $(x-1, y-1)$, $(x+1, y-1)$.

A Rikudo puzzle is defined in terms of:

Size An integer n denoting size of the grid. It can take values 37, 61 or 91.

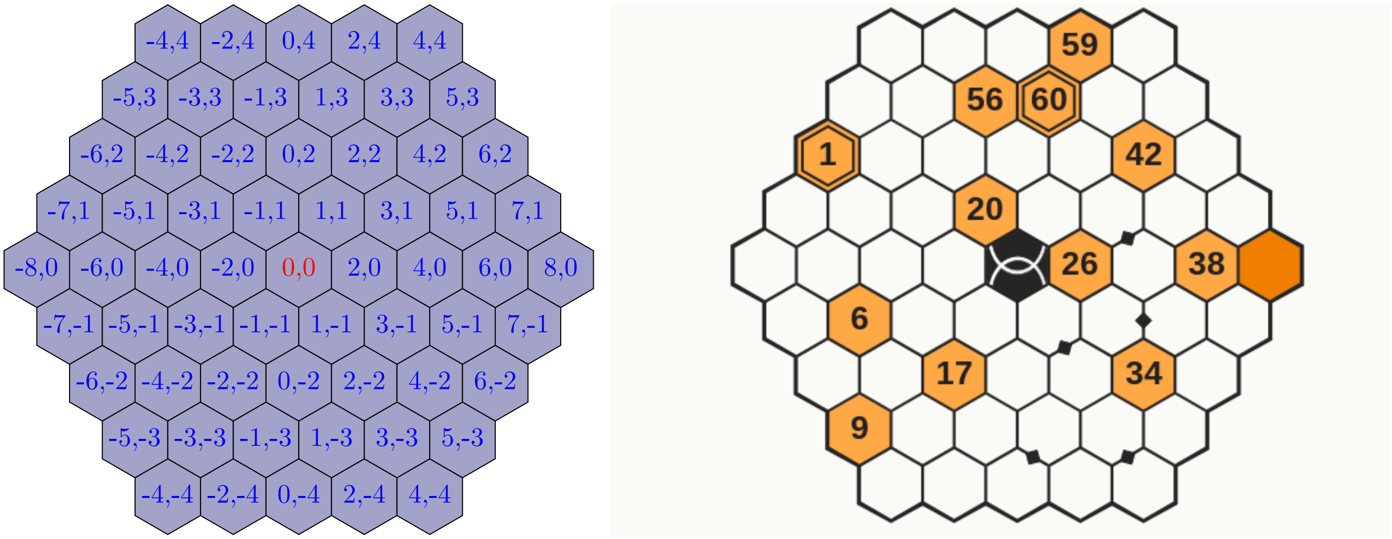

Figure 1: A Rikudo-61 grid with cells named by coordinates (left) and puzzle (right)

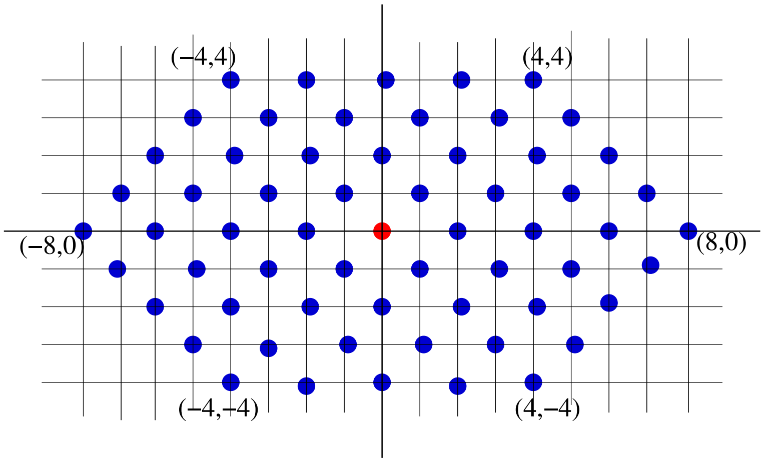

Figure 2: The coordinate system for the Rikudo-61 grid

Pre-filled Numbers A list of tuples (x, y, k) which shows that cell (x, y) has been pre-filled with the number k. For example the pre-filled items in the sample puzzle are

$$[(2,4,59),(-1,3,56),(1,3,60),(-6,2,1),(4,2,42),(-1,1,20),$$
$$(2,0,26),(6,0,38),(-5,-1,6),(-2,-2,17),(4,-2,34),(-5,-3,9)]$$

Links A list of tuples will be given denoting the links between neighbours. In the puzzle above the links are $[(3,1,4,0),(5,-1,3,-1),(1,-1,2,-2),(1,-3,0,-4),(3,-3,4,-4)]$.

Result The output is a list of 3-tuples of the form (x, y, k), where k, $1 \leq k < n$ $(n \in \{37, 61, 91\})$ consistent with pre-filled numbers and the links. All the numbers in the result should be from the range of allowed numbers with no repetitions. The center cell (cell $(0, 0)$) should contain the number -10.

What you have to do

- Create a file `<entry-number>.pl`

- Create a function `rikudo` to solve the puzzle. The function should take three arguments, namely *Size, Pre-filled* and *Links*, and return *Result* in the format defined above.

Submission Instruction

Submit a `<entry-number>.pl` file on Moodle https://moodle.iitd.ac.in

Important Notes

1. Do not change the name of the function given in this document. You are not even allowed to change upper-case letters to lower-case letters or vice-versa.
2. You may define any new functions you like besides those mentioned in the signature.
3. Follow the input output specification as given. We will be using automated scripts to execute the code for evaluation. In case of mismatch, you will be awarded zero marks.
4. All of your code should be in file `<entry-number>.pl`