Homework 03

Problem 1. (3.0pts) For each English sentence below, write the FOL sentence that best expresses its intended meaning. Use Cat(x) for "x is a cat," Mouse(x) for "x is a mouse," and Chases(x, y) for "x chases y."

1. Every cat chases every mouse.
2. For every cat, there is a mouse that the cat chases.
3. There is a cat who chases every mouse.
4. Some cat chases some mouse.
5. There is a mouse that every cat chases.
6. For every mouse, there is a cat who chases that mouse.

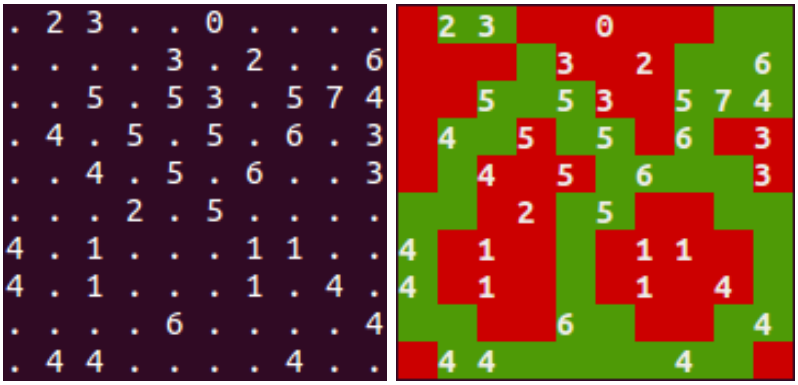
Problem 2. (2.0pts) Given a knowledge base as follows

$$P \vee Q,\ Q \rightarrow (R \wedge S),\ (P \vee R) \rightarrow U$$

Check whether each of the following sentences is entailed by KB using PL-Resolution

a) U b) $\neg U$

Problem 3. (5.0 pts) Given a Puzzle game board as below (Figure 01-a)



(a) **(b)**

Figure 01. *A sample of Puzzle game board*

Each cell in the board is assigned a number or a blank (dot cells). The player needs to fill all cells in green or red color so that the number of green cells around a cell with a number (including adjacent ones and itself) is exactly the number (Figure 01-b).

(a) (3.0 pts) Assign a logical variable to each cell in which the cell is green if the variable is True and red if False. Write CNF clauses to restrict the game rule above for the given cell below. Justify your procedure.

.	.	.
.	**2**	.
.	.	.

cell

a	**b**	**c**
d	**e**	**f**
g	**h**	**i**

variables

(b) (2.0 pts) Implement a program using Python and Glucose3 solver (Python-SAT: https://pypi.org/project/python-sat/0.1.1dev7/) to solve the game board in Figure 01-a.

- Input: **input.txt** (tab separated)
  - First line: 2 integers as the matrix shape
  - Rest line: a matrix of numbers and dots
- Output: **output.txt** (tab separated)
  - a matrix of letters G and R (green/red cells)
- *Source code: attached one file only, filename is formatted like [student_number].py*
  E.g. 12345678.py

Example for Glucose3

```
from pysat.solvers import Glucose3

g = Glucose3()
g.add_clause([-1, 2])
g.add_clause([-2, 3])

print (g.solve())
#True
print (g.get_model())
#[-1, -2, -3]
```