Report

PROJECT 2: Coloring Puzzle with PL

Course: CSC14003 – Artificial Intelligence

Class: 19CLC3

|Lecturers|

Nguyễn Ngọc Thảo

Lê Ngọc Thành

Hồ Thị Thanh Tuyến

HO CHI MINH CITY – 2021

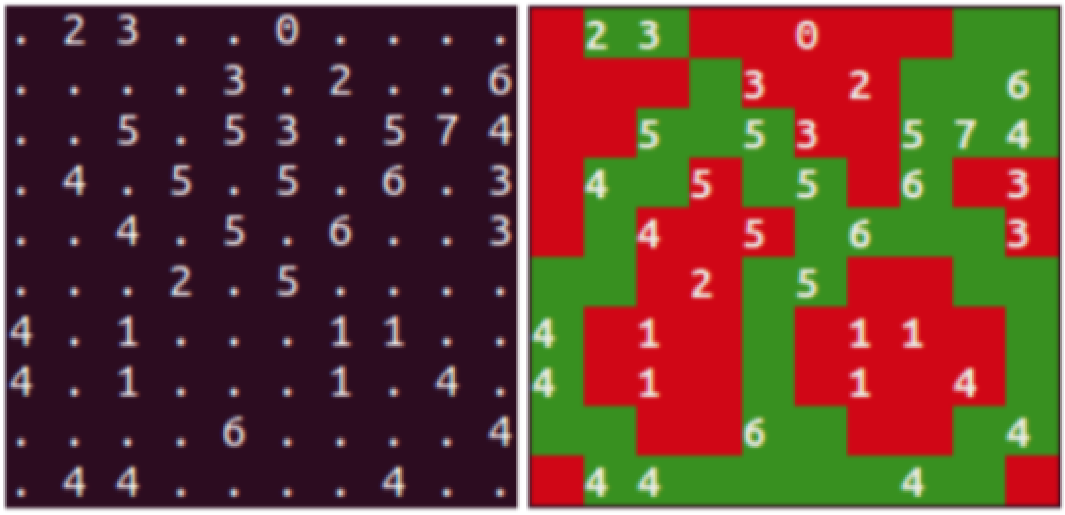
I. INFORMATION

STUDENT ID	FULL NAME	ASSIGNED WORK	COMPLETE PERCENT
19127003	NGUYỄN HỮU ĐẠT	Implement algorithms, back-end develope, write report	90%
19127395	PHAN ĐỨC HIỂN	GUI design for both all algorithms, Generate test case	100%
19127456	NGUYỄN THANH KIÊN	GUI design for both all algorithms, Generate test case, testing	100%

II. INTRODUCTION

1. Problem

We are asked to build a coloring puzzle solver by using the first order logic to CNF as described below: Given a matrix of size $m \times n$, where each cell will be a non-negative integer or zero (empty cell). Each cell is considered to be adjacent to itself and 8 surrounding cells. Our puzzle needs to color all the cells of the matrix with either blue or red, so that the number inside each cell corresponds to the number of blue squares adjacent to that cell.



```
. 2 3 . . 0 . . . .
. . . . 3 . 2 . . 6
. . 5 . 5 3 . 5 7 4
. 4 . 5 . 5 . 6 . 3
. . 4 . 5 . 6 . . 3
. . . 2 . 5 . . . .
4 . 1 . . . 1 1 . .
4 . 1 . . . 1 . 4 .
. . . . 6 . . . . 4
. 4 4 . . . . 4 . .
```

2. Progress

90% work completed

III. APPROACH

1. Generate CNFs automatically

- We use Octave (Matlab likely) to try to transform some propositional logics into reduced CNFs.
- We get some properties from the transformations and then prove it by induction.
    + Suppose we need to choose 3 out of 7 coins so that 3 are heads and 3 are tails.
    + We consider coins as literals a, b, c, d, e, f, g. If the literal has the value 0 it is tail, and 1 is head.
    + Suppose we have selected 3 heads (1-value), so when randomly picking 4 out of 7 sets, there will be at least 1 tails (0-value). So, first we get CNFs with 7C4 clauses, where the literals in the clauses are in the negative form. For example: (~a|~b|~c|~d) & (~a|~b|~c|~e) & (~a|~b|~c|~f) & ... & (~d|~e|~f|~g)
    + Next, suppose we have chosen 4 tails (0-value). Therefore, when randomly selecting 5 coins from 7 coins, there will be at least 1 heads (1-value). So we have CNFs again with 7C5 clauses, where the literals are in the assert form. Example: (a|b|c|d|e) & (a|b|c|d|f) & … & (c|d|e|f|g)
    + After all, when we combine the two constraints above to form the final CNFs, we will ensure that if there is a way to make it satisfy (True), the result will definitely be in the form of 3 heads - 4 tails. Imagine it as "*a birdcage"*, we bound on the roof and bound on the floor, so that the end result in the desired range is "*the bird"*.
- So if we need to color adjacent x in n cells, we will create CNFs with ***n*C*(x+1)*** clauses in negative form and ***n*C*(n-x+1)*** clauses in positive form. And this is also the shortened form of CNF.

2. Tools:

- Tkinter module to develope GUI
- pysat module to solve CNF automatically

- Octave (Matlab likely) to offline processing to prove inductively the reduced formula of CNFs

IV. EXPERIMENT

A has not been completed yet due to certain bugs, so this section only evaluates and compares the remaining 3 methods.*

- After implementing CNF automatically and applying pysat (or A* if completed), the execution time of the problem with a solution is less than 2s for pysat and 10s for A*. But in cases where the problem has no solution, A* will run for a long time, almost impossible, because A* is a complete search algorithm.
- Algorithms like Brute Force or Backtracking are effective only in small test cases (3x3, 5x5) with an average completion time of about 10 seconds. With larger test cases, the algorithm becomes slow and impossible.

V. CONCLUSION

- We have learned how to convert propositions into CNFs (depending on the complexity of the constraint).
- We learned how to use the powerful tools provided such as (Octave, Matlab, PySAT module) to quickly solve and implement algorithms and think about solutions efficiently.
- We have learned the methods of solving 2-SAT problems by algorithms such as DPLL, CDCL, ... in the process of searching related documents.

VI. REFERENCES

- Available documents of the project on Moodle

https://www.geeksforgeeks.org/how-to-use-thread-in-tkinter-python/

https://docs.python.org/3/library/tkinter.html