**MINISTRY OF EDUCATION AND TRAINING**

**VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY - VNUHCM**

**UNIVERSITY OF SCIENCE**

**FACULTY OF INFORMATION TECHNOLOGY**

­­­­­🙢🕮🙠 ­­­­­



**ARTIFICIAL INTELLIGENCE   
LAB 02: PL RESOLUTION**

**Instructor**: Bùi Tiến Lên, Trần Quốc Huy, Phạm Trọng Nghĩa

**Name**: Phạm Phú Toàn

**Identity**: 21127183

**class**: 21CLC08

Table of content

[**I.** **Checklist** 3](#_Toc143042097)

[**II.** **Describe Works** 3](#_Toc143042098)

[**III.** **Reference** 4](#_Toc143042099)

1. **Checklist**

|  |  |  |
| --- | --- | --- |
| No | Work | Score |
| 1 | Read the input data and successfully store it in some data structure | 10% |
| 2 | The output file strictly follows the lab specifications. | 10% |
| 3 | Implement the propositional resolution algorithm. | 20% |
| 4 | Provide a complete set of clauses and exact conclusion. | 30% |
| 5 | Five test cases: both input and output files. | 10% |
| 6 | Discussion on the algorithm’s efffiency and suggestions. | 20% |
| Total | | 100% |

→Complete all the requirements: 100%.

1. **Describe Works**

* *Input files* and *Output files*:

(Each output${number}.txt is corresponding to the file input${number}.txt, output1.txt and input1.txt is for the example in the Instruction PDF)

* *Src/Main.py*:
* **BFS, DFS, GBFS**, and **Asearch** are functions implemented the algorithms.

+ Input:

* + - N: size of the maze.
    - Entrance: the start node.
    - Goal: the end node.
    - Array: the array contains adjacency list

+ Ouput: explored Set, path, running time.

(Use module time to count time as microsecond.)

* **inputFile(), outPut()**: for reading, writing result to file. (Default: input.txt and output.txt)
* **checkIfLiteralIsOpposite(literal1, literal2)**:
  + check if two given literal is opposite.
  + For example:
    - ‘A’ and ‘-A’ are opposite, and function will return true.
    - ‘A’ and ‘B’ are not opposite, and function will return false.
* **returnOppositeQuery(query)**:
  + Return the opposite query (-α)
  + For example:
    - [‘A’] return [‘-A’]
    - [‘B or A’] return [‘-B’,’-A’]
    - [‘A or B’, ‘-C’] return [‘-A’,’-B’,’C’]
  + Note: the requirement is not really clear and there is no example when the query is more than two clauses or a clause with two literals, so I do with my own perspective.
* **checkIfSimilarClause(clause1, clause2):**
  + Return true when clause 1 is the same as clause 2
* **heuristicFunction()**: to count h(n) for each node. (Applied for the maze which is orderd like the maze in the task)

1. **Reference**

Bùi Tiến Lên. (n.d.). *Slide cơ sở trí tuệ nhân tạo.* Khoa công nghệ thông tin - Đại học khoa học tự nhiên.

*How to read numbers from file in Python*. (n.d.). Retrieved from StackOverFlow: https://stackoverflow.com/questions/6583573/how-to-read-numbers-from-file-in-python

*Python Tutorial*. (n.d.). Retrieved from W3 School: https://www.w3schools.com/python/default.asp