ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

----------&----------

CƠ SỞ TRÍ TUỆ NHÂN TẠO

LAB 02: LOGIC

SVTH: 21120280 Lý Minh Khuê Lớp 21_21

Thành phố Hồ Chí Minh - 2023

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

----------&----------

CƠ SỞ TRÍ TUỆ NHÂN TẠO

| Đề tài |

LOGIC

| Giáo viên hướng dẫn |

Thầy Lê Hoài Bắc
Thầy Nguyễn Bảo Long

Thành phố Hồ Chí Minh – 2023

LỜI CẢM ƠN

Lời đầu tiên, tụi em xin gửi cảm ơn chân thành đến thầy Lê Hoài Bắc và thầy Nguyễn Bảo Long, giảng viên bộ môn Cơ sở trí tuệ nhân tạo, Trường Đại học Khoa học Tự nhiên Thành phố Hồ Chí Minh đã giúp đỡ tận tình, tạo điều kiện để tụi em hoàn thành báo cáo một cách tốt nhất.

Sau đề tài nghiên cứu báo cáo đó, tụi em đã học hỏi và tích lũy được nhiều kinh nghiệm để hoàn thiện và phát triển bản thân. Tuy nhiên, do khả năng tiếp thu thực tế còn nhiều hạn hẹp, kiến thức chưa sâu rộng. Mặc dù bản thân đã cố gắng hết sức nhưng chắc chắn báo cáo khó tránh khỏi những thiếu sót, kính mong thầy xem xét và góp ý để bài báo cáo của tụi em được hoàn thiện và tốt hơn.

Xin chân thành cảm ơn!

CRITERIA

STT	CRITERIA	RATING
1	Read input data and store in suitable data structure	100%
2	Implementation of resolution method	100%
3	Inference process and results	100%
4	5 test cases for demo	100%
5	Evaluations about the advantages and disadvantages of resolution method for prepositional logic	100%
6	Report	100%

MỤC LỤC

GIỚI THIỆU

Bài báo cáo này nhằm mục đích thực hiện phương pháp **PL-Resolution** (**propositional logic resolution**) bằng cách sử dụng ngôn ngữ lập trình Python. Em sẽ thiết kế một hệ thống **Knowledge-base**(**KB**) để biểu diễn logic mệnh đề và triển khai phương pháp **PL-Resolution** để xác định xem KB có entail α không.

Điều này đòi hỏi chúng ta phải áp dụng phương pháp **PL-Resolution** một cách chính xác và hiệu quả. Báo cáo sẽ trình bày chi tiết việc triển khai của em, đồng thời đánh giá ưu và nhược điểm của phương pháp này.

PHƯƠNG PHÁP PL-RESOLUTION

1. Mã giả

PL-RESOLUTION(*KB*, *α*)

***Input: KB_clauses**, The knowledge base clauses in CNF representation*
***α**, The alpha clause in CNF representation*
***Output:** whether the knowledge base clauses entails the alpha clause.*

```
Assign CLAUSES as a set of CNF clauses of ¬α ∧ KB_clauses
loop
    Assign NEW_CLAUSES set as EMPTY
    foreach pair of clauses Ci, Cj in CLAUSES
        resolvents = resolve(Ci, Cj)
        if resolvents not in CLAUSES and resolvents not in
        NEW_CLAUSES and resolvents not EMPTY
            add resolvents to NEW_CLAUSES
    if NEW_CLAUSES is EMPTY
        return False
    Add NEW_CLAUSES to CLAUSES
    if '{}' in CLAUSES
        return True
end loop
```

Chú thích:

- ***CLAUSES***: là một tập hợp lưu mệnh đề dạng chuẩn CNF được khởi tạo từ

$$\neg\alpha \wedge KB_{clauses}.$$

- *resolvents*: là mệnh đề được tạo ra từ suy diễn dựa trên luật hợp giải
- ***NEW_CLAUSES***: là tập hợp lưu mệnh đề dạng chuẩn CNF lưu các *resolvents*.
- ***EMPTY CLAUSE '{}'***: là điều kiện để ***KB_clauses*** *entails* ***α***

2. Cài đặt phương pháp:

Lớp đối tượng:

```
class PLResolution
```

2.1. Các thuộc tính:

- **input_path (str):** đường dẫn tới file input.txt.
- **output_path (str):** đường dẫn tới file output.txt.
- **kb_clauses (list[list[str]]):** tập hợp mệnh đề **KB**.
- **alpha_clause (list[str]):** mệnh đề α .
- **loops (list[list]):** tập hợp các mệnh đề được suy diễn ra trong các vòng lặp.
- **entails (bool):** Cho biết các mệnh đề **KB** có mệnh đề α entails hay không.

2.2. Các phương thức:

- **__init__(self, input_path:str, output_path:str):**
- **custom_sort(self, literal:str):** Sort các mệnh đề đơn (literal) trong mệnh đề(clause) theo thứ tự bảng chữ cái.
- **parse_clause(self, clause_str:str):** Chuyển mệnh đề thành các mệnh đề đơn
- **read_input(self):** Đọc file input và trả về **alpha_clause** and **kb_clauses**.
- **negate_literal(self, literal:str):** Phủ định một mệnh đề đơn
- **write_output(self):** Xuất output data theo yêu cầu ra file output.txt.
- **pl_resolution(self):** Thực hiện thuật toán PL-Resolution.
  - is_valid_clause(clause: list): Kiểm tra xem mệnh đề đạt chuẩn CNF
  - is_complentary_literals(literal_1: str, literal_2: str): Kiểm tra hai literals có là phủ định của nhau không
  - resolve(clause1: list[list], clause2: list[list]): Thuật toán suy diễn theo luật hợp giải trả về một tập hợp các resolvents

2.3. Các điều kiện khi cài đặt thuật toán:

2.3.1. File input.txt:

- Dòng đầu tiên chứa mệnh đề α.

- Dòng thứ hai chứa N - số lượng mệnh đề trong **KB**.
- N dòng tiếp theo là các mệnh đề trong **KB**, một dòng cho mỗi mệnh đề.

2.3.2. File output.txt:

- Dòng đầu tiên chứa M1 - số lượng mệnh đề được tạo ra trong vòng lặp đầu tiên. M1 dòng tiếp theo mô tả các mệnh đề đó (bao gồm cả mệnh đề trống), mỗi dòng đại diện cho một mệnh đề. Mệnh đề trống được biểu diễn bởi "{}".
- Các vòng lặp tiếp theo cũng được biểu diễn như (M2, M3, …, Mn mệnh đề) trong vòng lặp đầu tiên.
- Dòng cuối cùng là mệnh đề kết luận, có nghĩa là "**KB** có entails α hay không?". In YES nếu **KB** entails $\boldsymbol{\alpha}$. In NO nếu ngược lại.
- Loại bỏ các mệnh đề dư thừa (các mệnh đề giống nhau trong cùng một vòng lặp hoặc các vòng lặp trước đó).

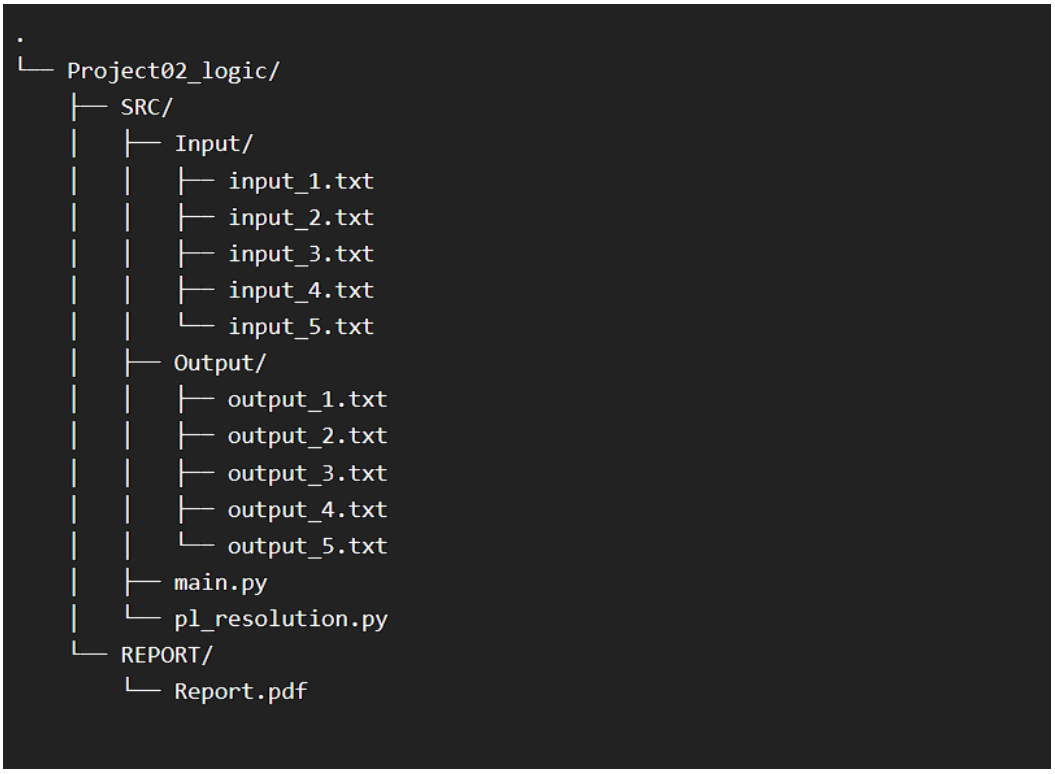
2.3.3. Hàm main():

- Đọc dữ liệu input và lưu trữ nó trong một cấu trúc dữ liệu phù hợp.
- Gọi hàm PL-Resolution để thực hiện thuật toán giải quyết.
- Xuất dữ liệu output theo định dạng yêu cầu.

2.3.4. Ghi chú:

- Lưu thông tin ngữ nghĩa về giá trị đúng và sai trong PL-RESOLUTION. Cần phủ định α.
- Các mệnh đề đơn(literal) trong một mệnh đề(clause) được sắp xếp theo thứ tự chữ cái.
- Điều kiện **KB** có entails $\boldsymbol{\alpha}$ không được kiểm tra trong mỗi vòng lặp khi tất cả các chân lý mới được tạo ra từ KB hiện tại.
- Những mệnh đề có định dạng $A \vee B \vee -B$ và giá trị ***True*** tương tự như $A \vee$ ***True*** có thể được loại bỏ.

2.4. Cây thư mục:



Hình 1. Cây thư mục

- Folder Input chứa các file testcase input.txt
- Folder Output chứa các file output được tạo ra từ các testcase
- main.py chứa hàm main xử lý các yêu cầu
- pl_resolution.py chứa class PLResolution

ĐÁNH GIÁ PHƯƠNG PHÁP PL-RESOLUTION

1. Ưu Điểm của Phương Pháp PL-Resolution

- PL-Resolution là phương pháp có tính hoàn thành(completeness), nghĩa là nó sẽ luôn tìm ra liệu **KB** có entails α hay không
- Luôn đưa ra kết luận chắc chắn(soundness). Không đưa ra thông tin sai lệch

2. Nhược điểm của Phương Pháp PL-Resolution

- Chi phí thời gian, tài nguyên tăng theo số mũ khi hệ thống **KB** lớn.

```
1:  Assign CLAUSES as a set of CNF clauses of ¬α ∧ KB_clauses
2:  loop
3:      Assign NEW_CLAUSES set as EMPTY
4:      foreach pair of clauses Ci, Cj in CLAUSES
5:          resolvents = resolve(Ci, Cj)
6:          if resolvents not in CLAUSES and resolvents not in
            NEW_CLAUSES and resolvents not EMPTY
7:              add resolvents to NEW_CLAUSES
8:      if NEW_CLAUSES is EMPTY
9:          return False
10:     Add NEW_CLAUSES to CLAUSES
11:     if '{}' in CLAUSES
12:         return True
13: end loop
```

Hình 2. Mã giả PL-Resolution

- Các cặp mệnh đề C_i, C_j được suy diễn nhiều lần mặc dù đã được suy diễn trong các vòng lặp trước đó. Và vấn đề này sẽ được lặp lại cho đến khi nào tìm ra được lời giải.

  Ví dụ:

  - **Giả thiết**: **KB** : C_1, C_2, C_3, C_4
  - **Vòng lặp 1:** C_5, C_6
  - **Vòng lặp 2:** C_7, C_8
  - **Vòng lặp 3:** C_9, C_{10}, C_{11}

Trong vòng lặp 1: Tất cả các cặp mệnh đề trong **KB** đã được suy diễn sinh ra các mệnh đề.

Trong vòng lặp 2: Các cặp mệnh đề từ **KB** lại được suy diễn với nhau như **{C_1, C_2}**,… Khi mà điều này là không cần thiết.

3. Giải pháp:

- Tại vòng lặp n, thì cặp mệnh đề C_i, C_j được suy diễn thì ($i < j$) và C_j là các mệnh đề được tạo ra từ vòng lặp $n - 1$.
- Từ ví dụ trên:
- Tại vòng lặp 3:
  - C_i = { C_1, C_2, C_3, C_4, C_5, C_6, C_7 }
  - C_j = { C_7, C_8}

TÀI LIỆU THAM KHẢO

Propositional Logic: Resolution and Limitations | Artificial Intelligence (engineeringenotes.com)Resolution Algorithm in Artificial Intelligence - GeeksforGeeks

lo.logic - Is propositional resolution a complete proof system? - Theoretical Computer Science Stack Exchange

Propositional Logic | Internet Encyclopedia of Philosophy (utm.edu)

Bài giảng của thầy Lê Hoài Bắc

Lab 2: Logic của thầy Nguyễn Bảo Long