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

**----o0o----**



BÁO CÁO ĐỒ ÁN 2

PROJECT LOGIC

**Lớp**: 20\_21

**Họ và tên**: Nguyễn Minh Hưng

**MSSV**: 20120491

Giảng viên: Lê Hoài Bắc

## Contents

[Contents 2](#_bookmark0)

1. [Tự đánh giá 3](#_bookmark1)
2. [Nội dung đồ án 3](#_bookmark2)
   * 1. [Giải thích code 3](#_bookmark3)
   1. [Input/Output 4](#_bookmark8)
      1. [Kịch bản 1 4](#_bookmark9)
      2. [Kịch bản 2 4](#_bookmark10)
      3. [Kịch bản 3 5](#_bookmark11)
      4. [Kích bản 4 6](#_bookmark12)
      5. [Kịch bản 5 6](#_bookmark13)
   2. [Đánh giá thuật toán 6](#_bookmark14)
      1. [Ưu điểm 6](#_bookmark15)
      2. [Khuyết điểm 6](#_bookmark16)
      3. [Đề xuất giải pháp 6](#_bookmark17)
3. [Tài liệu tham khảo: 6](#_bookmark18)

## **Tự đánh giá:**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đặc tả tiêu chí | Điểm | Hoàn thành |
| 1 | Đọc dữ liệu đầu vào và lưu trong cấu trúc dữ liệu phù hợp | 0.5 | 100% |
| 2 | Cài đặt giải thuật hợp giải trên logic mệnh đề | 1 | 100% |
| 3 | Các bước suy diễn phát sinh đủ mệnh đề và kết luận đúng | 2.5 | 100% |
| 4 | Tuân thủ mô tả định dạng của đề bài | 0.5 | 100% |
| 5 | Báo cáo test case và đánh giá | 0.5 | 100% |

## **Nội dung đồ án:**

## **Giải thích code**

## - Class Clause chứa alpha là giá trị đầu, clauses là danh sách các clause đã tạo, result là kết quả trả về

## - def set\_negative\_litral(literal): Hàm lấy giá trị đối của literal

## - def pl\_resolution (KB, alpha, fileName):

## B1: Tạo out\_clause bao gồm danh sách rỗng, alpha và result là False

## B2: Thêm alpha và KB ban đầu vào biến clauses kiểm tra

## B3: Tạo 1 vòng lặp while trong vòng lặp ta tạo biến temp\_clauses chứa danh sách các clause

## Kiểm tra. Điều kiện để dừng vòng lặp là temp\_clauses rỗng và trong temp\_clauses có chứa clause rỗng

## B4: Dùng 2 vòng lặp for để kiểm tra từng cặp clause trong clauses ứng với mỗi lần lặp gọi hàm def check\_two\_clause (clause1, clause2): để kiểm tra 2 clause

## Tạo biến check\_type để kiểm tra 2 clause

## Dùng 2 vòng lặp for để kiểm tra từng literal trong 2 clause: nếu như 2 literal đối nhau thì check\_type cộng thêm 1 và gán lần lượt 2 giá trị trong vòng lặp cho check1 và check2, nếu như check\_type > 1 thì cho hàm trả về (false, None)

## Sau khi kết thúc vòng lặp nếu như check\_type bằng 0 tức là không tồn tại các cặp literal nào đối nhau thì cho hàm trả về (false, None)

## Nếu như check\_type bằng 1 tức là có 1 cặp literal đối nhau thì ta cho hàm trả về giá trị True và - 1 clause mới không chứa cặp literal đối nhau ở trên nếu độ dài của 2 clause lớn hơn 1

## - () nếu độ dài của 2 clause bằng 1

## B5: Sau khi gọi hàm check\_two\_clause nếu giá trị trả về bằng True và clause trả về không tồn tại trong temp\_clauses thì thêm clause vừa trả về vào temp\_clause

## B6: Sau khi thực hiện 2 vòng lặp để kiểm tra kiểm tra xem nếu như

## temp\_clause rỗng thì gán False cho result của out\_clause và kết thúc vòng lặp while

## Trong danh sách phần tử của temp\_clause nếu như tồn tại 2 literal đối nhau thì gán giá trị True cho result của out\_clause. Còn nếu như không tồn tại 2 literal đối nhau thì thêm temp\_clause vào danh sách clause của out\_clause

## B5: Tiến hành ghi kết quả ra các file output bằng cách gọi hàm write\_file trong class HandleFile

## **Input/Output**

## Kịch bản 1

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Ghi chú** |
| -A OR C | 2 |  |
| 4 | A OR -C | (A OR -B) hợp giải với (B OR -C) |
| -A OR B | B | (-A OR B) hợp giải với (A) |
| B OR -C | 1 |  |
| A OR -B OR C | A OR C | (A OR -B OR C) hợp giải với (B) |
| A OR -B | 2 |  |
|  | A OR B | (B OR -C) hợp giải với (A OR C) |
|  | B OR C | (A OR C) hợp giải với (-A OR B) |
|  | 0 | Không sinh ra mệnh đề mới |
|  | NO | KB KHÔNG entail |

## Kịch bản 2

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Ghi chú** |
| -A OR B | 9 |  |
| 6 | -A | (-A OR B) hợp giải với (-B) |
| A | -D OR E | (-B OR -D OR E) hợp giải với (B OR E) |
| -A OR B | C OR -D OR E | (-B OR C OR -D) hợp giải với (B OR E) |
| -B OR COR -D | -A OR C OR -D | (-A OR B) hợp giải với (-B OR C OR -D) |
| -B OR -D OR E | -B OR C OR E | (-B OR -D OR E) hợp giải với (C OR D) |
| B OR E | -B OR C | (-B OR C OR -D) hợp giải với (C OR D) |
| C OR D | B | (-A OR B) hợp giải với (A) |
|  | E | (B OR E) hợp giải với (-B) |
|  | -A OR -D OR E | (-A OR B) hợp giải với (-B OR -D OR E) |
|  | 6 |  |
|  | C OR -D | (A) hợp giải với (-A OR C OR -D) |
|  | -A OR C | (-A OR B) hợp giải với (-B OR C) |
|  | -A OR C OR E | (-A OR B) hợp giải với (-B OR C OR E) |
|  | C | (B) hợp giải với (-B OR C) |
|  | C OR E | (B OR E) hợp giải với (-B OR C) |
|  | {} | (A) hợp giải với (-A) |
|  | YES | KB entails α |

## 3 Kịch bản 3

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Ghi chú** |
| A OR D | 4 |  |
| 4 | B OR -C | (B OR -C OR D) hợp giải với (-D) |
| -A OR B | -A OR C | (-A OR B) hợp giải với (-B OR C |
| B OR -C OR D | -C OR D OR E | (B OR -C OR D) hợp giải với (-B OR E) |
| C OR -B | -A OR E | (-A OR B) hợp giải với (-B OR E) |
| -B OR E | 4 |  |
|  | -A OR D OR E | (-A OR C) hợp giải với (-C OR D OR E) |
|  | -C OR E | (-B OR E) hợp giải với (B OR -C) |
|  | -B OR D OR E | (-B OR C) hợp giải với (-C OR D OR E) |
|  | -A OR B OR D | (B OR -C OR D) hợp giải với (-A OR C) |
|  | 1 |  |
|  | -A OR C OR D | (-B OR C) hợp giải với (-A OR B OR D) |
|  | 0 | Không sinh ra mệnh đề mới |
|  | NO | KB KHÔNG entail |

4 Kịch bản 4

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Ghi chú** |
| E | 4 |  |
| 6 | -A OR D OR E | (-A OR C) hợp giải với (-C OR D OR E) |
| A | -C OR D | (-C OR D OR E) hợp giải với (-E) |
| B | C | (A) hợp giải với (-A OR C) |
| -A OR C | -C OR E | (-C OR D OR E) hợp giải với (-D) |
| -A OR B | 6 |  |
| -C OR D OR E | D OR E | (A) hợp giải với (-A OR D OR E) |
| -D | -A OR E | (-A OR C) hợp giải với (-C OR E) |
|  | E | (C) hợp giải với (-C OR E) |
|  | -C | (-D) hợp giải với (-C OR D) |
|  | -A OR D | (-A OR C) hợp giải với (-C OR D) |
|  | D | (C) hợp giải với (-C OR D) |
|  | 2 |  |
|  | -A | (-A OR C) hợp giải với (-C) |
|  | {} | (-D) hợp giải với (D) |
|  | YES | KB entails α |

5 Kịch bản 5

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Ghi chú** |
| A OR C | 3 |  |
| 4 | -C OR -D | (B OR -D) + (-B OR -C) |
| B OR -D | B | (B OR -D) + (D) |
| -A OR C | -A OR -B | (-A OR C) + (-B OR -C) |
| -B OR -C | 1 |  |
| D | -A OR -D | (B OR -D) + (-A OR -B) |
|  | 0 | Không sinh ra mệnh đề mới |
|  | NO | KB KHÔNG entail |

# **Đánh giá thuật toán**

## Ưu điểm

## Vì thuật toán duyệt từ KB ban đầu và duyệt qua từng phần tử nên thuật toán luôn đảm vào tính tin cậy và việc duyệt qua từng phần tử nên thuật toán sẽ kiểm tra cho đến khi nào có kết quả thì th chính về thế luôn đảm bảo rằng kết quả thuật toán sẽ luôn là YES hoặc NO

## Khuyết điểm

* + - Vì thuật toán giải quyết vấn đề bằng cách duyệt qua tất cả các mệnh đề bằng cách sử dụng nhiều vòng lặp lồng nhau nên sẽ rất mất thời gian trong việc kiểm tra cũng như số lượng mệnh đề tạo ra là rất lớn
    - Thuật toán chỉ áp dụng cho mệnh đề dạng CNF và định dạng đầu vào chính xác

## Đề xuất giải pháp

* + - Chính vì việc lập đi qua nhiều phần tử như thế nên thời gian để thuật toán xử lý là khá lâu. Giải pháp ở đây là thay vì kiểm tra 2 clause dài thì ta sẽ ưu tiên với việc kiểm tra 2 clause ngắn trc để cho việc có kết quả 1 cách nhanh hơn với các mệnh đề tạo ra trong quá trình kiểm tra là ít hơn

## **Tài liệu tham khảo:**

* Giáo trình Cơ sở trí tuệ nhân tạo của Trường Đại học Khoa học Tự nhiên, ĐHQG TP.HCM
* <https://www.youtube.com/watch?v=xLyL7RcPaVQ>
* Sách Artificial Intelligence: A Modern Approach, Third Edition, Chương 7, Hình 7.12, hàm PL-RESOLUTION