TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

**KHOA CÔNG NGHỆ THÔNG TIN**



**Trương Anh Tuấn – 21120589**

**BÁO CÁO LAB02: LOGIC**

**Lớp: Cơ sở trí tuệ nhân tạo – 21\_22**

MỤC LỤC

[**I.** **TEST CASES** 3](#_Toc151713746)

[**II.** **GIẢI THÍCH HÀM** 3](#_Toc151713747)

[**1.** **def readInput(filename)** 3](#_Toc151713748)

[**2.** **def appendNegateAlphaToKB(alpha, KB)** 4](#_Toc151713749)

[**3.** **def are\_two\_opposite\_literals(l1, l2)** 4](#_Toc151713750)

[**4.** **def resolveTwoClauses(clause\_one, clause\_two)** 4](#_Toc151713751)

[**5.** **def PL\_RESOLUTION(clauses)** 4](#_Toc151713752)

[**6.** **def writeOutput(isSuccess, new\_clauses\_array)** 5](#_Toc151713753)

[**III.** **ƯU ĐIỂM VÀ NHƯỢC ĐIỂM CỦA PHƯƠNG PHÁP GIẢI QUYẾT MỆNH ĐỀ** 6](#_Toc151713754)

[**1.** **Ưu điểm** 6](#_Toc151713755)

[**2.** **Nhược điểm** 6](#_Toc151713756)

[**3.** **Đề xuất cách giải quyết** 6](#_Toc151713757)

[**IV.** **THAM KHẢO** 7](#_Toc151713758)

# **TEST CASES**

- Có 5 test cases, các test case nằm trong thư mục “input”.

A screenshot of a computer program

Description automatically generated- Mỗi test case nằm trong một tập tin “input<number>.txt”, ví dụ test case thứ nhất sẽ nằm trong tập tin, “input1.txt”, tương tự các test case tiếp theo:

**A screenshot of a computer

Description automatically generated**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

# **GIẢI THÍCH HÀM**

## **def readInput(filename)**

- Hàm này để đọc file input, các file input nằm trong thư mục input. Ví dụ tập tin “input1.txt” sẽ có đường dẫn tương đối là “input/input1.txt”

- Đọc dòng dầu tiên để lấy 𝛼, sau đó tách các literal trong 𝛼 thành các phần tử riêng của một mảng.

- Đọc dòng tiếp theo để lấy N mệnh đề trong KB (N là số lượng mệnh đề trong KB).

- Đọc N dòng tiếp theo, mỗi dòng sẽ lấy một mệnh đề, sau đó tách các literal trong mệnh đó thành các phần tử riêng của một mảng.

- Ví dụ file input có định dạng sau:

A screenshot of a computer program

Description automatically generated

- Sau khi đọc ta được kết quả:

**A black background with white text

Description automatically generated**

## **def appendNegateAlphaToKB(alpha, KB)**

- Hàm này để nối negate 𝛼 vào trong KB, tạo ra các mệnh đề mới (bao gồm cả KB), ta được kết quả.

## **def are\_two\_opposite\_literals(l1, l2)**

- Hàm này để kiểm xem hai literals có đối ngẫu hay không, ví dụ ‘B’ và ‘-B’ là đối ngẫu, ‘B’ và ‘C’ là không đối ngẫu, ‘B’ và ‘B’ là không đối ngẫu.

## **def resolveTwoClauses(clause\_one, clause\_two)**

- Hàm này để hợp giải hai mệnh đề, và kiểm tra xem có tạo ra một mệnh đề mới hay không, và sắp xếp các literal trong mệnh đó theo thứ tự bảng chữ cái.

- Ví dụ hợp giải hai mệnh đề ‘-R OR -T OR -Z’ và ‘R OR S OR -T, ta được một mệnh đề mới:



## **def PL\_RESOLUTION(clauses)**

- Hàm này sẽ chạy hai vòng lặp lồng nhau để hợp giải hai mệnh đề trong mảng ‘clauses’ được tạo ra từ hàm def appendNegateAlphaToKB(alpha, KB).

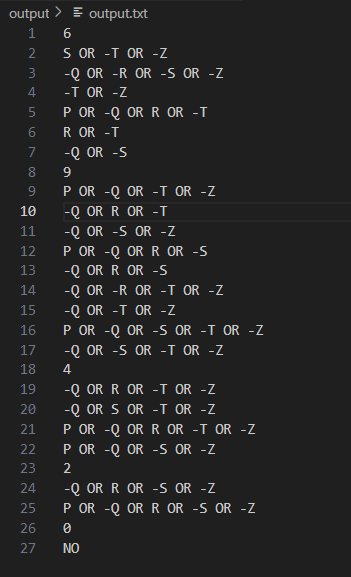
- Nếu trong lúc chạy hai vòng lặp này mà sinh ra một mệnh đề rỗng thì KB entails 𝛼.

- Nếu chạy xong hai vòng lặp này mà không được thêm mệnh đề nào thì KB not entail 𝛼.

- Nếu chạy xong hai vòng lặp này sinh ra các mệnh đề mới không rỗng, ta sẽ nối các mệnh đề đó vào mảng ‘clauses’, rồi tiếp tục chạy hai vòng lặp lồng nhau để hợp giải hai mệnh đề trong mảng ‘clauses’ và kiểm tra các điều kiện trên.

## **def writeOutput(isSuccess, new\_clauses\_array)**

- Hàm này để viết kết quả ra file output với cấu trúc theo đề yêu cầu, ví dụ với file input như trên thì khi chạy chương trình sẽ được file output như sau (file có đường dẫn tương đối là “output/output.txt”:



# **ƯU ĐIỂM VÀ NHƯỢC ĐIỂM CỦA PHƯƠNG PHÁP GIẢI QUYẾT MỆNH ĐỀ**

## **Ưu điểm**

- Dựa vào các luật suy diễn và KB, ta sẽ luôn chứng minh được tính đúng dắn hoặc tính không đúng dắn của một biểu thức logic.

- Phương pháp giải quyết mệnh đề thường được sử dụng trong các mô hình cơ bản và giáo trình đào tạo vì tính đơn giản và hiệu quả của nó.

## **Nhược điểm**

- Nếu như biểu thức phức tạp và KB quá lớn, phương pháp giải quyết mệnh đề có thể gặp khó khăn và mất thời gian khi chứng minh tính đúng hoặc không đúng của một biểu thức.

- Hiệu suất của phương pháp giải quyết có thể bị giảm nếu không áp dụng các kỹ thuật tối ưu hóa và chiến lược chọn lọc tốt.

- Ở thuật toán gốc của PL-RESOLUTION, khi chạy hai vòng lặp, mỗi vòng lặp sẽ duyệt qua hết các mệnh đề (bao gồm KB và các mệnh đề mới), như thế sẽ làm cho độ phức tập thời gian của thuật toán quá lớn, và sẽ sinh ra nhiều mệnh đề trùng nhau.

## **Đề xuất cách giải quyết**

Ở lần lặp thứ nhất khi hợp giải các mệnh đề trong KB, ta sinh ra được các mệnh đề mới, thì ở lần lặp thứ hai trở đi, ta chỉ cần hợp giải các mệnh đề trong KB và các mệnh đề mới được sinh ra, không cần phải hợp giải lại các mệnh đề trong KB nữa.

# **THAM KHẢO**

* <https://www21.in.tum.de/teaching/sar/SS20/3.pdf>
* <https://www.techfak.unibielefeld.de/ags/wbski/lehre/digiSA/WS0506/MDKI/Vorlesung/vl06_logicagent2.pdf>