

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



**BÁO CÁO ĐỒ ÁN 02**  
**LOGIC**

Môn học: Cơ sở trí tuệ nhân tạo

Giáo viên hướng dẫn:

Lê Hoài Bắc  
Nguyễn Ngọc Băng Tâm  
Nguyễn Duy Khánh

Sinh viên thực hiện:

Dũ Quốc Huy - 20120101

Thành phố Hồ Chí Minh, tháng 12 năm 2022

# Mục Lục

<b>1</b>	<b>Đánh giá theo các tiêu chí . . . . .</b>	<b>2</b>
<b>2</b>	<b>Mô tả chương trình . . . . .</b>	<b>2</b>
2.1	Đọc dữ liệu . . . . .	2
2.2	Mở file để ghi . . . . .	2
2.3	Thực hiện hợp giải . . . . .	2
2.4	Một số phương thức hỗ trợ đối tượng . . . . .	3
<b>3</b>	<b>Input và output của các testcase . . . . .</b>	<b>3</b>
3.1	Test case 0 . . . . .	4
3.2	Test case 1 . . . . .	4
3.3	Test case 2 . . . . .	5
3.4	Test case 3 . . . . .	5
3.5	Test case 4 . . . . .	5
<b>4</b>	<b>Đánh giá giải thuật hợp giải . . . . .</b>	<b>5</b>
4.1	Ưu điểm . . . . .	5
4.2	Khuyết điểm . . . . .	6
4.3	Đề xuất giải pháp khắc phục . . . . .	6
	<b>Tài liệu tham khảo . . . . .</b>	<b>6</b>

## 1. Đánh giá theo các tiêu chí

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%

## 2. Mô tả chương trình

Quá trình thực hiện hợp giải các mệnh đề của 1 test case sẽ gồm 3 giai đoạn được trình bày như dưới đây

### 2.1. Đọc dữ liệu

Chương trình thực hiện đọc toàn bộ file trong thư mục INPUT sau đó gọi hàm để xử lý Đọc một test case theo các quy ước được quy định trong yêu cầu đề bài:

- Đọc dòng đầu chứa câu  $\alpha$
- Dòng thứ 2 là số nguyên N - số mệnh đề có trong KB
- N dòng tiếp theo mỗi dòng là một mệnh đề trong KB

Với mỗi dòng mệnh đề được đọc vào (của KB và câu alpha), chương trình sẽ thực hiện xóa các khoảng trắng thừa, cắt chuỗi thành từng literal dựa vào từ khóa OR. Chương trình lưu các mệnh đề bằng list các Clause (xây dựng đối tượng), với mỗi Clause sẽ là một list các literal được sắp xếp theo thứ tự chữ cái A-Z.

Sau khi đã có được danh sách các Clause từ file input, ta thực hiện với bước tiếp theo.

### 2.2. Mở file để ghi

Chương trình sẽ thực hiện tạo và mở một thư mục output tương ứng với input để có thể ghi dữ liệu có được trong lúc xử lý.

### 2.3. Thực hiện hợp giải

Chương trình xử lý hợp giải nhờ vào một vài phương thức được khai báo trong đối tượng Clause như merge chỉ để hợp giải 2 mệnh đề.

Xử lý một vòng lặp lớn để liên tục xử lý các mệnh đề. Vòng lặp chỉ dừng khi không thể tạo ra mệnh đề mới hoặc trong số mệnh đề mới có mệnh đề rỗng.

trong vòng lặp lớn đó. Chúng ta sẽ khai báo ra một list các mệnh đề mới (newClauses), một mệnh đề mới có được khi thực hiện hợp giải 2 mệnh đề trong list mệnh đề cũ sẽ được theo vào list mệnh đề mới nếu nó:

- Là một đối tượng mệnh đề được tạo ra từ 2 mệnh đề cũ
- Không phải là mệnh đề vô ích
- Không nằm trong danh sách các mệnh đề cũ
- Không nằm trong danh sách các mệnh đề mới.

Khi tạo ra được list các mệnh đề mới, ta sẽ thực hiện việc ghi ra màn hình số lượng của list ra file output.

Nếu list các mệnh đề mới là rỗng thì trả về False (KB không entails  $\alpha$ ), ngược lại thì in từng phần tử trong list mệnh đề mới ra. Nếu list các mệnh đề mới tồn tại một mệnh đề rỗng thì trả về TRUE (KB entails  $\alpha$ )

Khi được thoát ra khỏi vòng lặp lớn, ta chỉ việc in kết quả YES/ NO tương ứng với giá trị trả về ở trên.

## 2.4. Một số phương thức hỗ trợ đối tượng

**\_\_init\_\_(self, clause:list)**: giúp khởi tạo một đối tượng mới, với đầu vào là list các literal, khi khởi tạo sẽ thực hiện format lại dữ liệu

**\_\_str\_\_(self)**: in đối tượng ra file output, việc in literal sẽ ngăn cách bằng từ khóa OR

**\_\_eq\_\_(self, obj)**: kiểm tra xem 2 đối tượng có bằng nhau hay không

**format(self)**: được thực hiện khi khởi tạo đối tượng, sắp xếp các literal của mệnh đề theo thứ tự A-Z

**nonsense(self)**: kiểm tra mệnh đề không phải là mệnh đề vô ích, trả ra False nếu là mệnh đề vô ích và ngược lại.

**negation(literal:str)**: hàm quan trọng cho việc tạo ra một literal là NOT của literal truyền vào. Việc này hỗ trợ để xác định có phải mệnh đề vô ích hay không, chuyển đổi các literal trong mệnh đề  $\alpha$

## 3. Input và output của các testcase

Các file input được lưu dạng .txt ở thư mục INPUT

Các file output sẽ được tạo ra tương ứng trong thư mục OUTPUT

### 3.1. Test case 0

input0.txt	output0.txt	Ghi chú
-A	5	
5	-A	(-A OR B) hop giai voi (-B)
B OR -C OR D	B	(-A OR B) hop giai voi (negative of -A)
A OR -B OR C	-C OR D	(B OR -C OR D) hop giai voi (-B)
-B	-B OR C OR D	(A OR -B OR C) hop giai voi (-A OR D)
-A OR D	D	(-A OR D) hop giai voi (negative of -A)
	7	
	-A OR C OR D	(-A OR B) hop giai voi (-B OR C OR D)
	-B OR C	(A OR -B OR C) hop giai voi (-A)
	A OR C	(A OR -B OR C) hop giai voi (B)
	A OR -B OR D	(A OR -B OR C) hop giai voi (-C OR D)
	{}	(-B) hop giai voi (B)
	C OR D	(B) hop giai voi (-B OR C OR D)
	-B OR D	(-C OR D) hop giai voi (-B OR C OR D)
	YES	KB entails alpha vi ton tai menh de rong trong KB

### 3.2. Test case 1

input1.txt	output1.txt	Ghi chú
-C	3	
5	B	(A) hop giai voi (-A OR B)
A	-A OR C OR D	(-A OR B) hop giai voi (-B OR C OR D)
-A OR B	-B OR C OR -E	(-B OR C OR D) hop giai voi (-D OR -E)
-B OR C OR D	3	(A OR -B OR C) hop giai voi (-A OR D)
-D OR -E	C OR D	(A) hop giai voi (-A OR C OR D)
-E	-A OR C OR -E	(-A OR B) hop giai voi (-B OR C OR -E)
	C OR -E	(B) hop giai voi (-B OR C OR -E)
	0	(A OR -B OR C) hop giai voi (-A)
	NO	KB KHONG entail alpha vi khong phat sinh duoc menh de moi va khong tim thay menh de rong

### 3.3. Test case 2

input2.txt	output2.txt	Ghi chú
-A OR E	6	
5	-A OR C	(-A OR B) hop giai voi (-B OR C)
-A OR B	-A	(-A OR B) hop giai voi (-B)
B OR -C	B	(-A OR B) hop giai voi (negative of -A)
-B OR C	-C	(B OR -C) hop giai voi (-B)
-B	B OR -D OR E	(B OR -C) hop giai voi (C OR -D OR E)
C OR -D OR E	C OR -D	(C OR -D OR E) hop giai voi (negative of E)
	5	
	B OR -D	(B OR -C) hop giai voi (C OR -D)
	C	(-B OR C) hop giai voi (B)
	{}	(-B) hop giai voi (B)
	-D OR E	(-B) hop giai voi (B OR -D OR E)
	-D	(-C) hop giai voi (C OR -D)
	YES	KB entails alpha vi ton tai menh de rong trong KB

### 3.4. Test case 3

input3.txt	output3.txt	Ghi chú
B OR -C	4	
5	B OR C	(-A OR B OR C) hop giai voi (A)
-A OR B OR C	-A OR C	(-A OR B OR C) hop giai voi (negative of B)
-A OR D	D	(-A OR D) hop giai voi (A)
C OR D OR F	F	(B OR F) hop giai voi (negative of B)
B OR F	0	
A	NO	KB KHONG entail alpha vi khong phat sinh duoc menh de moi va khong tim thay menh de rong

### 3.5. Test case 4

input4.txt	output4.txt	Ghi chú
A OR -C OR -D	3	
5	-A OR B OR D	(-A OR C OR D) hop giai voi (-A OR B OR -C)
-A OR C OR D	C OR D	(-A OR C OR D) hop giai voi (A OR D)
-A OR B OR -C	-A OR B	(-A OR B OR -C) hop giai voi (negative of -C)
B OR D	0	
B OR -C OR D	NO	KB KHONG entail alpha vi khong phat sinh duoc menh de moi va khong tim thay menh de rong
A OR D		

## 4. Đánh giá giải thuật hợp giải

### 4.1. Ưu điểm

Giải thuật dễ hiểu, dễ thực hiện, dựa vào quá trình giải thủ công, đem lại độ chính xác cao. Lưu từng mệnh đề (là list các literal) bằng đối tượng giúp chương trình trở nên khá đơn giản dễ dàng truy xuất và xử lí.

## 4.2. Khuyết điểm

Giải thuật sẽ thực hiện rất nhanh với những test case có số lượng mệnh đề nhỏ. Với số lượng mệnh đề lớn thì giải thuật thực hiện khá lâu do sử dụng nhiều vòng lặp lồng nhau cho quá trình hợp giải từng 2 mệnh đề.

## 4.3. Đề xuất giải pháp khắc phục

Em sẽ cố gắng để cải tiến giải thuật, hạn chế sử dụng vòng lặp hoặc tìm những giải thuật khác có tốc độ xử lý tốt hơn nhưng vẫn đảm bảo độ chính xác. Đồng thời tìm cách cấu trúc dữ liệu để thực hiện lưu từng clause hoặc chuỗi literal khác để quá trình thực hiện đọc và xử lý dữ liệu từ file input được nhanh hơn.

## Tài liệu tham khảo

- [1] Bài giảng của thầy Lê Hoài Bắc và hướng dẫn thực hiện đồ án của thầy Nguyễn Ngọc Băng Tâm, thầy Nguyễn Duy Khánh
- [2] Sách Artificial Intelligence: A Modern Approach, Third Edition, Chương 7, Hình 7.12, hàm PL-RESOLUTION <https://zoo.cs.yale.edu/classes/cs470/materials/aima2010.pdf>