

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



**BÁO CÁO LAB 2 - LOGIC**

Họ tên: Nguyễn Quang Huy

MSSV: 20120497

**| Giáo viên hướng dẫn |**  
**Nguyễn Duy Khánh**

Cơ sở trí tuệ nhân tạo

Thành phố Hồ Chí Minh , tháng 12 năm 2022

# Mục lục

<b>1. TỰ ĐÁNH GIÁ.....</b>	<b>3</b>
<b>2. GIẢI THÍCH CÀI ĐẶT.....</b>	<b>4</b>
<b>2.1. Biểu diễn dữ liệu.....</b>	<b>4</b>
<b>2.2. Mô tả thuật toán và cài đặt các hàm .....</b>	<b>4</b>
<b>3. ĐÁNH GIÁ THUẬT TOÁN.....</b>	<b>6</b>
<b>4. KIỂM THỬ .....</b>	<b>7</b>
<b>4.1. Test case 1 .....</b>	<b>7</b>
<b>4.2. Test case 2 .....</b>	<b>8</b>
<b>4.3. Test case 3 .....</b>	<b>9</b>
<b>4.4. Test case 4 .....</b>	<b>10</b>
<b>4.5. Test case 5 .....</b>	<b>10</b>
<b>5. TÀI LIỆU THAM KHẢO .....</b>	<b>12</b>

## 1. TỰ ĐÁNH GIÁ

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

## 2. GIẢI THÍCH CÀI ĐẶT

### 2.1. Biểu diễn dữ liệu

- Một clause được biểu diễn bằng một frozenset của python và các literal được biểu diễn là các string bên trong frozenset đó. Nguyên nhân sử dụng frozenset là do frozenset là hashable ( giúp cho nó có thể được nằm trong một set), không thể thay đổi giá trị ( ta không có nhu cầu đổi các giá trị literal bên trong) và tránh trùng lặp dữ liệu ( sẽ không có literal trùng nhau)

- Một knowledge base sẽ được biểu diễn bằng một set chứa nhiều clause bên trong để tránh trùng lặp các clause.

Ví dụ: một KB như sau:

A OR -C

-B OR D

=> Biểu diễn trong python: `KB = {frozenset({'A', '-C'}), frozenset({'-B', 'D'})}`

- Các kết quả hợp giải đều được biểu diễn giống như knowledge base ( sử dụng set)

### 2.2. Mô tả thuật toán và cài đặt các hàm

- `PL_RESOLVE (clause1, clause2)` : trả về một set chứa các clause có thể tạo ra bằng cách hợp giải 2 clause đầu vào

+ Các bước thuật toán:

- Bước 1: Khởi tạo biến resolvents chứa kết quả bằng set rỗng
- Bước 2: Duyệt qua tất cả các literal bên trong clause1 nếu như tìm thấy literal đối ngẫu với nó bên clause2 thì tiến đến bước 3. Sau đó đến bước 5
- Bước 3: Tạo một resolvent là một clause biểu diễn bằng một frozenset chứa các literal bên trong clause1 và clause2 mà khác với hai literal đang xét ở bước 2 ( hợp giải 2 clause đối ngẫu nhau)
- Bước 4: Kiểm tra xem resolvent có là một tautology không, nếu không thì cho vào set resolvents và quay về bước 2 còn không thì quay về bước 2
- Bước 5: Trả về resolvents

- PL\_RESOLUTION (KB, alpha): trả về một list chứa các bước của hợp giải và kết quả của câu hỏi “KB entails  $\alpha$ ?”
  - + Các bước thuật toán:
    - Bước 1: Khởi tạo biến clauses là một set chứa các clause ban đầu và giá trị bằng KB hợp với set chứa phủ định của  $\alpha$ , new là set chứa các clause tạo ra, result là một list chứa các set là những clause được tạo ra trong mỗi vòng lặp, biến boolean entail là kết quả của câu hỏi “KB entails  $\alpha$ ?”
    - Bước 2: Tạo vòng lặp vô hạn, mỗi bước được mô tả như sau:
      - Bước 2.1: Duyệt qua tất cả các cặp clause và lấy các clause là kết quả hợp giải thông qua hàm PL\_RESOLVE cho vào biến resolvents.
      - Bước 2.2: Cho set new hợp với resolvents
      - Bước 2.3: Tạo biến new\_clauses là set chứa các clause tạo ra trong vòng lặp tạo ra bằng cách lấy set clauses trừ đi set new.
      - Bước 2.4: Thêm new\_clauses vào result
      - Bước 2.5: Nếu như kích thước của new\_clauses bằng 0 thì nghĩa là không phát sinh mệnh đề mới, trả kết quả bao gồm result và entail = False và dừng thuật toán. Nếu new\_clauses chứa clause rỗng thì trả kết quả bao gồm result và entail = True và dừng thuật toán. Trường hợp còn lại thì tiếp tục vòng lặp.

Một số hàm hỗ trợ:

- read\_input\_from\_file (path): đọc file từ đường dẫn và trả về KB, alpha
- check\_tautological\_clause (clause): kiểm tra một clause có là một tautology
- negation\_of\_clause (clause): trả về một set chứa phủ định của clause
- complementary\_literal (literal): trả về literal đối ngẫu
- export\_output\_to\_file (path, result, entail): xuất output ra file ( trong hàm này có sort lại các literal trong một clause cho đúng định dạng để xuất ra file)
- compare(literal\_1, literal\_2); compare\_string(string\_1, string\_2): hai hàm hỗ trợ có chức năng so sánh hai literal để sort lại các literal.

### 3. ĐÁNH GIÁ THUẬT TOÁN

- **Ưu điểm:** đơn giản, dễ cài đặt, luôn kết thúc sau hữu hạn lần thực hiện thao tác và cho ra kết quả đúng, có tính tổng quát, tính phổ dụng.
- **Khuyết điểm:** độ phức tạp lớn với thời gian chạy có thể tăng theo cấp số mũ. Với việc phải chuyển các câu về dạng mệnh đề, có thể gây mất ý nghĩa của thông tin ban đầu.

#### 4. KIỂM THỬ

Phân tích các input, output của 5 test case.

##### 4.1. Test case 1

Input	Output	Ghi chú
K	6	
6	-B OR -C OR K	(-D OR K) hợp giải (-B OR -C OR D)
A	-F	(-F OR K) hợp giải (-K)
-A OR B	-D	(-D OR K) hợp giải (-K)
-E OR F	B	(-A OR B) hợp giải (A)
-D OR K	-A OR -C OR D	(-A OR B) hợp giải (-B OR -C OR D)
-B OR -C OR D	-E OR K	(-E OR F) hợp giải (-F OR K)
-F OR K	6	
	-C OR D	(-A OR -C OR D) hợp giải (A)
	-C OR K	(-B OR -C OR K) hợp giải (B)
	-B OR -C	(-B OR -C OR K) hợp giải (-K)
	-A OR -C OR K	(-A OR -C OR D) hợp giải (-D OR K)
	-E	(-E OR K) hợp giải (-K)
	-A OR -C	(-A OR -C OR D) hợp giải (-D)
	1	
	-C	(-A OR -C) hợp giải (A)
	0	
	NO	KB không entail $\alpha$ vì không phát sinh mệnh đề mới và không có mệnh đề rỗng

## 4.2. Test case 2

Input	Output	Ghi chú
F OR T	8	
6	-E	(-E OR F) hợp giải (-F)
-C OR E	E OR K	(-C OR E) hợp giải (C OR K)
C OR K	-C	(-C OR T) hợp giải (-T)
B OR -K	-C OR F	(-C OR E) hợp giải (-E OR F)
E OR -K	F OR -K	(-E OR F) hợp giải (E OR -K)
-C OR T	B OR C	(B OR -K) hợp giải (C OR K)
-E OR F	C OR E	(C OR K) hợp giải (E OR -K)
	K OR T	(-C OR T) hợp giải (C OR K)
	13	
	C OR F	(C OR E) hợp giải (-E OR F)
	-K	(-E) hợp giải (E OR -K)
	F OR T	(F OR -K) hợp giải (T OR K)
	B OR T	(B OR C) hợp giải (-C OR T)
	C	(-E) hợp giải (C OR E)
	B OR E	(B OR -K) hợp giải (E OR K)
	E OR T	(C OR E) hợp giải (-C OR T)
	F OR K	(-C OR F) hợp giải (C OR K)
	K	(C OR K) hợp giải (-C)
	B	(B OR C) hợp giải (-C)
	B OR F	(B OR C) hợp giải (-C OR F)
	E	(C OR E) hợp giải (-C)
	E OR F	(-C OR F) hợp giải (C OR E)
	3	
	{ }	(-K) hợp giải K
	T	(F OR T) hợp giải (-F)
	F	E hợp giải (-E OR F)
	YES	KB entail $\alpha$ vì có mệnh đề rỗng



## 4.3. Test case 3

Input	Output	Ghi chú
-A OR F	8	
4	-B OR D	(-B OR D OR F) hợp giải (-F)
-A OR B OR C	-C	(-C OR F) hợp giải (-F)
-B OR D OR F	-A OR C OR D OR F	(-B OR D OR F) hợp giải (-A OR B OR C)
-A OR -D OR F	-D OR F	(A) hợp giải (-A OR -D OR F)
-C OR F	B OR C	(A) hợp giải (-A OR B OR C)
	-A OR -D	(-A OR -D OR F) hợp giải (-F)
	-A OR B OR F	(-C OR F) hợp giải (-A OR B OR C)
	-A OR -B OR F	(-A OR -D OR F) hợp giải (-B OR D OR F)
	12	
	-A OR D OR F	(-A OR B OR F) hợp giải (-B OR D OR F)
	B OR F	(-C OR F) hợp giải (B OR C)
	-B OR F	(-D OR F) hợp giải (-B OR D OR F)
	B	(-C) hợp giải (B OR C)
	-D	(-D OR F) hợp giải (-F)
	-A OR -B	(-F) hợp giải (-A OR -B OR F)
	-A OR B	(-F) hợp giải (-A OR B OR F)
	C OR D OR F	(B OR C) hợp giải (-B OR D OR F)
	-A OR C OR F	(B OR C) hợp giải (-A OR -B OR F)
	-A OR F	(-A OR B OR F) hợp giải (-A OR -B OR F)
	-A OR C OR D	(-A OR C OR D OR F) hợp giải (-F)
	C OR D	(-B OR D) hợp giải (B OR C)
	9	
	-A OR C	(-A OR -B) hợp giải (B OR C)
	-B	(-D) hợp giải (-B OR D)
	F	(B OR F) hợp giải (-B OR F)
	C	(-D) hợp giải (C OR D)
	D	(-C) hợp giải (C OR D)
	D OR F	(C OR D) hợp giải (-C OR F)
	-A	(-A OR -B) hợp giải (-A OR B)
	-A OR D	(-B OR D) hợp giải (-A OR B)
	C OR F	(-B OR F) hợp giải (B OR C)
	1	
	{}	(-D) hợp giải (D)
	YES	KB entail $\alpha$ vì sinh ra mệnh đề rỗng

**4.4. Test case 4**

Input	Output	Ghi chú
A	7	
5	-A OR C	(B OR C) hợp giải (-A OR -B)
B OR C	-B OR D	(A OR D) hợp giải (-A OR -B)
-A OR -B	-C	(-A) hợp giải (A OR -C)
A OR D	B OR -D	(B OR C) hợp giải (-C OR -D)
A OR -C	A OR B	(B OR C) hợp giải (A OR -C)
-C OR -D	D	(-A) hợp giải (A OR D)
	-B OR -C	(A OR -C) hợp giải (-A OR -B)
	3	
	B	(A OR B) hợp giải (-A)
	C OR D	(-B OR D) hợp giải (B OR C)
	-A OR -D	(B OR -D) hợp giải (-A OR -B)
	0	
	NO	KB không entail $\alpha$ vì không phát sinh mệnh đề mới và không có mệnh đề rỗng

**4.5. Test case 5**

Input	Output	Ghi chú
P	7	
6	A	(K) hợp giải (A OR -K)
-A OR -B OR P	B OR -D	(K) hợp giải (B OR -D OR -K)
B OR -D OR -K	-B OR -K OR P	(-A OR -B OR P) hợp giải (A OR -K)
A OR -F	B OR -K	(D) hợp giải (B OR -D OR -K)
A OR -K	-A OR -B	(-P) hợp giải (-A OR -B OR P)
K	-B OR -F OR P	(A OR -F) hợp giải (-A OR -B OR P)
D	-A OR -D OR -K OR P	(B OR -D OR -K) hợp giải (-A OR -B OR P)
	15	
	-D OR -K OR P	(A OR -K) hợp giải (-A OR -D OR -K OR P)
	-B	(A) hợp giải (-A OR -B)
	B	(B OR -D) hợp giải (D)
	-D OR -F OR P	(B OR -D) hợp giải (-B OR -F OR P)
	-A OR -K OR P	(D) hợp giải (-A OR -D OR -K OR P)
	-B OR -K	(A OR -K) hợp giải (-A OR -B)
	-A OR -K	(B OR -K) hợp giải (-A OR -B)
	-A OR -D OR P	(B OR -D) hợp giải (-A OR -B OR P)
	-F OR -K OR P	(B OR -K) hợp giải (-B OR -F OR P)
	-D OR -F OR -K OR P	(-A OR -D OR -K OR P) hợp giải (A OR -F)
	-B OR -F	(-A OR -B) hợp giải (A OR -F)
	-A OR -D	(B OR -D) hợp giải (-A OR -B)
	-B OR P	(K) hợp giải (-B OR -K OR P)
	-K OR P	(-B OR -K OR P) hợp giải (B OR -K)
	-A OR -D OR -K	(-A OR -B) OR (B OR -D OR -K)
	13	
	-D OR P	(K) hợp giải (-D OR -K OR P)
	P	(K) hợp giải (-K OR P)
	-D OR -F OR -K	(A OR -F) hợp giải (-A OR -D OR -K)
	-A OR P	(D) hợp giải (-A OR -D OR P)
	-D OR -K	(-B OR -K) hợp giải (B OR -D OR -K)
	-K	(B OR -K) hợp giải (-B OR -K)
	-D	A OR (-A OR -D)
	-D OR -F	(A OR -F) hợp giải (-A OR -D)
	-F	(B) hợp giải (-B OR -F)
	-F OR -K	(B OR -K) hợp giải (-B OR -F)
	-A	(-A OR -D) hợp giải (D)
	{}	(-B) hợp giải (B)
	-F OR P	(-D OR -F OR P) hợp giải (D)
	YES	KB entail $\alpha$ vì sinh ra mệnh đề rỗng

## **5. TÀI LIỆU THAM KHẢO**

[1] Artificial Intelligence: A modern Approach Fourth Edition, trang 227-229