

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM**  
**KHOA CÔNG NGHỆ THÔNG TIN**

~0~



**BÁO CÁO ĐỒ ÁN 2 MÔN CƠ SỞ TRÍ TUỆ NHÂN TẠO:**  
**LOGIC**

20120313 – Phan Tấn Kiệt

*TP Hồ Chí Minh, ngày 3 tháng 12 năm 2022*

## MỤC LỤC

<b>I. PHẦN BÁO CÁO</b>	<b>3</b>
Tự đánh giá bài làm	3
<b>II. PHẦN NỘI DUNG</b>	<b>4</b>
1. Thuật toán và các hàm	4
2. Cấu trúc lưu trữ đầu vào và mô tả đầu ra	4
a. Cấu trúc lưu trữ đầu vào	4
b. Mô tả đầu ra	5
3. Test cases và đánh giá	5
a. Test cases	5
b. Đánh giá	6

# I. PHẦN BÁO CÁO

## Tự đánh giá bài làm

Câu	Tự đánh giá	Điểm
Đọc dữ liệu đầu vào và lưu trong cấu trúc dữ liệu phù hợp	Hoàn thành	0.5
Cài đặt giải thuật hợp giải trên logic mệnh đề	Hoàn thành	1
Các bước suy diễn phát sinh đủ mệnh đề và kết luận đúng	Hoàn thành	2.5
Tuân thủ mô tả định dạng của đề bài	Hoàn thành	0.5
Báo cáo test case và đánh giá	Không có giải thích về input và output, đánh giá chưa đầy đủ	0.2
Điểm trung bình tự đánh giá:		4.7

## II. PHẦN NỘI DUNG

### 1. Thuật toán và các hàm

Hàm	Mô tả
readInput(fileName)	Đọc dữ liệu từ input.txt
writeOutput(data, result, fileName)	Ghi dữ liệu vào output.txt
PLResolution(alpha, KB)	Thuật toán chính
PLResolve(c1, c2)	Kết hợp hai mệnh đề
isResolvable(c1, c2)	Kiểm tra hai mệnh đề có khả năng kết hợp hay
negativeLiteral(l)	Lấy giá trị đối của literal
negativeClause(c)	Lấy giá trị đối của mệnh đề
connect(c1,c2)	Nối các mệnh đề với nhau
hasTautology(c)	Kiểm tra mệnh đề có chứa hai literal đối của nhau
isInside(c1, c2)	Kiểm tra mệnh đề này có nằm trong mệnh đề còn lại
sort(c)	Sắp xếp mệnh đề

### 2. Cấu trúc lưu trữ đầu vào và mô tả đầu ra

#### a. Cấu trúc lưu trữ đầu vào

Đầu vào được lưu trữ bằng cách dùng kiểu dữ liệu danh sách (list) như sau:

```
-A
4
-A OR B
B OR -C
A OR -B OR C
-B|
```



```
Alpha: ['-A']
n: 4
KB: [['-A', 'B'], ['B', '-C'], ['A', '-B', 'C'], ['-B']]
```

### *b. Mô tả đầu ra*

```
Result: True
Steps: [[['C'], ['-A'], ['B'], ['-C']], [[], ['A', '-B'], ['A', 'C'], ['-B', 'C']]]
```



```
|4
C
-A
B
-C
4
A OR -B
A OR C
-B OR C
{}
YES
```

## 3. Test cases và đánh giá

### *a. Test cases*

#### ○ Case 1:

```
input0.txt x input1.txt input2.txt ... output0.txt x output1.txt output2.txt ...
INPUT > input0.txt
1 -B
2 4
3 -E
4 B OR -C
5 -C OR A
6 C OR E
OUTPUT > output0.txt
1 3
2 C
3 B OR E
4 A OR E
5 1
6 A
7 0
8 NO
```

#### ○ Case 2:

```
input0.txt input1.txt x input2.txt ... output1.txt x output2.txt output3.txt ...
INPUT > input1.txt
1 A OR C
2 4
3 -A OR B
4 B OR -C
5 A OR -B OR C
6 -B
OUTPUT > output1.txt
1 2
2 -B OR C
3 A OR -B
4 2
5 -A OR C
6 A OR -C
7 0
8 NO
```

○ Case 3:

```

INPUT > input2.txt
1  -B
2  5
3  -B OR D
4  -A OR -B
5  B OR C
6  -A OR C OR D
7  A
8

OUTPUT > output2.txt
1  5
2  C OR D
3  D
4  -A OR C
5  -B
6  -A
7  2
8  C
9  {}
10 YES
  
```

○ Case 4:

```

INPUT > input3.txt
1  R
2  3
3  P OR Q
4  -P OR R
5  -Q OR R
6

OUTPUT > output3.txt
1  4
2  Q OR R
3  P OR R
4  -P
5  -Q
6  3
7  P
8  Q
9  R
10 1
11 {}
12 YES
  
```

○ Case 5:

```

INPUT > input4.txt
1  A OR -B
2  4
3  -E OR -G
4  -B OR F
5  -F
6  A OR -F

OUTPUT > output4.txt
1  3
2  -B
3  A OR -B
4  F
5  2
6  A
7  {}
8  YES
  
```

**b. Đánh giá**

Ưu điểm: giải thuật hợp giải mang tính bao hàm  $KB \models \alpha$ . Rõ ràng, xúc tích dễ thực hiện.

Nhược điểm: Các mệnh đề mới được tạo ra từ sự kết hợp các cặp mệnh đề  $(i,j)$  trong cơ sở tri thức. Do vậy nhiều mệnh đề phức tạp và không cần thiết được sinh ra, làm tăng độ phức tạp của thuật toán.

Cách khắc phục: Có thể sử dụng một hàm kiểm tra hai mệnh đề có thể kết hợp của hai mệnh đề, nếu có thì thực hiện, không thì bỏ qua.