

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



MÔN HỌC
CƠ SỞ TRÍ TUỆ NHÂN TẠO

ĐỒ ÁN 02: LOGIC

Thông tin sinh viên:

Họ và tên: Dương Thanh Hiệp

Mssv: 19120505

Email: 19120505@student.hcmus.edu.vn

1. Đánh giá mức độ hoàn thành hoàn thành.

- Tất cả các yêu cầu của đề bài
- Đánh giá mức độ hoàn thành: 100%

2. Ý tưởng thực hiện thuật toán, mô tả các hàm chức năng.

- Thuật toán:

```
+def PL-RESOLUTION(KB,  $\alpha$ ):
    clauses  $\leftarrow$  KB  $\wedge \neg\alpha$ 
    new  $\leftarrow$  []
    while(1):
        resolvents = []
        for each pair of clauses  $C_i, C_j$  in clauses:
            temp  $\leftarrow$  PL-RESOLVE( $C_i, C_j$ )
            if temp not in resolvents and new:
                resolvents  $\leftarrow$  resolvents  $\cup$  temp
        filtTrueClause(resolvents)
        if resolvents contains []:
            return true
        new  $\leftarrow$  new  $\cup$  resolvents
        if new  $\subseteq$  clauses:
            return false
        clauses  $\leftarrow$  clauses  $\cup$  new
```

- + Dữ liệu đầu vào từ file input.txt được tổ chức lại thành:

- KB: mảng 2 chiều n dòng, mỗi dòng là một mệnh đề, trong đó mỗi phần tử của dòng là 1 literal đã được sắp xếp theo thứ tự.
- a: mảng 1 chiều, là mệnh đề cần chứng minh.

- Mô tả các hàm:

- + Hàm readFile(filename): đọc file input và trả về KB,a như cấu trúc ở trên.
- + Hàm findNegative(clause): tìm phủ định của 1 mệnh đề.
- + Hàm checkDuality(L1, L2): kiểm tra sự đối ngẫu của 2 literal L1 và L2: có = 1 | không = 0
- + Hàm checkTrueClause(clause): kiểm tra xem mệnh đề có ở dạng A v - A v B hay không. Có = 1 | không = 0
- + Hàm filtTrueClause(abc): lọc các mệnh đề trên khỏi abc.
- + Hàm checkContain(X, Y): kiểm tra X có chứa Y hay không.

- + Hàm sortClause(clause): sắp xếp các literal trong mệnh đề theo thứ tự tăng dần của bảng chữ cái.
- + Hàm PL_RESOLVE(C1,C2): hợp giải 2 mệnh đề, hợp giải được (có cặp đối ngẫu) trả về mệnh đề mới được phát sinh, ngược lại trả về 0.
- + Hàm PL_RESOLUTION(KB,a): hàm chính thực hiện thuật toán. Trả về kết quả bài toán, danh sách các mệnh đề phát sinh mới.
- + Hàm writeFile(filename, result, size, new): ghi kết quả ra file output
- + Hàm main(): thực thi bài làm.

3. Các kịch bản kiểm thử.

- Kịch bản 1: ví dụ trong đề, a là mệnh đề đơn.

```

cau4.py X input.txt
input.txt
1 -A
2 4
3 -A OR B
4 B OR -C
5 A OR -B OR C
6 -B

```

```

cau4.py output.txt
output.txt
1 3
2 -A
3 B
4 -C
5 4
6 -B OR C
7 A OR C
8 A OR -B
9 {}
10 YES

```

- Kịch bản 2:

Input	Output	Ghi chú
-A OR B	7	
6	A OR -B OR -D	-C OR -D hợp giải với A OR -B OR C
-C OR -D	A OR -C	-C OR -D hợp giải với A OR D
A OR -B OR C	A OR B OR -C	-C OR -D hợp giải với A OR B OR D
-A OR -B	-B OR C	A OR -B OR C hợp giải với -A OR -B
-C	A OR -B	A OR -B OR C hợp giải với -C
A OR D	A OR C OR D	A OR -B OR C hợp giải với A OR B OR D
A OR B OR D	-B OR D	-A OR -B hợp giải với A OR D
	5	
	-B OR -D	-C OR -D hợp giải với -B OR C
	-B OR -C	-C OR -D hợp giải với -B OR D
	-B OR C OR D	-A OR -B hợp giải với A OR C OR D
	A OR -C OR -D	A OR -B OR -D hợp giải với A OR B OR -C
	A OR -C OR D	A OR B OR -C hợp giải với -B OR D
	4	
	A OR -B OR D	A OR -B OR C hợp giải với A OR -C OR D
	-B OR -C OR -D	-A OR -B hợp giải với A OR -C OR -D
	-B OR -C OR D	-A OR -B hợp giải với A OR -C OR D
	A OR -B OR -C	-B OR D hợp giải với A OR -C OR -D
	0	
	NO	

- Kịch bản 3:

Input	Output	Ghi chú
A OR B	5	
5	-A OR C	-A OR B hợp giải với -B OR C
-A OR B	B OR C OR D	-A OR B hợp giải với A OR C OR D
B OR -C	A OR B OR D	-C OR -D hợp giải với A OR B OR D
A OR -B OR C	-C	B OR -C hợp giải với -B
-B OR C	C OR D	A OR C OR D hợp giải với -A
A OR C OR D	4	
	B OR D	-A OR B hợp giải với A OR B OR D
	A OR -B	A OR -B OR C hợp giải với -C
	A OR D	A OR C OR D hợp giải với -C
	D	-C hợp giải với C OR D
	1	
	A OR -C	B OR -C hợp giải với A OR -B
	0	
	NO	

- Kịch bản 4:a phức tạp, chứa nhiều literal

```
cau4.py  input4.txt
input >  input4.txt
1      -A OR -B OR C
2      4
3      -A OR B OR D OR E
4      -C OR -D
5      A OR -B OR C
6      -B OR -E
```

```
cau4.py  output4.txt
output > output4.txt
1      6
2      -A OR B OR -C OR E
3      B OR D OR E
4      A OR -B OR -D
5      A OR C
6      A OR -B
7      -E
8      8
9      B OR C OR D OR E
10     -A OR B OR D
11     B OR -C OR E
12     A OR -D
13     A OR C OR D OR E
14     -A OR B OR -C
15     A OR D OR E
16     B OR D
17     22
18     A OR -C OR E
19     B OR -C
20     A OR C OR D
21     -A OR D OR -E
22     A OR -B OR C OR D
23     -A OR -C OR -E
24     A OR -B OR D
25     D OR -E
26     B OR -C OR -D OR E
```

```
cau4.py  output4.txt
output > output4.txt
90     -A OR -C OR -D OR -E
91     A OR -C OR -E
92     A OR -B OR C OR -E
93     A OR -B OR -C OR -E
94     A OR -C OR -D OR -E
95     B OR C OR D OR -E
96     A OR B OR -E
97     B OR -C OR -D OR -E
98     A OR B OR -C OR D
99     A OR B OR D OR -E
100    -A OR -B OR -C OR -E
101    B OR -C OR D OR -E
102    -A OR -B OR D OR -E
103    A OR -C OR D OR -E
104    A OR B OR -C OR -D
105    A OR B OR C OR -E
106    A OR B OR -D OR -E
107    A OR B OR -C OR -E
108    2
109    A OR C OR -D OR -E
110    A OR B OR C OR -D
111    0
112    NO
```

- Kịch bản 5: phức tạp như kịch bản 4 nhưng chứa nhiều cặp đối ngẫu nên chạy rất nhanh

Input	Output	Ghi chú
-A OR -B OR C	8	
4	-A OR C OR D OR -E	-A OR B OR D OR -E hợp giải với -B OR C
-A OR B OR D OR -E	-A OR D OR -E	-A OR B OR D OR -E hợp giải với -B OR -E
-B OR C	B OR D OR -E	-A OR B OR D OR -E hợp giải với A
A OR -B OR C	C	-B OR C hợp giải với B
-B OR -E	-B	-B OR C hợp giải với -C
	A OR C	A OR -B OR C Hợp giải với -C
	A OR -B	A OR -B OR C hợp giải với -C
	-E	A OR -B OR C hợp giải với -C
	8	
	B OR C OR D OR -E	-A OR B OR D OR -E hợp giải với A OR C
	C OR D OR -E	-B OR C hợp giải với B OR D OR -E
	-B OR C OR D OR -E	B OR -C hợp giải với A OR -B
	A OR C OR D OR -E	A OR -B OR C Hợp giả với -A OR D OR -E

	D OR -E	-B OR -E hợp giải với B OR D OR -E
	{ }	B hợp giả với - B
	-B OR D OR -E	-A OR D OR - E Hợp giải với A OR -B
	A OR D OR -E	-C hợp giải với -A OR C OR D OR -E
	YES	Vì chứa mệnh đề rỗng

4. Đánh giá thuật toán.

- Điểm mạnh: luôn cho ra kết quả đúng (yes hoặc no). Ý tưởng rõ ràng dễ thực hiện.
- Điểm yếu: việc phát sinh mệnh đề mới không có chọn lọc làm tốn nhiều chi phí thực hiện. Duyệt tất cả nên ở vòng lặp sau sẽ bị duyệt lại những cặp mệnh đề đã duyệt rồi.
- Cách khắc phục: Thêm một cấu trúc để lưu các mệnh đề mới được phát sinh thay vì gộp vào clause như thuật toán trên. Ở vòng lặp tiếp theo chỉ bắt cặp chéo giữa tập các mệnh đề cũ và tập các mệnh đề mới phát sinh.