

# Cơ sở trí tuệ nhân tạo

## Project 2: Logic

Võ Thành Nam  
MSSV: 19120301

December 2021

## 1 Thông tin sinh viên

- Họ và tên: Võ Thành Nam
- MSSV: 19120301

## 2 Báo cáo bài làm

### 2.1 Ý tưởng thực hiện

Ta có dữ liệu đầu vào luôn đúng, do đó ta thực hiện hợp giải trên toàn bộ knowledge base hợp với phủ định của mệnh đề  $\alpha$ . Thuật toán kết thúc khi không thể tạo ra mệnh đề mới hoặc tạo ra mệnh đề rỗng.

Các bước thực hiện:

1. Khởi tạo mảng clause là hợp của knowledge base và phủ định của mệnh đề  $\alpha$
2. (a)  $i=0$   
(b) Với mỗi mệnh đề thứ  $i$ , ta hợp giải với mệnh đề thứ  $j > i$ , nếu kết quả trả về là mệnh đề luôn đúng hoặc mệnh đề vô ích thì tiếp tục thực hiện với mệnh đề  $j$  tiếp theo.

- (c) Nếu kết quả chưa có trong clause hoặc resolvents (các mệnh đề phát sinh của vòng lặp hiện tại) không chứa rỗng thì thêm vào resolvents.
  - (d) Thêm resolvents vào danh sách các mệnh đề đã phát sinh new.
  - (e)  $i += 1$
3. Thêm resolvents vào danh sách các mệnh đề đã phát sinh sau mỗi vòng lặp.
  4. Nếu trong resolvents có mệnh đề rỗng thì trả về true và kết thúc.
  5. Nếu new là con của clause thì trả về false và kết thúc.
  6. Thêm new vào clause và quay lại bước 2.

## 2.2 Pseudocode

---

**Algorithm 1** PL\_Resolution

---

**Require:**  $KB$ : knowledge base

$\alpha$ : the query  
 $clause \leftarrow KB \wedge \neg\alpha$   
 $new \leftarrow \{\}$   
 $operations \leftarrow \{\}$   
**while** True **do**  
    **for**  $C_i, C_j \in clause$  and  $C_i \neq C_j$  **do**  
         $resolve \leftarrow PL\_ReSolve(C_i, C_j)$   
        **if**  $resolve \neq True$  and  $resolve \notin clause$  and  $resolvents$  not contain  
        empty clause **then**  
             $resolvents.append(resolve)$   
        **end if**  
         $new = new \cup resolvents$   
    **end for**  
     $operations.append(resolvents)$   
    **if**  $resolvents$  contain empty clause **then**  
        **return**  $operations, True$   
    **end if**  
    **if**  $new \subseteq clause$  **then**  
        **return**  $operations, False$   
    **end if**  
     $clause = clause \cup new$   
**end while**

---

## 2.3 Các kịch bản kiểm thử

1. Kịch bản 1 (input1.txt)

Input	Output	Hợp giải
A OR C	3	
6	-B OR C OR D	(A OR -B OR C) và (-A OR D)
-A OR B	A OR -B	(A OR -B OR C) và (-C)
B OR -C	-B OR C	(A OR -B OR C) và (-A)
A OR -B OR C	4	
-B	-A OR C OR D	(-A OR B) và (-B OR C OR D)
-A OR D	-A OR C	(-A OR B) và (-B OR C)
-C	A OR -C	(B OR -C) và (A OR -B)
	-B OR D	(-A OR D) và (A OR -B)
	3	
	-A OR B OR D	(B OR -C) và (-A OR C OR D)
	-C OR D	(B OR -C) và (-B OR D)
	A OR -B OR D	(-B OR C OR D) và (A OR -C)
	2	
	A OR -C OR D	(A OR -B OR D) và (B OR -C)
	B OR -C OR D	(-A OR B OR D) và (A OR -C)
	0	
	NO	

2. Kịch bản 2 (input2.txt)

Input	Output	Hợp giải
C OR -E	5	
4	-B OR C OR D	(-A OR -B OR C) và (A OR D)
-A OR -B OR C	-A OR C OR -E	(-A OR -B OR C) và (B OR -E)
A OR D	-A OR -B	(-A OR -B OR C) và (-C)
B OR -E	A	(A OR D) và (-D)
-D	B	(B OR -E) và (E)
	9	
	-B OR C	(-A OR -B OR C) và (A)
	-A OR C	(-A OR -B OR C) và (B)
	C OR D OR -E	(A OR D) và (-A OR C OR -E)
	-B OR D	(A OR D) và (-A OR -B)
	-A OR -E	(B OR -E) và (-A OR -B)
	C OR D	(-B OR C OR D) và (B)
	C OR -E	(-A OR C OR -E) và (A)
	-B	(-A OR -B) và (A)
	-A	(-A OR -B) và (B)
	5	
	D OR -E	(A OR D) và (-A OR -E)
	D	(A OR D) và (-A)
	-E	(B OR -E) và (-B)
	C	(-D) và (C OR D)
	{}	(A) và (-A)
	YES	

3. Kịch bản 3 (input3.txt)

Input	Output	Hợp giải
-A OR B OR C	4	
5	-B OR C	(-A OR -B) và (A OR C)
-A OR -B	-A OR -C	(-A OR -B) và (B OR -C)
-C	A OR -B	(-C) và (A OR -B OR C)
A OR C	A OR B	(A OR C) và (B OR -C)
A OR -B OR C	2	
B OR -C	A OR -C	(B OR -C) và (A OR -B)
	-B OR -C	(-A OR -C) và (A OR -B)
	0	
	NO	

4. Kịch bản 4 (input4.txt)

Input	Output	Hợp giải
-A OR B OR C OR -D OR E	5	
5	B OR -C	(-A OR B OR -C) và (A)
-A OR B OR -C	-A OR -C	(-A OR B OR -C) và (-B)
-C OR E	B OR -C OR D	(-C OR E) và (B OR D OR -E)
B OR D OR -E	A OR E	(-C OR E) và (A OR C)
A OR C	D OR -E	(B OR D OR -E) và (-B)
D	5	
	B OR -C OR E	(-A OR B OR -C) và (A OR E)
	-C OR D	(-C OR E) và (D OR -E)
	A OR B OR D	(B OR D OR -E) và (A OR E)
	A OR B	(A OR C) và (B OR -C)
	A OR D	(A OR E) và (D OR -E)
	1	
	A OR B OR E	(A OR C) và (B OR -C OR E)
	0	
	NO	

5. Kịch bản 5 (input5.txt)

Input	Output	Hợp giải
C		
4		
-A OR -B OR C	5	
-A OR B	-A OR C	(-A OR -B OR C) và (-A OR B)
A	-B OR C	(-A OR -B OR C) và (A)
B OR D	-A OR C OR D	(-A OR -B OR C) và (B OR D)
	-A OR -B	(-A OR -B OR C) và (-C)
	B	(-A OR B) và (A)
	5	
	-A	(-A OR B) và (-A OR -B)
	C	(A) và (-A OR C)
	C OR D	(A) và (-A OR C OR D)
	-B	(A) và (-A OR -B)
	-A OR D	(B OR D) và (-A OR -B)
	2	
		(A) và (-A)
	D	(B OR D) và (-B)
	YES	

## 2.4 Đánh giá về giải thuật

1. Điểm mạnh: giải thuật luôn đảm bảo tìm được kết quả hợp giải logic mệnh đề.
2. Điểm yếu: giải thuật sử dụng vòng lặp để hợp quá nhiều, tốc độ xử lý không nhanh do phải duyệt lại mảng clause.

Đề xuất cách cải tiến thuật giải: ở bước hợp giải hai mệnh đề trong clause, thay vì phải duyệt toàn bộ clause, ta có thể thay đổi j để chỉ hợp giải với những mệnh đề vừa phát sinh ở vòng lặp trước đó, giúp giảm được lượng lớn thao tác.

## 3 Tài liệu tham khảo

1. Slide bài giảng
2. [https://en.wikipedia.org/wiki/List\\_of\\_rules\\_of\\_inference](https://en.wikipedia.org/wiki/List_of_rules_of_inference)

3. <https://cs.calvin.edu/courses/cs/344/kvlinden/resources/AIMA-3rd-edition.pdf>