

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



LAB 2
LOGIC

Giảng viên hướng dẫn: Lê Hoài Bắc

Môn học: Cơ sở trí tuệ nhân tạo

Lớp: 20_21

Họ tên: Nguyễn Minh Kha

MSSV: 20120502

NGUYỄN MINH KHA

Thành phố Hồ Chí Minh, ngày 4 tháng 12 năm 2022

MỤC LỤC

A. TỰ ĐÁNH GIÁ	3
B. MÔ TẢ CÀI ĐẶT	4
1. Dữ liệu	4
a. Câu alpha và các clauses	4
b. Knowledge Base	4
c. Kết quả bài toán	4
2. Cài đặt	5
a. Hàm negative_clause	5
b. Hàm remove_duplicate_element_in_list	5
c. Hàm PL_RESOLVE	5
d. Hàm PL_RESOLUTION	5
C. ĐÁNH GIÁ GIẢI THUẬT HỢP GIẢI TRÊN LOGIC MỆNH ĐỀ	7
1. Ưu điểm	7
2. Khuyết điểm	7
3. Cải tiến	7
D. CÁC TEST CASE	8
1. Testcase 1	8
2. Testcase 2	8
3. Testcase 3	9
4. Testcase 4	9
5. Testcase 5	10
E. TÀI LIỆU THAM KHẢO	11

A. TỰ ĐÁNH GIÁ

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

B. MÔ TẢ CÀI ĐẶT

1. Dữ liệu

a. Câu alpha và các clauses

- Các clauses và câu alpha được đọc theo từng dòng dưới dạng string và được tách (split) bởi chuỗi “_OR_” (_ là khoảng trắng).
- Lưu các clauses và câu alpha dưới dạng list string chứa các literals (literal kiểu là string).
- Các literals trong các clauses và câu alpha được sắp xếp theo thứ tự chữ cái A đến Z.

Ví dụ: $A \text{ OR } \neg B \text{ OR } C \Leftrightarrow [“A”, “\neg B”, “C”]$

b. Knowledge Base

- Được lưu thành một list chứa các clauses

Ví dụ: Knowledge Base bao gồm các clauses

$$\begin{cases} A \text{ OR } \neg B \text{ OR } C \\ A \text{ OR } \neg B \\ \neg C \end{cases} \Leftrightarrow [[“A”, “\neg B”, “C”], [“A”, “\neg B”], [“\neg C”]]$$

c. Kết quả bài toán

- Được lưu dạng List chứa các resolvents (trong PL-RESOLUTION) khi hợp giải trong mỗi lần lặp.

Ví dụ:

Output: 2 A OR -B C 2 -A OR -C B 2 -C {}	Output = [[[“A”, “-B”], [“C”]], [[“-A”, “-C”], [“B”]], [[“-C”], []]]
---	--

2. Cài đặt

a. Hàm `negative_clause`

- Input: câu alpha hoặc một clause
- Output: một knowledge base với các clause là negative form của literals của input

Ví dụ: clause ["A", "-B", "C"] \Leftrightarrow [{"-A"}, [{"B"}, [{"-C"}]]

b. Hàm `remove_duplicate_element_in_list`

- Input: Một list
- Output: List input đã xóa các phần tử trùng

c. Hàm `PL_RESOLVE`

- Input: Hai clauses (clauseA, clauseB)
- Output: Hợp giải của hai clauses hoặc clause True
- Mô tả:
 - Đếm số lượng cặp đối ngẫu giữa hai clauses
 - Trường hợp số lượng cặp đối ngẫu khác 1 (không có cặp đối ngẫu hoặc có từ 2 cặp đối ngẫu) trả về [True] (true clause)
 - Trường hợp số lượng cặp đối ngẫu bằng 1 tiến hành hợp giải hai clauses thành `combined_clause`
 - Xóa cặp đối ngẫu ở hai clauses
 - Hợp hai clauses (`combined_clause = clauseA + clauseB`)
 - Sắp xếp `combined_clause` theo thứ tự chữ cái từ A đến Z
 - Lọc các literals giống nhau trong `combined_clause` (dùng hàm `remove_duplicate_element_in_list(combined_clause)`)
 - Trả về `combined_clause`

d. Hàm `PL_RESOLUTION`

- Input: Knowledge Base và câu alpha
- Output:
 - Biến kết quả (True or False)
 - Output (phần 1.c)
- Mô tả:

- Pseudo code

```

function PL-RESOLUTION( $KB, \alpha$ ) returns true or false
  inputs:  $KB$ , the knowledge base, a sentence in propositional logic
            $\alpha$ , the query, a sentence in propositional logic

   $clauses \leftarrow$  the set of clauses in the CNF representation of  $KB \wedge \neg\alpha$ 
   $new \leftarrow \{ \}$ 
  loop do
    for each pair of clauses  $C_i, C_j$  in  $clauses$  do
       $resolvents \leftarrow$  PL-RESOLVE( $C_i, C_j$ )
      if  $resolvents$  contains the empty clause then return true
       $new \leftarrow new \cup resolvents$ 
    if  $new \subseteq clauses$  then return false
     $clauses \leftarrow clauses \cup new$ 

```

- all_clauses = Knowledge Base + negative_clause(câu alpha)
- Lặp (while True)
 - resolvents append PL_RESOLVE của mỗi cặp clauses[i], clauses[j] trong all_clauses
 - Loại bỏ các [True] trong resolvents
 - Loại bỏ các clauses lặp trong resolvents
 - Output append resolvents
 - Nếu resolvents chứa clause rỗng [] thì trả về kết quả bài toán true và output
 - Nếu resolvents rỗng (không sinh ra các clauses mới) thì trả về kết quả bài toán false và output
 - Hợp all_clauses và resolvents

C. ĐÁNH GIÁ GIẢI THUẬT HỢP GIẢI TRÊN LOGIC MỆNH ĐỀ

1. Ưu điểm

2. Nhược điểm

- Ở bước duyệt từng cặp clauses, việc duyệt lại tất cả clause là không cần thiết mà chỉ cần duyệt các clauses mới được hợp giải thành ở lần lặp trước rồi hợp giải với các clauses đã có.

3. Cải tiến

- Sau mỗi lần lặp, chỉ xét các clauses mới được hợp giải thành với các clauses đã có (Không xét các clause không được sinh ra ở lần lặp trước).

D. CÁC TEST CASE

1. Testcase 1

INPUT	OUTPUT	Giải thích
A OR C OR -D OR -F	13	
7	A OR -D	(A OR C OR -D) hợp giải (-C) ở -alpha
A OR C OR -D	C OR -D OR E	(A OR C OR -D) hợp giải (-A OR E)
A OR C	C OR -D	(A OR C OR -D) hợp giải (-A) ở -alpha
-C	A	(A OR C) hợp giải (-C)
-A OR E	C OR E	(A OR C) hợp giải (-A OR E)
C OR -E OR -F	C	(A OR C) hợp giải (-A) ở -alpha
-F	-E OR -F	(C OR -E OR -F) hợp giải (-C) ở -alpha
-D OR E OR -F	-A OR C OR -F	(C OR -E OR -F) hợp giải (-A OR E)
	C OR -D OR -F	(C OR -E OR -F) hợp giải (-D OR E OR -F)
	C OR -E	(C OR -E OR -F) hợp giải (F) ở -alpha
	{}	(-F) hợp giải (F) ở -alpha
	E OR -F	(-D OR E OR -F) hợp giải (D) ở -alpha
	-D OR E	(-D OR E OR -F) hợp giải (F) ở -alpha
	YES	KB entails alpha do chứa clause {}

2. Testcase 2

INPUT	OUTPUT	Giải thích
-B OR E	10	
5	B OR E	(-C OR E) hợp giải (B OR C OR E)
-C OR E	A OR B OR E	(-C OR E) hợp giải (A OR B OR C)
E OR -B	A OR -C	(-C OR E) hợp giải (A OR -E)
B OR C OR E	-C	(-C OR E) hợp giải (-E) ở -alpha
A OR B OR C	C OR E	(E OR -B) hợp giải (B OR C OR E)
A OR -E	A OR C OR E	(E OR -B) hợp giải (A OR B OR C)
	A OR -B	(E OR -B) hợp giải (A OR -E)
	E	(E OR -B) hợp giải (B) ở -alpha
	-B	(E OR -B) hợp giải (-E) ở -alpha
	B OR C	(B OR C OR E) hợp giải (-E) ở -alpha
	6	
	A OR B	(B OR E) ở lần 1 hợp giải (A OR -E)
	A OR E	(B OR E) ở lần 1 hợp giải (A OR -B) ở lần 1
	A OR C	(C OR E) ở lần 1 hợp giải (A OR -E)
	C	(C OR E) ở lần 1 hợp giải (-E) ở -alpha
	A	(A OR -B) ở lần 1 hợp giải (B) ở -alpha
	{}	(E) ở lần 1 hợp giải (-E) ở -alpha
	YES	KB entails alpha do chứa clause {}

3. Testcase 3

INPUT	OUTPUT	Giải thích
A OR -F	5	
4	A OR -C	(A OR -B) hợp giải (B OR -C)
A OR -B	-B	(A OR -B) hợp giải (-A) ở -alpha
B OR -C	B OR -E	(B OR -C) hợp giải (C OR -E)
C OR -E	C OR -F	(C OR -E) hợp giải (E OR -F)
E OR -F	E	(E OR -F) hợp giải (F) ở -alpha
	7	
	A OR -E	(A OR -C) ở lần 1 hợp giải (C OR -E)
	-C	(A OR -C) ở lần 1 hợp giải (-A) ở -alpha
	-E	(B OR -E) ở lần 1 hợp giải (-B) ở lần 1
	A OR -F	(C OR -F) ở lần 1 hợp giải (A OR -C) ở lần 1
	B OR -F	(C OR -F) ở lần 1 hợp giải (B OR -C)
	B	(E) ở lần 1 hợp giải (B OR -E) ở lần 1
	C	(E) ở lần 1 hợp giải (C OR -E)
	3	
	A	(A OR -E) ở lần 2 hợp giải (E) ở lần 1
	-F	(A OR -F) ở lần 2 hợp giải (-A) ở -alpha
	{}	(B) ở lần 2 hợp giải (-B) ở lần 1
	YES	KB entails alpha do chứa clause {}

4. Testcase 4

INPUT	OUTPUT	Giải thích
B	6	
5	A OR -B OR -D OR E	(-B OR C OR -D OR E) hợp giải (A OR -C OR E)
-B OR C OR -D OR E	C OR -D OR E	(-B OR C OR -D OR E) hợp giải (B OR E)
A OR -C OR E	A OR B OR D OR E	(A OR -C OR E) hợp giải (A OR B OR C OR D OR E)
B OR E	E	(B OR E) hợp giải (-B) ở -alpha
A OR B OR C OR D OR E	A OR B OR C OR E	(A OR B OR C OR D OR E) hợp giải (-D OR E)
-D OR E	A OR C OR D OR E	(A OR B OR C OR D OR E) hợp giải (-B) ở -alpha
	6	
	A OR -D OR E	(A OR -B OR -D OR E) ở lần 1 hợp giải (B OR E)
	A OR C OR -D OR E	(A OR -B OR -D OR E) ở lần 1 hợp giải (A OR B OR C OR E) ở lần 1
	A OR -B OR C OR E	(A OR -B OR -D OR E) ở lần 1 hợp giải (A OR -B OR C OR E) ở lần 1
	A OR C OR E	(C OR -D OR E) ở lần 1 hợp giải (A OR C OR D OR E) ở lần 1
	A OR B OR E	(A OR B OR D OR E) ở lần 1 hợp giải (-D OR E)
	A OR D OR E	(A OR B OR D OR E) ở lần 1 hợp giải (-B) ở -alpha
	2	
	A OR E	(A OR -D OR E) ở lần 2 hợp giải (A OR D OR E) ở lần 2
	A OR -B OR E	(A OR -B OR C OR E) ở lần 2 hợp giải (A OR -C OR E)
	0	
	NO	Không phát sinh thêm clause mới

5. Testcase 5

INPUT	OUTPUT	Giải thích
B OR H	5	
4	C OR E OR F	(B OR C OR E OR F) hợp giải (-B) ở -alpha
B OR C OR E OR F	-B OR H	(E OR H) hợp giải (-B OR -E OR H)
E OR H	E	(E OR H) hợp giải (-H) ở -alpha
-B OR -E OR H	-B OR -E	(-B OR -E OR H) hợp giải (-H) ở -alpha
C OR F OR H	C OR F	(C OR F OR H) hợp giải (-H) ở -alpha
	3	
	-B OR C OR F OR H	(C OR E OR F) ở lần 1 hợp giải (-B OR -E OR H)
	-B OR C OR F	(C OR E OR F) ở lần 1 hợp giải (-B OR -E) ở lần 1
	C OR E OR F OR H	(-B OR H) ở lần 1 hợp giải (B OR C OR E OR F)
	0	
	NO	Không phát sinh thêm clause mới

E. TÀI LIỆU THAM KHẢO

[1] Artificial Intelligence: A Modern Approach, 4th edition