

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

Báo cáo

Đề án: Lab2 Logic

Môn học: Cơ sở trí tuệ nhân tạo

Sinh viên thực hiện:

Long Văn Thắng (22120331)

Giáo viên hướng dẫn:

Giảng viên: Nguyễn Ngọc Đức

Giảng viên: Nguyễn Thị Thu Hằng

Giảng viên: Nguyễn Trần Duy Minh

Ngày 24 tháng 11 năm 2024



Mục lục

1 Sinh Viên	2
2 Phân Tích Chi Tiết Các Hàm	2
2.1 Class Clause	2
2.1.1 Clause.parse(clausestr: str)	2
2.1.2 Clause.negate()	3
2.1.3 Clause.negate_literal(literal: str)	3
2.1.4 Clause.resolve(other: Clause)	3
2.1.5 Clause.contains_tautology()	3
2.2 Class KnowledgeBase	4
2.2.1 KnowledgeBase.add_clause(clause: Clause)	4
2.2.2 KnowledgeBase.print_kb()	4
2.2.3 KnowledgeBase.print_resolutions(resolutions)	4
2.3 Other functions	5
2.3.1 print_kb()	5
2.3.2 print_resolutions(self, resolutions)	5
2.3.3 format_output(all_steps: List[List[Clause]], entails: bool)	6
2.3.4 parse_input(filepath: str)	6
2.3.5 main()	6
3 Thuật toán PLResolution	7
3.1 Chữ ký Hàm	7
3.2 Giải thích Chi tiết	7
3.2.1 Phủ định của Truy vấn	7
3.2.2 Khởi tạo Cấu trúc Dữ liệu	7
3.2.3 Vòng lặp Chính của Suy Diễn	7
3.2.4 Kiểm tra Kết luận	9
3.2.5 Thêm Mệnh Đề Mới và Lặp Lại	10
4 Giải Thích Chi Tiết Các Trường Hợp Kiểm Tra	10
4.1 input01.txt	10

4.2	input02.txt	12
4.3	input03.txt	13
4.4	input04.txt	16
4.5	input05.txt	18
4.6	input06.txt	20
4.7	input07.txt	22
4.8	input08.txt	24
5	Đánh Giá Phương Pháp Giải Quyết cho Logic Phân Tích	27
5.1	Ưu Điểm	27
5.2	Nhược Điểm	27
6	Giải Pháp Đề Xuất Cho Vấn Đề Cụ Thể	27

1 Sinh Viên

- **Full name:** Long Văn Thắng
- **Student ID:** 22120331

2 Phân Tích Chi Tiết Các Hàm

2.1 Class Clause

2.1.1 Clause.parse(clausestr: str)

Mục đích: Hàm này phân tích chuỗi mệnh đề thành đối tượng `Clause`, nơi mỗi mệnh đề là một phép hợp (disjunction) của các mệnh đề con.

Ví dụ:

- Với chuỗi đầu vào "A OR -B OR C", kết quả trả về sẽ là `Clause({A, -B, C})`.
- Với chuỗi đầu vào "-X OR Y", kết quả trả về sẽ là `Clause({-X, Y})`.

2.1.2 `Clause.negate()`

Mục đích: Trả về một danh sách các đối tượng `Clause`, mỗi đối tượng trong đó đại diện cho sự phủ định của từng mệnh đề con trong mệnh đề ban đầu.

Ví dụ:

- Với mệnh đề `Clause({A, -B})`, khi gọi `negate()`, kết quả sẽ là danh sách chứa các mệnh đề `Clause({-A})` và `Clause({B})`.
- Với mệnh đề `Clause({X, Y})`, khi gọi `negate()`, kết quả sẽ là danh sách `Clause({-X})`, `Clause({-Y})`.

2.1.3 `Clause.negate_literal(literal: str)`

Mục đích: Trả về sự phủ định của một mệnh đề con duy nhất.

Ví dụ:

- `Clause.negate_literal(A)` trả về `-A`.
- `Clause.negate_literal(-B)` trả về `B`.
- `Clause.negate_literal(X)` trả về `-X`.

2.1.4 `Clause.resolve(other: Clause)`

Mục đích: Giải quyết mệnh đề hiện tại với một mệnh đề khác để tìm ra các mệnh đề mới.

Ví dụ:

- Nếu ta có `Clause({A, B})` và `Clause({-A, C})`, thì gọi `resolve()` sẽ trả về `Clause({B, C})`.
- Nếu ta có `Clause({P, Q})` và `Clause({-P, R})`, kết quả của `resolve()` sẽ là `Clause({Q, R})`.

2.1.5 `Clause.contains_tautology()`

Mục đích: Kiểm tra xem mệnh đề có chứa một tautology (mâu thuẫn nội tại) hay không, tức là mệnh đề đó có chứa một mệnh đề con và sự phủ định của nó.

Ví dụ:

- Mệnh đề `Clause({A, -A, B})` chứa một tautology vì có `A` và `-A`, nên `contains_tautology()` trả về `True`.
- Mệnh đề `Clause({X, Y})` không chứa tautology vì không có `X` và `-X`, nên `contains_tautology()` trả về `False`.

2.2 Class KnowledgeBase

2.2.1 KnowledgeBase.add_clause(clause: Clause)

Mục đích: Thêm một mệnh đề vào cơ sở tri thức (Knowledge Base).

Ví dụ:

- Khi thêm mệnh đề `Clause({A, -B})` vào cơ sở tri thức, kết quả là cơ sở tri thức chứa mệnh đề đó: `KnowledgeBase({Clause({A, -B})})`.
- Khi thêm mệnh đề `Clause({-X, Y})` vào cơ sở tri thức, kết quả sẽ là: `KnowledgeBase({Clause({-X, Y})})`.

2.2.2 KnowledgeBase.print_kb()

Mục đích: In ra trạng thái hiện tại của cơ sở tri thức (Knowledge Base) trên màn hình.

Ví dụ:

- Nếu cơ sở tri thức chứa các mệnh đề `Clause({A, -B})` và `Clause({-X, Y})`, khi gọi `print_kb()` sẽ in ra:

– `-A OR B`

– `-X OR Y`

2.2.3 KnowledgeBase.print_resolutions(resolutions)

Mục đích: In ra các phép giải quyết (resolutions) được thực hiện trong mỗi vòng lặp.

Ví dụ:

- Nếu trong vòng lặp đầu tiên, mệnh đề `Clause({-A, B})` và `Clause({A, -B})` được giải quyết, và kết quả là mệnh đề trống, khi gọi `print_resolutions(resolutions)`, sẽ in ra:

- Loop 1:
- Resolving: $\neg A \text{ OR } B$ with $A \text{ OR } \neg B$
- Result: $\{\}$

2.3 Other functions

2.3.1 `print_kb()`

Mục đích: Hiển thị trạng thái hiện tại của cơ sở tri thức, cho thấy tất cả các mệnh đề trong cơ sở tri thức.

Ví dụ:

- Nếu cơ sở tri thức chứa các mệnh đề $\text{Clause}(\{\neg A, B\})$ và $\text{Clause}(\{A, \neg B\})$, khi gọi `print_kb()` sẽ in ra:

- $\neg A \text{ OR } B$
- $A \text{ OR } \neg B$
- ---

2.3.2 `print_resolutions(self, resolutions)`

Mục đích: In ra kết quả của quá trình giải quyết trong mỗi vòng lặp, bao gồm các mệnh đề được giải quyết và kết quả của chúng. Cung cấp giải thích nếu mệnh đề trống được tạo ra.

Ví dụ:

- Nếu giải quyết $\neg A \text{ OR } B$ với $A \text{ OR } \neg B$ trong vòng lặp đầu tiên tạo ra mệnh đề trống, kết quả in ra sẽ là:

- Loop 1:
- Resolving: $\neg A \text{ OR } B$ with $A \text{ OR } \neg B$
- Result: $\{\}$
- ---

2.3.3 `format_output(all_steps: List[List[Clause]], entails: bool)`

Mục đích: Định dạng kết quả đầu ra theo định dạng yêu cầu, hiển thị tất cả các mệnh đề đã được giải quyết và cho biết liệu câu hỏi có thể suy luận từ cơ sở tri thức hay không.

Ví dụ:

- Với danh sách các bước `List({Clause({A, -B}), Clause({A, C})})` và kết quả suy luận là `True`, khi gọi `format_output()`, kết quả sẽ là:

- Step 1: A OR -B
- Step 2: A OR C
- Entails: YES

2.3.4 `parse_input(filepath: str)`

Mục đích: Phân tích tệp đầu vào để trích xuất câu hỏi và các mệnh đề trong cơ sở tri thức.

Ví dụ:

- Khi gọi `parse_input("input.txt")`, nếu tệp chứa câu hỏi `Clause({A})` và các mệnh đề trong cơ sở tri thức là `Clause({B})` và `Clause({-B})`, kết quả sẽ là:

- Câu hỏi: `Clause({A})`
- Cơ sở tri thức: `[Clause({B}), Clause({-B})]`

2.3.5 `main()`

Mục đích: Hàm chính để chạy chương trình PL-Resolution dựa trên các tham số dòng lệnh.

Ví dụ:

- Chạy chương trình với tham số đầu vào và đầu ra:
 - `python source_code.py -i input.txt -o output.txt` sẽ đọc tệp đầu vào `input.txt` và ghi kết quả vào `output.txt`.
- Chạy chương trình với tất cả các tệp trong thư mục `Input`:
 - `python source_code.py -all` sẽ chạy chương trình với tất cả các tệp trong thư mục `Input`.

3 Thuật toán PLResolution

3.1 Chữ ký Hàm

1. FUNCTION `pl_resolution(alpha: Clause)` RETURNS (List of List of Clause, Boolean) **Mục đích:** Xác định xem cơ sở tri thức có suy diễn được một truy vấn đã cho bằng phương pháp suy diễn logic mệnh đề hay không.

3.2 Giải thích Chi tiết

3.2.1 Phủ định của Truy vấn

- `negated_alpha_clauses = NEGATE(alpha)`
- FOR EACH `negated_clause` IN `negated_alpha_clauses`:
- `ADD_CLAUSE_TO_KB(negated_clause)`

Hành động: Phủ định truy vấn α và thêm các mệnh đề phủ định vào cơ sở tri thức. **Lý do:** Bằng cách thêm $\neg\alpha$, chúng ta kiểm tra xem sự giao nhau giữa cơ sở tri thức và truy vấn phủ định có thỏa mãn hay không. **Ví dụ:** Nếu truy vấn α là $A \vee B$, thì: $\neg\alpha = \neg A \wedge \neg B = \{\{-A\}, \{-B\}\}$ Sau khi thêm vào cơ sở tri thức: $KB = \{\{-A\}, \{-B\}\}$, các mệnh đề ban đầu của KB

3.2.2 Khởi tạo Cấu trúc Dữ liệu

- `all_clauses = COPY_OF(KB)`
- `all_steps = EMPTY_LIST()`
- `all_resolutions = EMPTY_LIST()`

Hành động: Tạo bản sao của các mệnh đề và khởi tạo các danh sách để theo dõi quá trình suy diễn. **Lý do:** Những cấu trúc này giúp theo dõi quá trình suy diễn và tạo ra các mệnh đề mới.

3.2.3 Vòng lặp Chính của Suy Diễn

- WHILE TRUE:

- INCREMENT(loop_count)
- pairs = GENERATE_UNIQUE_PAIRS(all_clauses)
- step_clauses = EMPTY_LIST()
- FOR EACH (clause1, clause2) IN pairs:
- resolvents = RESOLVE(clause1, clause2)
- FOR EACH resolvent IN resolvents:
- IF resolvent IS EMPTY:
- ADD_TO(step_clauses, resolvent)
- ADD_TO(all_steps, COPY_OF(step_clauses))
- PRINT_RESOLUTIONS(all_resolutions)
- RETURN (all_steps, True)
- IF NOT EXISTS(resolvent, all_clauses) AND NOT EXISTS(resolvent, step_clauses):
- ADD_TO(step_clauses, resolvent)
- ADD_TO(all_resolutions, (clause1, clause2, resolvent))

Mục đích: Tiếp tục suy diễn các cặp mệnh đề cho đến khi tìm được mệnh đề rỗng hoặc không thể tạo ra mệnh đề mới.

Chi tiết vòng lặp con: 1. Lựa chọn Cặp Mệnh đề:

- pairs = GENERATE_UNIQUE_PAIRS(all_clauses)
- **Hành động:** Chọn tất cả các cặp mệnh đề duy nhất (c_i, c_j) .
- **Lý do:** Tìm ra các mệnh đề kết hợp mới từ các cặp này.

2. Tạo Mệnh đề Kết hợp (Resolvent):

- resolvents = RESOLVE(clause1, clause2)

- **Hành động:** Giải quyết các cặp mệnh đề để tạo ra các mệnh đề mới.
- **Lý do:** Áp dụng quy tắc suy diễn để loại bỏ các mệnh đề mâu thuẫn.

3. Kiểm tra Mệnh đề Rỗng:

- IF `resolvent` IS EMPTY:
- `ADD_TO(step_clauses, resolvent)`
- `ADD_TO(all_steps, COPY_OF(step_clauses))`
- `PRINT_RESOLUTIONS(all_resolutions)`
- `RETURN (all_steps, True)`

Hành động: Kiểm tra xem mệnh đề kết hợp có phải là mệnh đề rỗng không. **Lý do:** Mệnh đề rỗng chỉ ra sự mâu thuẫn, chứng minh rằng có sự suy diễn.

4. Thêm Mệnh Đề Không Lặp Lại:

- IF NOT EXISTS(`resolvent`, `all_clauses`) AND NOT EXISTS(`resolvent`, `step_clauses`):
- `ADD_TO(step_clauses, resolvent)`
- `ADD_TO(all_resolutions, (clause1, clause2, resolvent))`

Hành động: Thêm mệnh đề kết hợp nếu nó không bị trùng lặp và không phải mệnh đề đơn giản.

Lý do: Tránh thêm mệnh đề lặp lại và đảm bảo quá trình suy diễn hiệu quả.

Ví dụ: Giải quyết các mệnh đề sau: - Mệnh đề 1 = $\{A, B\}$ - Mệnh đề 2 = $\{-A, C\}$ Kết quả giải quyết: - Giải quyết Mệnh đề 1 và Mệnh đề 2 loại bỏ A và $-A$, tạo ra mệnh đề kết hợp: - $R1 = \{B, C\}$ Mệnh đề rỗng $\{\}$ chỉ ra sự mâu thuẫn.

3.2.4 Kiểm tra Kết luận

- IF `step_clauses` IS EMPTY:
- `ADD_TO(all_steps, step_clauses)`
- `PRINT_RESOLUTIONS(all_resolutions)`

- RETURN (all_steps, False)

Hành động: Nếu không có mệnh đề mới được thêm vào, kết luận rằng không có sự suy diễn. **Lý do:** Nếu không có sự mâu thuẫn nào, thì $KB \wedge \neg\alpha$ là có thể thỏa mãn.

3.2.5 Thêm Mệnh Đề Mới và Lặp Lại

- SORT_AND_DEDUPLICATE(step_clauses)
- ADD_TO(all_steps, step_clauses)
- EXTEND(all_clauses, step_clauses)
- PRINT_RESOLUTIONS(all_resolutions)

Hành động: Sắp xếp, loại bỏ mệnh đề trùng lặp và thêm mệnh đề mới vào vòng lặp tiếp theo. **Lý do:** Mở rộng cơ sở tri thức để phục vụ cho các vòng lặp suy diễn tiếp theo.

Ví dụ về Toàn Bộ Quá Trình: Nếu ta giải quyết: - Mệnh đề 1 = $\{A, B\}$ - Mệnh đề 2 = $\{-A, C\}$
Ta sẽ có: - Mệnh đề kết hợp = $\{B, C\}$

Sau khi thêm các mệnh đề mới và lặp lại quá trình suy diễn, ta kiểm tra xem có mệnh đề rỗng được sinh ra không (chứng minh suy diễn). Nếu không có mệnh đề mới được sinh ra sau một vài lần lặp, ta kết luận không thể suy diễn được.

4 Giải Thích Chi Tiết Các Trường Hợp Kiểm Tra

4.1 input01.txt

- 1. $\neg A$
- 2. $A \text{ OR } B$
- 3. $B \text{ OR } \neg C$
- 4. $A \text{ OR } \neg B \text{ OR } C$
- 5. $\neg B$

Knowledge Base after adding negation of alpha:

- $\neg A \text{ OR } B$
- $B \text{ OR } \neg C$
- $A \text{ OR } \neg B \text{ OR } C$
- $\neg B$
- A

Loop 1:

- Resolving: $\neg A \text{ OR } B$ with $\neg B \Rightarrow \text{Get } \neg A$
- Resolving: $\neg A \text{ OR } B$ with $A \Rightarrow \text{Get } B$
- Resolving: $B \text{ OR } \neg C$ with $\neg B \Rightarrow \text{Get } \neg C$

Loop 2:

- Resolving: B with $\neg B \Rightarrow \text{Get } \{\}$
- Resolving: $A \text{ OR } \neg B \text{ OR } C$ with $B \Rightarrow \text{Get } A \text{ OR } C$
- Resolving: $A \text{ OR } \neg B \text{ OR } C$ with $\neg C \Rightarrow \text{Get } A \text{ OR } \neg B$
- Resolving: $A \text{ OR } \neg B \text{ OR } C$ with $\neg A \Rightarrow \text{Get } \neg B \text{ OR } C$

Output:

- We got $\neg A$ after running PL-resolution, so KB entails α

Final Knowledge Base:

- 1. $A \Rightarrow B$
- 2. A
- 3. B
- 4. $\neg C$
- 5. $\{\}$

- 6. $A \text{ OR } C$
- 7. $A \text{ OR } \neg B$
- 8. $\neg B \text{ OR } C$
- 9. YES

4.2 input02.txt

- 1. A
- 2. $\neg A \text{ OR } B$
- 3. $\neg B \text{ OR } C$
- 4. $A \text{ OR } \neg B$
- 5. $C \text{ OR } \neg A$

Knowledge Base after adding negation of alpha:

- $\neg A \text{ OR } B$
- $\neg B \text{ OR } C$
- $A \text{ OR } \neg B$
- $\neg A \text{ OR } C$
- $\neg A$

Loop 1:

- Resolving: $\neg A \text{ OR } B$ with $\neg B \text{ OR } C \Rightarrow \text{Result: } \neg A \text{ OR } C$
- Resolving: $A \text{ OR } \neg B$ with $\neg A \text{ OR } C \Rightarrow \text{Result: } \neg B \text{ OR } C$
- Resolving: $A \text{ OR } \neg B$ with $\neg A \Rightarrow \text{Result: } \neg B$

Loop 2:

- Resolving: $\neg A \text{ OR } B$ with $\neg B \text{ OR } C \Rightarrow \text{Result: } \neg A \text{ OR } C$
- Resolving: $A \text{ OR } \neg B$ with $\neg A \text{ OR } C \Rightarrow \text{Result: } \neg B \text{ OR } C$
- Resolving: $A \text{ OR } \neg B$ with $\neg A \Rightarrow \text{Result: } \neg B$
- Resolving: $\neg A \text{ OR } B$ with $\neg B \text{ OR } C \Rightarrow \text{Result: } \neg A \text{ OR } C$
- Resolving: $\neg A \text{ OR } B$ with $\neg B \Rightarrow \text{Result: } \neg A$
- Resolving: $A \text{ OR } \neg B$ with $\neg A \text{ OR } C \Rightarrow \text{Result: } \neg B \text{ OR } C$
- Resolving: $A \text{ OR } \neg B$ with $\neg A \Rightarrow \text{Result: } \neg B$

Output:

- We got $\neg B$ after running PL-resolution, so KB does not entail α

Final Knowledge Base:

- 1. $\neg B$
- 2. $\{\}$
- 3. NO

4.3 input03.txt

- 1. A
- 2. $\neg A \text{ OR } B$
- 3. $B \text{ OR } C$
- 4. $\neg C \text{ OR } D$
- 5. $D \text{ OR } \neg A$
- 6. $A \text{ OR } C$

Knowledge Base after adding negation of alpha:

- $\neg A \text{ OR } B$
- $B \text{ OR } C$
- $\neg C \text{ OR } D$
- $\neg A \text{ OR } D$
- $A \text{ OR } C$
- $\neg A$

Loop 1:

- Resolving: $\neg A \text{ OR } B$ with $A \text{ OR } C \Rightarrow \text{Result: } B \text{ OR } C$
- Resolving: $B \text{ OR } C$ with $\neg C \text{ OR } D \Rightarrow \text{Result: } B \text{ OR } D$
- Resolving: $\neg C \text{ OR } D$ with $A \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } D$
- Resolving: $\neg A \text{ OR } D$ with $A \text{ OR } C \Rightarrow \text{Result: } C \text{ OR } D$
- Resolving: $A \text{ OR } C$ with $\neg A \Rightarrow \text{Result: } C$

Loop 2:

- Resolving: $\neg A \text{ OR } B$ with $A \text{ OR } C \Rightarrow \text{Result: } B \text{ OR } C$
- Resolving: $B \text{ OR } C$ with $\neg C \text{ OR } D \Rightarrow \text{Result: } B \text{ OR } D$
- Resolving: $\neg C \text{ OR } D$ with $A \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } D$
- Resolving: $\neg A \text{ OR } D$ with $A \text{ OR } C \Rightarrow \text{Result: } C \text{ OR } D$
- Resolving: $A \text{ OR } C$ with $\neg A \Rightarrow \text{Result: } C$
- Resolving: $\neg A \text{ OR } B$ with $A \text{ OR } C \Rightarrow \text{Result: } B \text{ OR } C$
- Resolving: $\neg A \text{ OR } B$ with $A \text{ OR } D \Rightarrow \text{Result: } B \text{ OR } D$
- Resolving: $B \text{ OR } C$ with $\neg C \text{ OR } D \Rightarrow \text{Result: } B \text{ OR } D$
- Resolving: $\neg C \text{ OR } D$ with $A \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } D$

- Resolving: $-C \text{ OR } D$ with $C \Rightarrow$ Result: D
- Resolving: $-C \text{ OR } D$ with $C \text{ OR } D \Rightarrow$ Result: D
- Resolving: $-A \text{ OR } D$ with $A \text{ OR } C \Rightarrow$ Result: $C \text{ OR } D$
- Resolving: $-A \text{ OR } D$ with $A \text{ OR } D \Rightarrow$ Result: D
- Resolving: $A \text{ OR } C$ with $-A \Rightarrow$ Result: C
- Resolving: $-A$ with $A \text{ OR } D \Rightarrow$ Result: D

Loop 3:

- Resolving: $-A \text{ OR } B$ with $A \text{ OR } C \Rightarrow$ Result: $B \text{ OR } C$
- Resolving: $B \text{ OR } C$ with $-C \text{ OR } D \Rightarrow$ Result: $B \text{ OR } D$
- Resolving: $-C \text{ OR } D$ with $A \text{ OR } C \Rightarrow$ Result: $A \text{ OR } D$
- Resolving: $-A \text{ OR } D$ with $A \text{ OR } C \Rightarrow$ Result: $C \text{ OR } D$
- Resolving: $A \text{ OR } C$ with $-A \Rightarrow$ Result: C
- Resolving: $-A \text{ OR } B$ with $A \text{ OR } C \Rightarrow$ Result: $B \text{ OR } C$
- Resolving: $-A \text{ OR } B$ with $A \text{ OR } D \Rightarrow$ Result: $B \text{ OR } D$
- Resolving: $B \text{ OR } C$ with $-C \text{ OR } D \Rightarrow$ Result: $B \text{ OR } D$
- Resolving: $-C \text{ OR } D$ with $A \text{ OR } C \Rightarrow$ Result: $A \text{ OR } D$
- Resolving: $-C \text{ OR } D$ with $C \Rightarrow$ Result: D
- Resolving: $-C \text{ OR } D$ with $C \text{ OR } D \Rightarrow$ Result: D
- Resolving: $-A \text{ OR } D$ with $A \text{ OR } C \Rightarrow$ Result: $C \text{ OR } D$
- Resolving: $-A \text{ OR } D$ with $A \text{ OR } D \Rightarrow$ Result: D
- Resolving: $A \text{ OR } C$ with $-A \Rightarrow$ Result: C
- Resolving: $-A$ with $A \text{ OR } D \Rightarrow$ Result: D

Output:

- 4
- C
- $A \text{ OR } D$
- $B \text{ OR } D$
- $C \text{ OR } D$
- 1
- D
- 0
- NO

4.4 input04.txt

- 1. B
- 2. 4
- 3. $\neg A \text{ OR } B$
- 4. $A \text{ OR } \neg B$
- 5. $B \text{ OR } C$
- 6. $C \text{ OR } \neg A$

Knowledge Base after adding negation of alpha:

- $\neg A \text{ OR } B$
- $A \text{ OR } \neg B$
- $B \text{ OR } C$

- $\neg A \text{ OR } C$
- $\neg B$

Loop 1:

- Resolving: $\neg A \text{ OR } B$ with $\neg B \Rightarrow \text{Result: } \neg A$
- Resolving: $A \text{ OR } \neg B$ with $B \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } C$
- Resolving: $A \text{ OR } \neg B$ with $\neg A \text{ OR } C \Rightarrow \text{Result: } \neg B \text{ OR } C$
- Resolving: $B \text{ OR } C$ with $\neg B \Rightarrow \text{Result: } C$

Loop 2:

- Resolving: $\neg A \text{ OR } B$ with $\neg B \Rightarrow \text{Result: } \neg A$
- Resolving: $A \text{ OR } \neg B$ with $B \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } C$
- Resolving: $A \text{ OR } \neg B$ with $\neg A \text{ OR } C \Rightarrow \text{Result: } \neg B \text{ OR } C$
- Resolving: $B \text{ OR } C$ with $\neg B \Rightarrow \text{Result: } C$
- Resolving: $\neg A \text{ OR } B$ with $\neg B \Rightarrow \text{Result: } \neg A$
- Resolving: $\neg A \text{ OR } B$ with $\neg B \text{ OR } C \Rightarrow \text{Result: } \neg A \text{ OR } C$
- Resolving: $\neg A \text{ OR } B$ with $A \text{ OR } C \Rightarrow \text{Result: } B \text{ OR } C$
- Resolving: $A \text{ OR } \neg B$ with $B \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } C$
- Resolving: $A \text{ OR } \neg B$ with $\neg A \text{ OR } C \Rightarrow \text{Result: } \neg B \text{ OR } C$
- Resolving: $A \text{ OR } \neg B$ with $\neg A \Rightarrow \text{Result: } \neg B$
- Resolving: $B \text{ OR } C$ with $\neg B \Rightarrow \text{Result: } C$
- Resolving: $B \text{ OR } C$ with $\neg B \text{ OR } C \Rightarrow \text{Result: } C$
- Resolving: $\neg A \text{ OR } C$ with $A \text{ OR } C \Rightarrow \text{Result: } C$
- Resolving: $\neg A$ with $A \text{ OR } C \Rightarrow \text{Result: } C$

Output:

- 4
- $\neg A$
- C
- $A \text{ OR } C$
- $\neg B \text{ OR } C$
- 0
- NO

4.5 input05.txt

- 1. C
- 2. 3
- 3. $\neg A \text{ OR } B$
- 4. $B \text{ OR } \neg C$
- 5. $A \text{ OR } C$

Knowledge Base after adding negation of alpha:

- $\neg A \text{ OR } B$
- $B \text{ OR } \neg C$
- $A \text{ OR } C$
- $\neg C$

Loop 1:

- Resolving: $\neg A \text{ OR } B$ with $A \text{ OR } C \Rightarrow \text{Result: } B \text{ OR } C$

- Resolving: $B \text{ OR } -C$ with $A \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } B$
- Resolving: $A \text{ OR } C$ with $-C \Rightarrow \text{Result: } A$

Loop 2:

- Resolving: $-A \text{ OR } B$ with $A \text{ OR } C \Rightarrow \text{Result: } B \text{ OR } C$
- Resolving: $B \text{ OR } -C$ with $A \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } B$
- Resolving: $A \text{ OR } C$ with $-C \Rightarrow \text{Result: } A$
- Resolving: $-A \text{ OR } B$ with $A \text{ OR } C \Rightarrow \text{Result: } B \text{ OR } C$
- Resolving: $-A \text{ OR } B$ with $A \Rightarrow \text{Result: } B$
- Resolving: $-A \text{ OR } B$ with $A \text{ OR } B \Rightarrow \text{Result: } B$
- Resolving: $B \text{ OR } -C$ with $A \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } B$
- Resolving: $B \text{ OR } -C$ with $B \text{ OR } C \Rightarrow \text{Result: } B$
- Resolving: $A \text{ OR } C$ with $-C \Rightarrow \text{Result: } A$
- Resolving: $-C$ with $B \text{ OR } C \Rightarrow \text{Result: } B$

Loop 3:

- Resolving: $-A \text{ OR } B$ with $A \text{ OR } C \Rightarrow \text{Result: } B \text{ OR } C$
- Resolving: $B \text{ OR } -C$ with $A \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } B$
- Resolving: $A \text{ OR } C$ with $-C \Rightarrow \text{Result: } A$
- Resolving: $-A \text{ OR } B$ with $A \text{ OR } C \Rightarrow \text{Result: } B \text{ OR } C$
- Resolving: $-A \text{ OR } B$ with $A \Rightarrow \text{Result: } B$
- Resolving: $-A \text{ OR } B$ with $A \text{ OR } B \Rightarrow \text{Result: } B$
- Resolving: $B \text{ OR } -C$ with $A \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } B$
- Resolving: $B \text{ OR } -C$ with $B \text{ OR } C \Rightarrow \text{Result: } B$

- Resolving: $A \text{ OR } C$ with $\neg C \Rightarrow \text{Result: } A$
- Resolving: $\neg C$ with $B \text{ OR } C \Rightarrow \text{Result: } B$

Output:

- 3
- A
- $A \text{ OR } B$
- $B \text{ OR } C$
- 1
- B
- 0
- NO

4.6 input06.txt

- 1. $\neg A$
- 2. 5
- 3. $A \text{ OR } \neg B$
- 4. $\neg B \text{ OR } C$
- 5. $C \text{ OR } D$
- 6. $\neg D \text{ OR } A$
- 7. $B \text{ OR } C$

Knowledge Base after adding negation of alpha:

- $A \text{ OR } \neg B$

- $\neg B \text{ OR } C$
- $C \text{ OR } D$
- $A \text{ OR } \neg D$
- $B \text{ OR } C$
- A

Loop 1:

- Resolving: $A \text{ OR } \neg B$ with $B \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } C$
- Resolving: $\neg B \text{ OR } C$ with $B \text{ OR } C \Rightarrow \text{Result: } C$
- Resolving: $C \text{ OR } D$ with $A \text{ OR } \neg D \Rightarrow \text{Result: } A \text{ OR } C$

Loop 2:

- Resolving: $A \text{ OR } \neg B$ with $B \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } C$
- Resolving: $\neg B \text{ OR } C$ with $B \text{ OR } C \Rightarrow \text{Result: } C$
- Resolving: $C \text{ OR } D$ with $A \text{ OR } \neg D \Rightarrow \text{Result: } A \text{ OR } C$
- Resolving: $A \text{ OR } \neg B$ with $B \text{ OR } C \Rightarrow \text{Result: } A \text{ OR } C$
- Resolving: $\neg B \text{ OR } C$ with $B \text{ OR } C \Rightarrow \text{Result: } C$
- Resolving: $C \text{ OR } D$ with $A \text{ OR } \neg D \Rightarrow \text{Result: } A \text{ OR } C$

Output:

- 2
- C
- $A \text{ OR } C$
- 0
- NO

4.7 input07.txt

- 1. C
- 2. 4
- 3. $A \text{ OR } B$
- 4. $B \text{ OR } C$
- 5. $\neg A \text{ OR } D$
- 6. $\neg D \text{ OR } C$

Knowledge Base after adding negation of alpha:

- $A \text{ OR } B$
- $B \text{ OR } C$
- $\neg A \text{ OR } D$
- $C \text{ OR } \neg D$
- $\neg C$

Loop 1:

- Resolving: $A \text{ OR } B$ with $\neg A \text{ OR } D \Rightarrow \text{Result: } B \text{ OR } D$
- Resolving: $B \text{ OR } C$ with $\neg C \Rightarrow \text{Result: } B$
- Resolving: $\neg A \text{ OR } D$ with $C \text{ OR } \neg D \Rightarrow \text{Result: } \neg A \text{ OR } C$
- Resolving: $C \text{ OR } \neg D$ with $\neg C \Rightarrow \text{Result: } \neg D$

Loop 2:

- Resolving: $A \text{ OR } B$ with $\neg A \text{ OR } D \Rightarrow \text{Result: } B \text{ OR } D$
- Resolving: $B \text{ OR } C$ with $\neg C \Rightarrow \text{Result: } B$
- Resolving: $\neg A \text{ OR } D$ with $C \text{ OR } \neg D \Rightarrow \text{Result: } \neg A \text{ OR } C$

- Resolving: $C \text{ OR } \neg D$ with $\neg C \Rightarrow \text{Result: } \neg D$
- Resolving: $A \text{ OR } B$ with $\neg A \text{ OR } D \Rightarrow \text{Result: } B \text{ OR } D$
- Resolving: $A \text{ OR } B$ with $\neg A \text{ OR } C \Rightarrow \text{Result: } B \text{ OR } C$
- Resolving: $B \text{ OR } C$ with $\neg C \Rightarrow \text{Result: } B$
- Resolving: $\neg A \text{ OR } D$ with $C \text{ OR } \neg D \Rightarrow \text{Result: } \neg A \text{ OR } C$
- Resolving: $\neg A \text{ OR } D$ with $\neg D \Rightarrow \text{Result: } \neg A$
- Resolving: $C \text{ OR } \neg D$ with $\neg C \Rightarrow \text{Result: } \neg D$
- Resolving: $C \text{ OR } \neg D$ with $B \text{ OR } D \Rightarrow \text{Result: } B \text{ OR } C$
- Resolving: $\neg C$ with $\neg A \text{ OR } C \Rightarrow \text{Result: } \neg A$
- Resolving: $\neg D$ with $B \text{ OR } D \Rightarrow \text{Result: } B$

Loop 3:

- Resolving: $A \text{ OR } B$ with $\neg A \text{ OR } D \Rightarrow \text{Result: } B \text{ OR } D$
- Resolving: $B \text{ OR } C$ with $\neg C \Rightarrow \text{Result: } B$
- Resolving: $\neg A \text{ OR } D$ with $C \text{ OR } \neg D \Rightarrow \text{Result: } \neg A \text{ OR } C$
- Resolving: $C \text{ OR } \neg D$ with $\neg C \Rightarrow \text{Result: } \neg D$
- Resolving: $A \text{ OR } B$ with $\neg A \text{ OR } D \Rightarrow \text{Result: } B \text{ OR } D$
- Resolving: $A \text{ OR } B$ with $\neg A \text{ OR } C \Rightarrow \text{Result: } B \text{ OR } C$
- Resolving: $B \text{ OR } C$ with $\neg C \Rightarrow \text{Result: } B$
- Resolving: $\neg A \text{ OR } D$ with $C \text{ OR } \neg D \Rightarrow \text{Result: } \neg A \text{ OR } C$
- Resolving: $\neg A \text{ OR } D$ with $\neg D \Rightarrow \text{Result: } \neg A$
- Resolving: $C \text{ OR } \neg D$ with $\neg C \Rightarrow \text{Result: } \neg D$
- Resolving: $C \text{ OR } \neg D$ with $B \text{ OR } D \Rightarrow \text{Result: } B \text{ OR } C$

- Resolving: $\neg C$ with $\neg A \text{ OR } C \Rightarrow \text{Result: } \neg A$
- Resolving: $\neg D$ with $B \text{ OR } D \Rightarrow \text{Result: } B$

Output:

- 4
- B
- $\neg D$
- $\neg A \text{ OR } C$
- $B \text{ OR } D$
- 1
- $\neg A$
- 0
- NO

4.8 input08.txt

- 1. E
- 2. 4
- 3. $\neg A \text{ OR } B$
- 4. $B \text{ OR } \neg C$
- 5. $C \text{ OR } \neg D$
- 6. $D \text{ OR } E$

Knowledge Base after adding negation of alpha:

- $\neg A \text{ OR } B$

- $B \text{ OR } \neg C$
- $C \text{ OR } \neg D$
- $D \text{ OR } E$
- $\neg E$

Loop 1:

- Resolving: $B \text{ OR } \neg C$ with $C \text{ OR } \neg D \Rightarrow \text{Result: } B \text{ OR } \neg D$
- Resolving: $C \text{ OR } \neg D$ with $D \text{ OR } E \Rightarrow \text{Result: } C \text{ OR } E$
- Resolving: $D \text{ OR } E$ with $\neg E \Rightarrow \text{Result: } D$

Loop 2:

- Resolving: $B \text{ OR } \neg C$ with $C \text{ OR } \neg D \Rightarrow \text{Result: } B \text{ OR } \neg D$
- Resolving: $C \text{ OR } \neg D$ with $D \text{ OR } E \Rightarrow \text{Result: } C \text{ OR } E$
- Resolving: $D \text{ OR } E$ with $\neg E \Rightarrow \text{Result: } D$
- Resolving: $B \text{ OR } \neg C$ with $C \text{ OR } \neg D \Rightarrow \text{Result: } B \text{ OR } \neg D$
- Resolving: $B \text{ OR } \neg C$ with $C \text{ OR } E \Rightarrow \text{Result: } B \text{ OR } E$
- Resolving: $C \text{ OR } \neg D$ with $D \text{ OR } E \Rightarrow \text{Result: } C \text{ OR } E$
- Resolving: $C \text{ OR } \neg D$ with $D \Rightarrow \text{Result: } C$
- Resolving: $D \text{ OR } E$ with $\neg E \Rightarrow \text{Result: } D$
- Resolving: $D \text{ OR } E$ with $B \text{ OR } \neg D \Rightarrow \text{Result: } B \text{ OR } E$
- Resolving: $\neg E$ with $C \text{ OR } E \Rightarrow \text{Result: } C$
- Resolving: $B \text{ OR } \neg D$ with $D \Rightarrow \text{Result: } B$

Loop 3:

- Resolving: $B \text{ OR } \neg C$ with $C \text{ OR } \neg D \Rightarrow \text{Result: } B \text{ OR } \neg D$

- Resolving: $C \text{ OR } \neg D$ with $D \text{ OR } E \Rightarrow$ Result: $C \text{ OR } E$
- Resolving: $D \text{ OR } E$ with $\neg E \Rightarrow$ Result: D
- Resolving: $B \text{ OR } \neg C$ with $C \text{ OR } \neg D \Rightarrow$ Result: $B \text{ OR } \neg D$
- Resolving: $B \text{ OR } \neg C$ with $C \text{ OR } E \Rightarrow$ Result: $B \text{ OR } E$
- Resolving: $C \text{ OR } \neg D$ with $D \text{ OR } E \Rightarrow$ Result: $C \text{ OR } E$
- Resolving: $C \text{ OR } \neg D$ with $D \Rightarrow$ Result: C
- Resolving: $D \text{ OR } E$ with $\neg E \Rightarrow$ Result: D
- Resolving: $D \text{ OR } E$ with $B \text{ OR } \neg D \Rightarrow$ Result: $B \text{ OR } E$
- Resolving: $\neg E$ with $C \text{ OR } E \Rightarrow$ Result: C
- Resolving: $\neg E$ with $B \text{ OR } E \Rightarrow$ Result: B
- Resolving: $B \text{ OR } \neg D$ with $D \Rightarrow$ Result: B

Output:

- 3
- D
- $C \text{ OR } E$
- $B \text{ OR } \neg D$
- 3
- B
- C
- $B \text{ OR } E$
- 0
- NO

5 Đánh Giá Phương Pháp Giải Quyết cho Logic Phân Tích

5.1 Ưu Điểm

- **Tính Đầy Đủ:** Phương pháp giải quyết là đầy đủ đối với logic phân tích, nghĩa là nếu một công thức không thể thỏa mãn, phương pháp sẽ dẫn đến một mệnh đề rỗng, xác nhận tính không thể thỏa mãn của công thức đó.
- **Đơn Giản:** Phương pháp này rất đơn giản và chỉ dựa vào một quy tắc suy diễn duy nhất, điều này làm cho nó dễ dàng triển khai hơn so với các phương pháp suy diễn logic khác.

5.2 Nhược Điểm

- **Tăng Trưởng Theo Mũ:** Không gian tìm kiếm có thể tăng trưởng theo hàm mũ khi các kết quả giải quyết mới được sinh ra. Điều này làm cho phương pháp trở nên tốn kém về mặt tính toán và có thể gây ra sự kém hiệu quả trong thực tế, đặc biệt là khi cơ sở tri thức rất lớn.
- **Sự Dư Thừa:** Phương pháp có thể sinh ra nhiều mệnh đề dư thừa không đóng góp gì vào việc tìm kiếm giải pháp, làm tăng khối lượng tính toán mà không mang lại giá trị thêm.

6 Giải Pháp Đề Xuất Cho Vấn Đề Cụ Thể

Để giải quyết các thách thức về sự không hiệu quả và dư thừa trong phương pháp giải quyết, tôi đề xuất các cải tiến sau:

- **Lưu Trữ Kết Quả (Memoization):** Sử dụng kỹ thuật lưu trữ kết quả để lưu trữ và tái sử dụng kết quả của các phép giải quyết trước đó, giúp tránh việc tính toán lại và tăng tốc quá trình giải quyết.
- **Kiểm Tra Sự Bao Phủ (Subsumption Checking):** Tích hợp kiểm tra sự bao phủ để loại bỏ các mệnh đề bị bao phủ bởi các mệnh đề tổng quát hơn, từ đó giảm số lượng mệnh đề được xem xét tại mỗi bước.
- **Xử Lý Song Song (Parallel Processing):** Tận dụng xử lý song song để xử lý đồng thời nhiều cặp mệnh đề, giúp giảm thời gian tính toán tổng thể.

Tài Liệu Tham Khảo

1. github <https://github.com/phungvhbui/AI-Lab-PL-Resolution>
2. GeeksforGeeks. . Resolution algorithm in artificial intelligence. GeeksforGeeks. Retrieved from <https://www.geeksforgeeks.org/resolution-algorithm-in-artificial-intelligence/>