

FundaOAI-HW02-Logic

Thông tin chung

Thông tin lớp

- **Mã lớp:** 22_4
- **Giảng viên:** Thầy Bùi Duy Đăng, Thầy Lê Nhựt Nam

Thông tin sinh viên

- **Mã sinh viên:** 22120412
- **Họ và tên:** Nguyễn Anh Tường

Hướng dẫn sử dụng

- Chạy `menu()` trong `Source/Main.py`
- Chọn tệp cần chạy.

Định dạng đầu ra trên cửa sổ terminal:

```
# Cú pháp: [literal A] + [literal B] => [literal C]
['D', 'B', '-C'] + ['C'] => ['B', 'D'] # Ví dụ
['A', '-B'] + ['A', 'B', 'D'] => ['A', 'D']
['-D', 'C'] + ['B', 'D'] => ['B', 'C']
# ...
```

Tiêu chí đánh giá

STT	Đặc điểm tiêu chí	Điểm	IsDone
1	Đọc dữ liệu đầu vào và lưu trong cấu trúc dữ liệu phù hợp	0.5	Có
2	Triển khai thuật toán để giải quyết trên logic mệnh đề	1.0	Có
3	Các bước suy diễn tạo ra đủ mệnh đề và kết luận đúng	2.5	Có
4	Tuân thủ mô tả định dạng của bài tập	0.5	Có
5	Báo cáo và đánh giá trường hợp thử nghiệm	0.5	Có

Trình bày báo cáo

1. Triển khai thuật toán để giải quyết logic mệnh đề

function PL-RESOLUTION(KB, α) **returns** *true* or *false*

inputs: KB , the knowledge base, a sentence in propositional logic
 α , 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*

$clauses \leftarrow clauses \cup new$

2. Một số hàm quan trọng trong quá trình xử lý

- **def negate_atom(alpha):** Phủ định một literal.
 - **Ví dụ:** chuyển 'A' thành '-A'.
- **def negate_clause(clause):** Phủ định một câu, sử dụng cho α .
 - **Ví dụ:** chuyển ['A', 'B', '-C'] thành ['-A', '-B', 'C'].
- **def clauses_cleaned(clauses):** Dọn sạch một danh sách gồm các mệnh đề, bao gồm việc loại bỏ các trường hợp đối dấu nội tại và trùng lặp:
 - **Ví dụ:**

```
['A', '-A'] -> []
[['A', 'B'], ['A', 'B']] -> [['A', 'B']]
```

def resolved(*clause_i*, *clause_j*): Hàm hợp giải hai mệnh đề.

- Nếu kết quả hợp giải là rỗng thì sẽ trả về `['{}']`.
- Ngược lại sẽ trả về kết quả hợp giải của hai mệnh đề.
- **Ví dụ:**

```
['A', '-B'] + ['A', 'B', 'D'] => ['A', 'D']
['D', 'B', '-C'] + ['C'] => ['B', 'D']
['D', 'B', '-C'] + ['A', 'B', 'C'] => ['A', 'B', 'D']
```

- **def PL_resolution_output(*alpha*, *KB*, *filepath*)**: Hàm thực hiện hợp giải và ghi kết quả vào file output theo đường dẫn *filepath* được đặt sẵn.

- **Ví dụ:**

■ *Input.txt*

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

■ *Output.txt*

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

■ *Terminal*

```
Select input file from the following list:
```

1. input_1.txt
2. input_2.txt
3. input_3.txt
4. input_4.txt
5. input_5.txt
6. input_6.txt
7. input_7.txt
8. input_8.txt
9. input_9.txt
10. input_10.txt

```
Enter the file number (1-10): 1
```

```
Selected file: input_1.txt
```

```
Alpha: ['-A']
```

```
The number of sentences in KB: 4
```

```
KB: [['B', '-A'], ['-C', 'B'], ['A', 'C', '-B'], ['-B']]
```

```
The run 1:
```

```
-----  
['B', '-A'] + ['A', 'C', '-B'] => ['C']  
-----
```

```
['B', '-A'] + ['-B'] => ['-A']  
-----
```

```
['B', '-A'] + ['A'] => ['B']  
-----
```

```
['-C', 'B'] + ['-B'] => ['-C']  
-----
```

```
The run 2:
```

```
-----  
['A', 'C', '-B'] + ['-A'] => ['-B', 'C']  
-----
```

```
['A', 'C', '-B'] + ['B'] => ['A', 'C']  
-----
```

```
['A', 'C', '-B'] + ['-C'] => ['A', '-B']  
-----
```

```
['-B'] + ['B'] => ['{}']
```

Đánh giá ngắn gọn về ưu điểm và khuyết điểm của giải thuật hợp giải trên logic mệnh đề, cũng như đề xuất giải pháp để khắc phục vấn đề.

Đánh giá giải thuật hợp giải trên logic mệnh đề

Ưu điểm

1. Hiệu quả về tính toán:

- Hoạt động trực tiếp trên logic mệnh đề, không yêu cầu chuyển đổi sang dạng khác như đại số Boole.
- Dễ dàng áp dụng cho các bài toán kiểm tra tính thỏa mãn (satisfiability) hoặc chứng minh định lý.

2. Đơn giản về mặt khái niệm:

- Quy tắc hợp giải (resolution) dễ hiểu và triển khai.
- Không cần mở rộng ngôn ngữ hay cấu trúc phức tạp, chỉ cần làm việc với các mệnh đề dưới dạng chuẩn tắc Chia (CNF).

3. Chứng minh bằng phản chứng:

- Có thể áp dụng để kiểm tra tính nhất quán hoặc tìm mâu thuẫn trong một tập hợp mệnh đề.

4. Tính hoàn chỉnh:

- Giải thuật hợp giải là **hoàn chỉnh** cho logic mệnh đề, nghĩa là nó đảm bảo tìm ra kết quả nếu tập hợp mệnh đề không thỏa mãn.

Khuyết điểm

1. Không hiệu quả với tập dữ liệu lớn:

- Khi số lượng mệnh đề và các biến tăng lên, không gian tìm kiếm trở nên lớn, dẫn đến hiệu năng giảm mạnh (bùng nổ tổ hợp).

2. Không trực quan với bài toán cụ thể:

- Giải thuật này làm việc dựa trên các quy tắc hình thức, không khai thác thông tin đặc thù của bài toán.

3. Dễ gặp vấn đề dư thừa:

- Có thể sinh ra nhiều mệnh đề trung gian không cần thiết, làm tăng kích thước tập hợp cần xử lý.

4. Khó áp dụng trong logic bậc cao hơn:

- Hợp giải hiệu quả với logic mệnh đề nhưng không dễ mở rộng trực tiếp sang logic bậc cao (predicate logic) mà không cần bổ sung các bước phức tạp.

Đề xuất giải pháp khắc phục

1. Cải thiện hiệu năng:

- **Tối ưu hóa biểu diễn mệnh đề:** Sử dụng cấu trúc dữ liệu nén (như BDD - Binary Decision Diagram) để giảm kích thước tập hợp mệnh đề.
- **Hạn chế mệnh đề dư thừa:** Áp dụng heuristics (thuật toán gợi ý) để chọn lọc mệnh đề cần hợp giải, tránh sinh ra các mệnh đề không cần thiết.

2. Chia nhỏ bài toán:

- Phân chia bài toán lớn thành các bài toán nhỏ hơn, giải quyết từng phần độc lập trước khi hợp nhất kết quả.

3. Kết hợp với các kỹ thuật khác:

- **Backtracking:** Kết hợp hợp giải với kỹ thuật quay lui để giảm không gian tìm kiếm.
- **SAT Solver:** Sử dụng các giải pháp tối ưu hóa hiện đại từ SAT solvers (như MiniSAT, Z3) để tăng tốc độ giải quyết bài toán.

4. Chuyển đổi sang các phương pháp mạnh hơn:

- Đối với các bài toán logic bậc cao, chuyển sang sử dụng hợp giải trên logic bậc cao hoặc áp dụng các kỹ thuật như lập luận dựa trên mô hình (model-based reasoning).

5. Song song hóa xử lý:

- Chia quy trình hợp giải thành các tác vụ có thể thực hiện song song trên nhiều bộ xử lý để cải thiện hiệu suất.

Ghi nhận và Đánh giá các trường hợp thử nghiệm

1. Các trường hợp alpha chỉ có 1 literal

- Các trường hợp dễ với KB ít hơn 4 mệnh đề: *input_1, input_2, input_3, input_4*.
- Các trường hợp trung bình với KB nhiều hơn 4 mệnh đề: *input_5, input_6*.

2. Các trường hợp alpha lớn hơn 2 literal

- Các trường hợp trung bình khó với KB bé hơn hoặc bằng 4 mệnh đề: *input_7, input_8*.
- Các trường hợp khó với KB lớn hơn 4 mệnh đề: *input_9, input_10*.

3. Ghi nhận kết quả

- Các trường hợp có lời giải: 1, 3, 8, 9.
- Các trường hợp không có lời giải: 2, 4, 5, 6, 7, 10.

Diễn giải các trường hợp

1. Mẫu thử số 1:

```
Submission > 22120412 > src > Input > ≡ input_1.txt
1      -A
2      4
3      -A OR B
4      B OR -C
5      A OR -B OR C
6      -B
```

Đề bài:

```
Submission > 22120412 > src > Output > ≡ output_1.txt
1      4
2      C
3      -A
4      B
5      -C
6      4
7      -B OR C
8      A OR C
9      A OR -B
10     {}
11     YES
```

Lời giải:

2. Mẫu thử số 2:

```
Submission > 22120412 > src > Input > ≡ input_2.txt
1      -D
2      4
3      B OR -C OR D
4      A OR -B
5      C OR -D
6      C OR A OR B
```

Đề bài:

```
Submission > 22120412 > src > Output > ≡ output_2.txt
1      5
2      A OR -C OR D
3      B
4      A OR B OR D
5      A OR C
6      C
7      4
8      B OR D
9      A
10     A OR D
11     A OR B OR C
12     1
13     B OR C
14     Ø
15     NO
```

Lời giải:

3. Mẫu thử số 3:

```
Submission > 22120412 > src > Input > ≡ input_3.txt
1      B
2      3
3      -A OR B OR C
4      -C OR D
5      A OR -D
```

Đề bài:

```
Submission > 22120412 > src > Output > ≡ output_3.txt
1      4
2      -A OR B OR D
3      B OR C OR -D
4      -A OR C
5      A OR -C
6      6
7      B
8      -A OR D
9      C OR -D
10     B OR -C OR D
11     A OR B OR -D
12     {}
13     YES
```

Lời giải:

4. Mẫu thử số 4:

```
Submission > 22120412 > src > Input > ≡ input_4.txt
1      -D
2      3
3      A OR B
4      -B OR C
5      D OR -A OR -C
```

Đề bài:

```
Submission > 22120412 > src > Output > ≡ output_4.txt
1      3
2      A OR C
3      B OR -C OR D
4      -A OR -B OR D
5      3
6      A OR B OR D
7      -B OR C OR D
8      -A OR -C OR D
9      1
10     A OR C OR D
11     Ø
12     NO
```

Lời giải:

5. Mẫu thử số 5:

```
Submission > 22120412 > src > Input > ≡ input_5.txt
1      -C
2      5
3      B OR -A
4      A OR C
5      -B OR C OR D
6      C OR -D
7      -A OR B
```

Đề bài:

```
Submission > 22120412 > src > Output > ≡ output_5.txt
1      3
2      B OR C
3      -A OR C OR D
4      -B OR C
5      2
6      -A OR C
7      C OR D
8      Ø
9      NO
```

Lời giải:

6. Mẫu thử số 6:

```
Submission > 22120412 > src > Input > ≡ input_6.txt
1      -D
2      6
3      B OR -C OR D
4      A OR -B
5      C OR -D
6      C OR A OR B
7      A OR C OR B
8      A OR B OR D
```

Đề bài:

```
Submission > 22120412 > src > Output > ≡ output_6.txt
1      7
2      A OR -C OR D
3      B
4      A OR B OR D
5      A OR C
6      A OR D
7      A OR B OR C
8      C
9      2
10     B OR D
11     A
12     1
13     B OR C
14     Ø
15     NO
```

Lời giải:

7. Mẫu thử số 7:

```
Submission > 22120412 > src > Input > ≡ input_7.txt
1  -A OR B
2  3
3  A OR B OR -C
4  -D OR -A OR C
5  C OR B
```

Đề bài:

```
Submission > 22120412 > src > Output > ≡ output_7.txt
1  5
2  B OR -D
3  A OR B
4  A OR -C
5  C OR -D
6  C
7  4
8  A OR B OR -D
9  B OR C OR -D
10 -D
11 A OR -D
12 Ø
13 NO
```

Lời giải:

8. Mẫu thử số 8:

```
Submission > 22120412 > src > Input > ≡ input_8.txt
1  A OR -C
2  4
3  B OR -A
4  C OR -B
5  A OR B OR -C
6  D OR -A OR B
```

Đề bài:

Submission > 22120412 > src > Output > ≡ output_8.txt

```
1 6
2 -A OR C
3 B OR -C
4 A
5 -A OR C OR D
6 B OR -C OR D
7 A OR B
8 9
9 B
10 {}
11 D
12 A OR C
13 B OR D
14 -A OR B OR D
15 B OR C
16 C OR D
17 B OR C OR D
18 YES
```


Lời giải:

9. Mẫu thử số 9:

Submission > 22120412 > src > Input > ≡ input_9.txt

```
1 B OR -A
2 5
3 A OR -B OR -C
4 C OR D OR -A
5 -B OR -D
6 B OR -C OR A
7 -A OR B OR -D
```

Đề bài:

Submission > 22120412 > src > Output >  output_9.txt

```
1 11
2 -B OR D
3 A OR -C
4 -C OR -D
5 -A OR -B OR C
6 B OR D
7 -A OR B OR C
8 C OR D
9 A OR -C OR -D
10 -A OR -D
11 B OR -C OR -D
12 B OR -D
13 23
14 A OR -C OR D
15 {}
16 A OR -B OR D
17 -B OR -C OR -D
18 D
19 -A
20 -A OR C
21 -A OR B
22 -A OR C OR -D
23 -B OR C
24 -D
25 B
26 A OR B OR D
27 B OR C
28 -B OR -C
29 -A OR -B
30 -C
31 A OR D
32 -A OR -B OR -D
33 B OR -C
34 -A OR B OR -D
35 -B OR -D
36 A OR B OR -C
37 YES
```

Lời giải:

10. Mẫu thử số 10:

```
Submission > 22120412 > src > Input > ≡ input_10.txt
1      D OR -C
2      6
3      -B OR D
4      C OR A
5      -B OR C OR D
6      A OR -D
7      C OR -A
8      C OR -D
```

Đề bài:

```
Submission > 22120412 > src > Output > ≡ output_10.txt
1      5
2      A OR -B
3      -B OR C
4      -B
5      A OR -B OR C
6      C OR -D
7      0
8      NO
```

Lời giải: