

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
Khoa Công nghệ Thông tin



BÁO CÁO ĐỒ ÁN CƠ SỞ TRÍ TUỆ NHÂN TẠO
LAB 02: LOGIC

Sinh viên thực hiện: Võ Văn Nam

Mã số sinh viên: 22120222

Lớp: CQ2022/22

MỤC LỤC

1. THÔNG TIN SINH VIÊN.....	3
2. MỨC ĐỘ HOÀN THÀNH.....	3
3. LÝ THUYẾT	4
a. Thuật toán PL-Resolution	4
b. Mã giả	4
c. Ưu điểm	5
d. Nhược điểm	5
e. Đề xuất giải pháp	5
4. CÁC TESTCASE VÀ CÁC HÀM SỬ DỤNG	6
a. Các testcase	6
b. Các hàm sử dụng	9
5. REFERENCES.....	10

1. THÔNG TIN SINH VIÊN

- Họ và tên: Võ Văn Nam
- Mã số sinh viên: 22120222
- Lớp: CQ2022/22

2. MỨC ĐỘ HOÀN THÀNH

- Điểm tự đánh giá: 10/10
- Các yêu cầu trong phần đánh giá của đồ án:

STT	Description	Ratio	Note
1	Read input data and store in suitable data structure	0.5	Completed
2	Implementation of resolution method	1.0	Completed
3	Inference process and results	2.5	Completed
4	Testcases, report, evaluations	1.0	Completed
Total		10	

- Các yêu cầu trong mã nguồn:

STT	Yêu cầu	Ghi chú
1	Have at least 5 test cases	Hoàn thành
2	Remove redundant clause (same clauses in same loop, or the loops before)	Hoàn thành
3	Read the input data and store it in a suitable data structure	Hoàn thành
4	Call PL-Resolution function for execute resolution algorithm	Hoàn thành
5	Export the output data with the required format	Hoàn thành
6	Store the semantic information of true and false value in PL-RESOLUTION and negate alpha	Hoàn thành

7	Literals in a clause is sorted by alphabetical ordering	Hoàn thành
8	Infered condition is checked in each loop as when all the new clauses are generated from the current KB	Hoàn thành
9	Clauses that have $A \vee B \vee \neg B$ format and True value are similar to $A \vee \text{True}$ can be removed	Hoàn thành

3. LÝ THUYẾT

a. Thuật toán PL-Resolution

- PL-Resolution (Propositional Logic Resolution) là một quy tắc suy luận dùng để suy luận trong logic mệnh đề, được sử dụng trong trí tuệ nhân tạo (AI).
- Thuật toán này được sử dụng để kiểm tra xem một query α có thể suy ra từ một cơ sở tri thức KnowledgeBase hay không.
- Thuật toán sử dụng phương pháp chứng minh bằng mâu thuẫn (proof-by-contradiction).
- Các khái niệm:
 - Chứng minh bằng mâu thuẫn: Để chứng minh rằng một cơ sở tri thức KB suy dẫn α , chúng ta thêm phủ định $\neg\alpha$ vào cơ sở tri thức. Ta chứng minh $KB \wedge \neg\alpha$ dẫn đến một mâu thuẫn. Nếu có mâu thuẫn thì $KB \models \alpha$.
 - Chuyển đổi sang CNF (Conjunctive Normal Form): Để có thể hợp giải thì cần chuyển các mệnh đề thành dạng CNF.
 - Quy tắc hợp giải: Tạo ra các mệnh đề mới bằng cách loại bỏ các complementary literals (ví dụ: P và $\neg P$). Nếu một mệnh đề có P và một mệnh đề có $\neg P$, ta có thể được kết hợp chúng, loại bỏ P và $\neg P$, tạo mệnh đề mới. Lặp lại quá trình cho đến khi tìm thấy mâu thuẫn hoặc không thể tạo ra các literal mới.

b. Mã giả

- Mã giả của thuật toán PL-Resolution:

```

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

```

Hình 1. Mã giả của thuật toán. Nguồn: [GeeksforGeeks](https://www.geeksforgeeks.org/resolution-in-propositional-logic/)

c. Ưu điểm

- Thuật toán đơn giản, không khó hiểu, dễ dàng trong việc triển khai và kiểm tra vì thuật toán hoạt động bằng cách loại bỏ các complementay literals.
- Với một câu α bất kỳ, nếu nó có thể được suy dẫn từ cơ sở tri thức KB, ta có thể dùng luật hợp giải để kiểm tra $KB \models \alpha$.

d. Nhược điểm

- Thuật toán không đảm bảo tìm được tất cả các câu được suy dẫn từ cơ sở tri thức KB.
- Thuật toán có thể gây tốn kém về mặt tính toán do không gian tìm kiếm lớn.
- Việc duyệt hết các mệnh đề trong KB mất nhiều thời gian và không khả thi.

e. Đề xuất giải pháp

- Sử dụng phương pháp Davis Putnam. Với mỗi mệnh đề sơ cấp mà có cặp mệnh đề phức hợp để hợp giải:
 - Hợp giải tất cả cặp các mệnh đề phức hợp có thể có, dừng khi gặp mâu thuẫn $\rightarrow KB \models \alpha$.
 - Thêm các mệnh đề phức hợp kết quả, bỏ các mệnh đề phức hợp chứa biến mệnh đề này.
 - Nếu không xuất hiện cặp mệnh đề mâu thuẫn: $KB \models \alpha$.

- Sử dụng kỹ thuật heuristics (ví dụ như lan truyền đơn vị) để giảm bớt các mệnh đề ngay từ ban đầu. Trước khi hợp giải, có thể dùng một heuristic ưu tiên các mệnh đề dễ dàng dẫn đến mâu thuẫn, giúp giảm bớt các bước không cần thiết.
- Kết hợp hợp giải với các SAT solvers hiện đại, các kỹ thuật tối ưu hóa như CDCL - Conflict-Driven Clause Learning. Phương pháp này học từ các xung đột và lưu trữ các mệnh đề đã học được, giúp giảm việc kiểm tra lại các mệnh đề và tăng tốc quá trình suy luận.
- Song song hóa xử lý thuật toán trên nhiều bộ xử lý hoặc hệ thống tính toán để giảm sự tồn kém về mặt tính toán của thuật toán.

4. CÁC TESTCASE VÀ CÁC HÀM SỬ DỤNG

a. Các testcase

- Chương trình bao gồm 7 testcase, trong đó 2 testcase input01.txt và input02.txt là testcase mẫu mà yêu cầu đồ án cung cấp, các testcase từ input03.txt đến input07.txt là testcase bên ngoài.
- Testcase 01:

input01.txt	output01.txt	Ghi chú
-A	3	
4	-A	(-A OR B) hợp giải với (-B)
-A OR B	B	(-A OR B) hợp giải với (A)
B OR -C	-C	(B OR -C) hợp giải với (-B)
A OR -B OR C	4	
-B	-B OR C	(A OR -B OR C) hợp giải với (-A)
	A OR C	(A OR -B OR C) hợp giải với (B)
	A OR -B	(A OR -B OR C) hợp giải với (-C)
	{ }	(-B) hợp giải với B
	YES	KB $\models \alpha$ vì tồn tại mệnh đề α trong KB

- Testcase 02:

input02.txt	output02.txt	Ghi chú
A	2	
4	-C	(B OR -C) hợp giải với (-B)
-A OR B	-B OR C	(A OR -B OR C) hợp giải với (-A)

B OR -C	2	
A OR -B OR C	-A OR C	(-A OR B) hợp giải với (-B OR C)
-B	A OR -B	(A OR -B OR C) hợp giải với (-C)
	1	
	A OR -C	(B OR -C) hợp giải với (A OR -B)
	0	
	NO	KB không suy dẫn ra α vì không phát sinh được mệnh đề mới và không tìm thấy mệnh đề rỗng

- Testcase 03:

input03.txt	output03.txt	Ghi chú
-A	3	
4	-A OR -C	(B OR -C) hợp giải với (-A OR -B)
A OR B	-B	(-A OR -B) hợp giải với (A)
B OR -C	-B OR C	(C OR -B OR -A) hợp giải với (A)
-A OR -B	2	
C OR -B OR -A	A OR C	(A OR B) hợp giải với (-B OR C)
	-C	(B OR -C) hợp giải với (-B)
	0	
	NO	KB không suy dẫn ra α vì không phát sinh được mệnh đề mới và không tìm thấy mệnh đề rỗng

- Testcase 04:

input04.txt	output04.txt	Ghi chú
-B	5	
4	A OR C	(A OR B) hợp giải với (-B OR C)
A OR B	A OR -B	(-B OR C) hợp giải với (A OR -C)
-B OR C	C	(-B OR C) hợp giải với (B)
-A OR -B	-B OR -C	(-A OR -B) hợp giải với (A OR -C)

A OR -C	-A	(-A OR -B) hợp giải với (B)
	3	
	A	(A OR B) hợp giải với (A OR -B)
	-B	(-B OR C) hợp giải với (-B OR -C)
	-C	(A OR -C) hợp giải với (-A)
	1	
	{ }	(B) hợp giải với (-B)
	YES	KB $\models \alpha$ vì tồn tại mệnh đề rỗng trong KB

- Testcase 05:

input05.txt	output05.txt	Ghi chú
D	4	
5	B OR -D	(-D OR A) hợp giải với (-A OR B)
-D OR A	C OR -D OR E	(-D OR A) hợp giải với (-A OR C OR E)
-A OR B	-A OR C	(-A OR B) hợp giải với (-B OR C)
-B OR C	B OR -E	(-A OR B) hợp giải với (A OR -E)
A OR -E	5	
-A OR C OR E	C OR -D	(-D OR A) hợp giải với (-A OR C)
	C OR -E	(-B OR C) hợp giải với (B OR -E)
	A OR C OR -D	(A OR -E) hợp giải với (C OR -D OR E)
	-A OR B OR C	(-A OR C OR E) hợp giải với (B OR -E)
	B OR C OR -D	(C OR -D OR E) hợp giải (B OR -E)
	1	
	B OR C OR -E	(A OR -E) OR (-A OR B OR C)
	0	
	NO	KB không suy dẫn ra α vì không phát sinh được mệnh đề mới và không tìm thấy mệnh đề rỗng

- Testcase 06:

input06.txt	output06.txt	Ghi chú
T	4	
3	T OR Y	(T OR X) hợp giải với (-X OR Y)
T OR X	X	(T OR X) hợp giải với (-T)
-Y OR T	T OR -X	(-Y OR T) hợp giải với (-X OR Y)
-X OR Y	-Y	(-Y OR T) hợp giải với (-T)
	3	
	T	(T OR X) hợp giải với (T OR -X)
	Y	(-X OR Y) hợp giải với (X)
	-X	(-X OR Y) hợp giải với (-Y)
	1	
	{}	(-T) hợp giải với (T)
	YES	KB $\models \alpha$ vì tồn tại mệnh đề rỗng trong KB

- Testcase 07:

input07.txt	output07.txt	Ghi chú
K	5	
5	K	(K OR L) hợp giải với (-L OR K)
K OR L	L	(K OR L) hợp giải với (-K)
M OR -K	-L OR M	(M OR -K) hợp giải với (-L OR K)
-N OR O	K OR M	(L OR M) hợp giải với (-L OR K)
L OR M	-L	(-L OR K) hợp giải với (-K)
-L OR K	2	
	M	(M OR -K) hợp giải với (K)
	{}	(-K) hợp giải với K
	YES	KB $\models \alpha$ vì tồn tại mệnh đề rỗng trong KB

b. Các hàm sử dụng

- def read_file(filename): Hàm đọc dữ liệu từ file input, trả về alpha (lưu các literal trong một list) và KB (một list các mệnh đề, mỗi mệnh đề là một list các literal).

- `def write_file(result, isTrue, filename):` Hàm ghi kết quả ra file output. Hàm thực hiện ghi số mệnh đề tạo ra sau mỗi vòng lặp, các mệnh đề trong vòng lặp đó và kết quả hợp giải.
- `def PL_resolution(KB, alpha):` Hàm thực hiện thuật toán PL-Resolution.
- Các hàm hỗ trợ khác được comment trong mã nguồn.

5. REFERENCES

- Slide bài giảng môn học Cơ sở trí tuệ nhân tạo, Thầy Nguyễn Ngọc Đức
<https://cloud.ducnn.com/index.php/s/cqCO2iiQt1Sm7Xv#pdfviewer>
- GeeksforGeeks - Resolution Algorithm in Artificial Intelligence
<https://www.geeksforgeeks.org/resolution-algorithm-in-artificial-intelligence/>
- Stack Overflow - How does a Resolution algorithm work for propositional logic?
<https://stackoverflow.com/questions/12460818/how-does-a-resolution-algorithm-work-for-propositional-logic>