

Lab 02: Logic

Introduction to Artificial Intelligence

Giáo viên hướng dẫn: Nguyen Tran Duy Minh
Nguyen Ngoc Duc

REPORT

Sinh viên: Ngô Thanh Lực

Mã số sinh viên: 20120325

MỤC LỤC

1. Tự đánh giá	3
2. Cài đặt.....	3
3. Testcases.....	4
3.1. Testcase 01	4
3.2. Testcase 02	4
3.3. Testcase 03	5
3.4. Testcase 04	5
3.5. Testcase 05	5
4. Nhận xét và giải pháp cải tiến.....	6
4.1. Nhận xét.....	6
4.1.1. Ưu điểm.....	6
4.1.2. Nhược điểm.....	6
4.1.3. Giải pháp cải tiến	6

1. Tự đánh giá

Tiêu chí	Mức độ hoàn thiện
Đọc dữ liệu đầu vào và lưu trữ trong cấu trúc dữ liệu phù hợp	100%
Cài đặt và thực hiện thuật giải hợp giải	100%
Quá trình suy luận và kết quả	100%
Báo cáo, test case và đánh giá	85%

2. Cài đặt

- Sử dụng Python để cài đặt **thuật toán PL-Resolution**, được sử dụng để xác định xem một **knowledge base** có suy ra được một logic mệnh đề hay không (**alpha**).
- Biến flag** được sử dụng để theo dõi xem mệnh đề rỗng có được tạo ra trong quá trình giải quyết hay không. Nếu mệnh đề rỗng được tạo ra, thì nghĩa là KB suy ra được câu cho trước.
- Hàm read_input()** đọc tệp đầu vào và trả về câu alpha và cơ sở tri thức (KB).
- Hàm parse_clause ()** phân tích một mệnh đề và trả về một danh sách được sắp xếp các literals.
- Hàm to_cnf()** chuyển đổi một danh sách các mệnh đề thành dạng chuẩn tắc(CNF).
- Hàm negate_alpha()** sẽ phủ định câu alpha hoặc literal bằng cách thêm một ký hiệu phủ định trước nó.
- Hàm resolution(clause1,clause2)** nhận hai mệnh đề làm đầu vào và trả về một mệnh đề kết quả. Hàm này kiểm tra xem có tồn tại các nguyên tử phủ định nhau trong hai mệnh đề hay không và tạo ra một mệnh đề kết quả mới bằng cách loại bỏ các nguyên tử này.
- Hàm pl_resolution(alpha, kb)** thực hiện thuật toán PL Resolution bằng cách lặp lại qua tất cả các cặp mệnh đề trong KB và giải quyết chúng cho đến khi hoặc mệnh đề rỗng được tạo ra hoặc không có mệnh đề mới nào. Nó trả về một danh sách các mệnh đề lịch sử và một giá trị boolean cho biết KB có suy ra được câu alpha hay không.
 - alpha: mệnh đề cần kiểm tra tính tương đương với tập kb.
 - kb: tập các mệnh đề, cùng với alpha, được đưa vào để kiểm tra tính tương đương.
 - Đầu tiên, dùng **negate_alpha** đảo ngược mệnh đề alpha. Sau đó, hàm to_cnf() được sử dụng để chuyển tập các mệnh đề trong kb về dạng chuẩn tắc hợp (CNF).
 - Trong vòng lặp while vô hạn, lặp qua mọi cặp mệnh đề trong kb và sử dụng resolution để tạo ra các mệnh đề mới. Nếu tạo ra một mệnh đề trống (**{}**) trong quá trình resolution, nghĩa là tập kb không thể chứa đồng thời hai mệnh đề đối lập, tức là chứa cả một mệnh

đề và phủ định của nó. Lúc này, hàm `pl_resolution` sẽ trả về các resolvents đã được ghi lại trước đó và giá trị `True`. **Tức là KB entails α vì tồn tại mệnh đề rỗng trong KB**

- Nếu tạo ra một mệnh đề mới, hàm kiểm tra xem mệnh đề đó đã có trong tập kb chưa và chưa được tạo ra từ các mệnh đề mới. Nếu không có, thì hàm thêm mệnh đề mới vào tập kb. Đồng thời, hàm lưu các mệnh đề mới vào biến `new_resolvents` để sử dụng sau này.
- Sau khi kết thúc chu kì, nếu biến `new_resolvents` không chứa mệnh đề mới, nghĩa là không thể tạo ra thêm mệnh đề mới từ tập kb. Khi đó, hàm trả về một tuple chứa list các mệnh đề mới và giá trị boolean `False`. **Tức là KB KHÔNG entail α vì không phát sinh được mệnh đề mới và không tìm thấy mệnh đề rỗng.**
- Nếu biến `new_resolvents` chứa mệnh đề mới thì ta sẽ thêm các mệnh đề mới vào kb. `Resolvents_history` theo dõi tất cả các mệnh đề đã được tạo ra ở mỗi lần lặp của vòng lặp.
- Hàm `export_output()` xuất ra file output danh sách các resolvent được tạo ra trong quá trình tìm kiếm và một dựa vào `entails_alpha` để xác định liệu KB có suy luận được alpha hay không.

3. Testcases

3.1. Testcase 01

INPUT1.TXT	OUTPUT1.TXT	Ghi chú
B	6	
7	-A OR B	'-A OR B OR C hợp giải -A OR B OR -C'
-A OR B OR C	-A OR C	'-A OR B OR C hợp giải -A OR -B OR C'
-A OR -B OR -C	-A OR -C	'-A OR -B OR -C hợp giải -A OR B OR -C'
B OR C	B	'B OR C hợp giải B OR -C'
-A OR B OR -C	C	'B OR C hợp giải -B'
-A OR -B OR C	-C	'B OR -C hợp giải -B'
B OR -C	2	
-A OR -B	-A	'-A OR -B hợp giải -A OR B'
	{}	C hợp giải với -C
	YES	

3.2. Testcase 02

INPUT2.TXT	OUTPUT2.TXT	Ghi chú
-A	2	
6	-B	'-A OR -B hợp giải A'
A OR B OR -C	C	'-A OR C hợp giải A'
-B OR C	2	
-A OR -B OR C	A OR -C	'A OR B OR -C hợp giải -B'
A OR B OR -C	A OR B	'A OR B OR -C hợp giải C'
-A OR -B	4	

C OR -A	A OR -B	'-B OR C hợp giải A OR -C'
	A OR C	'-B OR C hợp giải A OR B'
	-B OR -C	'-A OR -B hợp giải A OR -C'
	B OR C	'-A OR C hợp giải A OR B'
	0	
	NO	

3.3. Testcase 03

INPUT2.TXT	OUTPUT2.TXT	Ghi chú
C	8	
7	-A OR C	-A OR B OR C hợp giải -B OR C
-A OR B OR C	B OR C	-A OR B OR C hợp giải A OR B OR C
A OR -C	-A OR B	-A OR B OR C hợp giải -C
-B OR C	A OR -B	A OR -C hợp giải -B OR C
-A OR -B OR -C	A OR B	'A OR -C hợp giải A OR B OR C
-B OR -C	-A OR -B	-B OR C hợp giải -A OR -B OR -C
A OR B OR -C	-B	-B OR C hợp giải -B OR -C'
A OR B OR C	A OR C	-B OR C hợp giải A OR B OR C'
	6	
	B OR -C	'A OR -C hợp giải -A OR B'
	A	'A OR -C hợp giải A OR C'
	C	'-B OR C hợp giải B OR C'
	-A OR -C	'-A OR -B OR -C hợp giải -A OR B'
	-A	'-C hợp giải -A OR C'
	B	'-C hợp giải B OR C'
	1	
	{}	-B hợp giải B
	YES	

3.4. Testcase 04

INPUT2.TXT	OUTPUT2.TXT	Ghi chú
B	3	
5	-A OR C	'-A OR B hợp giải -B OR C'
-A OR B	-A	'-A OR B hợp giải -B'
B OR -C	-C	'B OR -C hợp giải -B'
A OR -B OR C	1	
-B OR C	A OR -B	'A OR -B OR C hợp giải -C'
B OR -A	1	
	A OR -C	'B OR -C hợp giải A OR -B'
	0	
	NO	

3.5. Testcase 05

INPUT2.TXT	OUTPUT2.TXT	Ghi chú
-C	6	
7	-A OR C	'-A OR B OR C hợp giải -B OR C'
-A OR B OR C	A OR -B	'A OR -C hợp giải -B OR C'
A OR -C	A	'A OR -C hợp giải C'

-B OR C	-A OR -B	'-B OR C hợp giải -A OR -B OR -C'
-A OR -B OR -C	-B	'-B OR C hợp giải -B OR -C'
-B OR -C	A OR B	'A OR B OR -C hợp giải C'
A OR B OR -C	2	
-A OR -B OR C	B OR C	'-A OR B OR C hợp giải A'
	A OR C	'-B OR C hợp giải A OR B'
	0	
	NO	

4. Nhận xét và giải pháp cải tiến

4.1. Nhận xét

4.1.1. Ưu điểm

- PI-resolution là một phương pháp suy luận logic tuyến tính dựa trên resolution, có thể kiểm tra tính nhất quán và tính suy diễn của các câu lệnh logic mệnh đề.
- PI-resolution có thể xử lý các bài toán logic có chứa các biểu thức phức tạp, bằng cách chuyển đổi chúng về dạng chuẩn tắc (CNF) và áp dụng resolution để tạo ra các mệnh đề mới.
- PI-resolution có thể loại bỏ các mệnh đề vô nghĩa (tautology) trong quá trình suy luận, nhờ đó giảm thiểu số lượng mệnh đề cần xét và tăng hiệu quả của thuật toán.
- PI-resolution có thể được cài đặt bằng nhiều ngôn ngữ lập trình khác nhau, ví dụ như SQL, một ngôn ngữ lập trình hướng thủ tục được cung cấp bởi Oracle, có nhiều tính năng và ưu điểm như tích hợp, tính di động và bảo mật cao.

4.1.2. Nhược điểm

- Phải chuyển đổi các mệnh đề về dạng chuẩn tắc (CNF) trước khi áp dụng quy tắc resolution, điều này có thể làm tăng số lượng các mệnh đề và phức tạp hơn.
- Phải duyệt qua tất cả các cặp mệnh đề trong tập hợp các mệnh đề để tìm kiếm các biểu thức phủ định, điều này có thể làm tốn nhiều thời gian và bộ nhớ.
- Có thể sinh ra nhiều mệnh đề mới không liên quan đến mục tiêu hoặc không có ý nghĩa, điều này gây ra sự lãng phí và khó kiểm soát.
- Không thể áp dụng quy tắc resolution nếu hai mệnh đề có cùng biểu thức không phủ định, điều này hạn chế khả năng suy luận.
- Có thể gặp phải vòng lặp vô hạn nếu tập hợp các mệnh đề mới là tập con của tập hợp các mệnh đề cũ, điều này dẫn đến sự bế tắc.

4.1.3. Giải pháp cải tiến

- Sử dụng các kỹ thuật cắt tỉa để loại bỏ những mệnh đề không cần thiết hoặc trùng lặp.
- Sử dụng các kỹ thuật sắp xếp để chọn ra những mệnh đề có khả năng giải quyết cao nhất.
- Sử dụng các kỹ thuật biểu diễn hiệu quả để lưu trữ và xử lý các mệnh đề.