

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 THỰC HÀNH

**MÔN: CƠ SỞ TRÍ TUỆ NHÂN TẠO
(CSC14003 – 21CLC01)**

LAB02: LOGIC

GIẢNG VIÊN LÝ THUYẾT

BÙI TIẾN LÊN

GIẢNG VIÊN THỰC HÀNH

NGUYỄN NGỌC ĐỨC

SINH VIÊN THỰC HIỆN

21127621 – ÂU DƯƠNG KHANG

MỤC LỤC

1) Cài đặt

2) Danh sách testcase

3) Đánh giá

1) Cài đặt

- class Clause: đại diện cho mệnh đề. Bao gồm các phương thức và thuộc tính sau:
 - + literal_list: danh sách các literal của mệnh đề.
 - + count: số lượng literal của mệnh đề. Riêng đối với mệnh đề rỗng ({}) thì count là -1.
 - + def negative: phương thức trả về mệnh đề phủ định (danh sách mệnh đề dạng hội chuẩn CNF) của mệnh đề gốc.
 - + def __str__: override lại phương thức __str__ để phục vụ cho việc xuất ra file output.
 - + def __eq__: override lại phương thức __eq__. Khi so sánh bằng hai Clause thì sẽ trả về kết quả so sánh của 2 list literal_list.
- def NegativeLiteral: hàm trả về phủ định của một literal.
- def SortKey: hàm trả về ký tự cuối cùng của literal để hỗ trợ cho việc sắp xếp các literal trong mệnh đề khi sử dụng hàm sort của hệ thống.
- def CanResolve: hàm trả về True nếu 2 mệnh đề đang xét có thể hợp giải (chỉ khi có 1 cặp literal đối ngẫu), ngược lại, trả về False.
- def Resolve: hàm hợp giải trên hai mệnh đề c1 (Clause 1) và c2 (Clause 2). Khi hợp giải mệnh đề, từng cặp literal của c1 và c2 sẽ được kiểm tra xem có phải đối ngẫu của nhau hay không. Nếu là đối ngẫu, thì sẽ loại bỏ 2 literal đối ngẫu tương ứng trong c1 và c2, đồng thời tạo ra danh sách chứa các literal còn lại của 2 Clause, gán check = 1 và thoát khỏi vòng lặp. Sau đó, nếu check bằng 1 thì kiểm tra nếu danh sách các literal còn lại là rỗng thì return về mệnh đề rỗng, ngược lại sẽ return về Clause mới với literal_list chính là danh sách các literal còn lại.
- def PL_Resolution: hàm thực thi giải thuật hợp giải. Trong hàm này sẽ thực hiện đồng thời hai việc là hợp giải và xuất output. Đầu tiên, tiến hành phủ định câu alpha, và thêm vào KB có được từ input để tạo thành KB mới. Thực hiện vòng lặp True, đối với mỗi vòng lặp tiến hành xét từng cặp mệnh đề (Ci, Cj). Nếu cặp mệnh đề (Ci, Cj) là có thể hợp giải thì tiến hành hợp giải. Sau khi hợp giải ra kết quả, Clause mới sẽ được kiểm tra, nếu nó không tồn tại trong KB, không tồn tại trong danh sách Clause mới và có count khác 0 thì sẽ được thêm vào danh sách Clause mới. Khi kết thúc vòng lặp với KB hiện hành, ghi vào file output số mệnh đề mới được phát sinh, và các mệnh đề trong danh sách Clause mới, đồng thời kiểm tra xem có mệnh đề rỗng trong danh sách này hay không, nếu có thì gán solve bằng 1. Nếu không phát sinh được mệnh đề nào mới thì ghi vào file output NO và return False, nếu solve bằng 1 thì ghi vào file output YES và return True, ngược lại lưu lại số mệnh đề có trong KB hiện hành, đồng thời thêm các mệnh đề mới vào KB tạo thành KB mới. Từ vòng lặp thứ hai, kiểm tra các Clause mới với nhau, sau đó sẽ kiểm tra các Clause mới này với Clause cũ đã có trong

KB của vòng lặp trước và bỏ qua việc kiểm tra các Clause cũ đã có trong KB trước đó với nhau.

- def Input: hàm đọc dữ liệu đầu vào.
- def main: hàm thực thi flow chính. Tiến hành lấy danh sách các file trong thư mục INPUT, sau đó với từng file input, tiến hành đọc dữ liệu đầu vào, sau đó gọi hàm PL_Resolution để hợp giải và ghi dữ liệu đầu ra vào file output tương ứng.

2) Danh sách testcase

input0.txt	output0.txt	Ghi chú
-A	3	
4	-A	(-A OR B) hợp giải với -B
-A OR B	B	(-A OR B) hợp giải với (negative -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 hợp giải với (negative -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
	YES	

input1.txt	output1.txt	Ghi chú
A	2	
4	-C	(-C OR B) hợp giải với -B
-A OR B	-B OR C	(A OR C OR -B) hợp giải với (negative A)
-C OR B	2	
A OR C OR -B	A OR -B	(A OR C OR -B) hợp giải với (-C)
-B	-A OR C	(-A OR B) hợp giải với (-B OR C)
	1	
	A OR -C	(A OR -B) hợp giải với (-C OR B)
	0	
	NO	

input2.txt	output2.txt	Ghi chú
A	3	
4	A	(-B OR A) hợp giải với B
-A OR B OR C	-B	(-B OR A) hợp giải với (negative A)
-B OR A	-C	(-C OR A) hợp giải với (negative A)
-C OR A	4	
B	B OR C	(-A OR B OR C) hợp giải với A
	{}	B hợp giải với -B
	-A OR C	(-A OR B OR C) hợp giải với -B
	-A OR B	(-A OR B OR C) hợp giải với -C
	YES	

input3.txt	output3.txt	Ghi chú
-C	4	
3	-A OR -C	(-A OR B) hợp giải với (-B OR -C)
-A OR B	B OR -C	(-A OR B) hợp giải với (A OR -C)
-B OR -C	-B	(-B OR -C) hợp giải với (negative -C)
A OR -C	A	(A OR -C) hợp giải với (negative -C)
	3	
	-C	(-A OR -C) hợp giải với A
	-A	(-A OR -C) hợp giải với (negative -C)
	B	(B OR -C) hợp giải với (negative -C)
	1	
	{}	-B hợp giải với B
	YES	

input4.txt	output4.txt	Ghi chú
-B	1	
3	A	(A OR -B) hợp giải với (negative -B)
-A OR B OR C	1	
A OR -B	B OR C	(-A OR B OR C) hợp giải với A
A OR -C	2	
	A OR C	(A OR -B) hợp giải với (B OR C)
	A OR B	(A OR -C) hợp giải với (B OR C)
	0	
	NO	

3) Đánh giá

- Ưu điểm: Giải thuật cho ra kết quả, phát sinh được hết tất cả các mệnh đề sinh ra trong quá trình hợp giải.
- Nhược điểm: Đối với các trường hợp mà có kết luận KB entails α thì không nhất thiết phải phát sinh hết tất cả các mệnh đề để tìm ra kết quả. Đối với bài toán có input phức tạp, nhiều literal đơn lẻ thì thuật toán có thể xử lý chậm.