



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



CƠ SỞ TRÍ TUỆ NHÂN TẠO

BÁO CÁO ĐỒ ÁN THỰC HÀNH PROJECT 2

Biểu diễn tri thức bằng logic

LỚP CỬ NHÂN TÀI NĂNG

Nhóm thực hiện:

1712152 - Nguyễn Thị Mai Thanh

1712856 - Huỳnh Văn Tú

1712858 - Nguyễn Ngọc Tú

MỤC LỤC

BÀI 1: HỢP GIẢI TRÊN LOGIC MỆNH ĐỀ	1
I. Chi tiết thuật toán hợp giải trên các logic mệnh đề PL-Resolution:	1
II. Cấu trúc chương trình:	2
1. Cấu trúc tập tin nhập và tập tin xuất:	2
2. Cấu trúc tổ chức dữ liệu:	2
III. Báo cáo test case:	3
IV. Nhận xét và đề xuất giải pháp cải tiến của giải thuật hợp giải:	7
1. Nhận xét về hợp giải:	7
2. Đề xuất giải pháp cải tiến:	8
V. Đánh giá mức độ hoàn thành của nhóm:	8
BÀI 2: BIỂU DIỄN TRI THỨC BẰNG LOGIC BẬC NHẤT VỚI PROLOG	9
I. Cây phả hệ gia đình Hoàng gia Anh:	9
1. Xây dựng cơ sở tri thức mô tả các quan hệ:	9
2. Định nghĩa các vị từ sau dựa vào những vị từ đã có ở trên:	9
3. Các câu hỏi xây dựng:	9
4. Ví dụ chạy chương trình:	10
II. Quản lý trường đại học:	11
1. Xây dựng các quan hệ trong trường học:	12
2. Xây dựng mô hình:	12
3. Các câu hỏi xây dựng:	14
BÀI 3: SUY DIỄN TRÊN LOGIC BẬC NHẤT	16
I. Một số thay đổi của hệ thống tự cài đặt so với Prolog:	16
II. Kiểm chứng kết quả của hệ thống tự cài đặt so với Prolog	17
1. Hoàng gia Anh	17
2. Trường đại học	20

BÀI 1: HỢP GIẢI TRÊN LOGIC MỆNH ĐỀ

I. Chi tiết thuật toán hợp giải trên các logic mệnh đề PL-Resolution:

- **Định lý:** (Phản chứng)
Với mọi **KB** và câu **α** ta có $KB \models \alpha \iff KB \cup \{\alpha\}$ là **không** được thỏa mãn
- **Cơ sở của hợp giải:**
 - Ở mỗi lần lặp, hai mệnh đề gọi là mệnh đề cha được so sánh (hay giải quyết – resolved) để tạo ra mệnh đề mới. Giả sử trong hệ thống có hai mệnh đề:

$$A \vee B \text{ và } \neg A \vee C \text{ có thể dẫn xuất thành } B \vee C$$

- Quá trình áp dụng luật suy diễn hợp giải **dừng lại** khi:
 - + **Không** có biểu thức mới nào được sinh ra nữa
 - + Sinh ra câu **rỗng** $\{\}$, tức là có mâu thuẫn (contradiction) xảy ra: Sau khi hợp giải, thu được biểu thức rỗng $p \wedge \neg p \Rightarrow \{\}$
- **Giải thuật hợp giải cho Logic mệnh đề:**

```
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  $C_i, C_j$  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
     $clauses \leftarrow clauses \cup new$ 
```

Đầu vào: Cơ sở tri thức (một tập các mệnh đề) **KB** và một mệnh đề **α** cần chứng minh

Đầu ra: Chứng minh là khẳng định $KB \models \alpha$ đúng hay sai

Giải thuật:

Bước 1: Lấy phủ định của mệnh đề α

Bước 2: Thêm mệnh đề phủ định của α vào tập các mệnh đề đã cho

Bước 3: Thực hiện vòng lặp cho đến khi tìm thấy sự mâu thuẫn hoặc không thể tiếp tục:

B3.1: Chọn hai mệnh đề. Gọi là các mệnh đề cha

+ Hợp giải các mệnh đề đã nhóm. Mệnh đề kết quả là tuyển của tất cả các biến mệnh đề trong các mệnh đề cha trừ: nếu có bất kỳ các cặp biến mệnh đề A và $\neg A$ và một nằm trong mệnh đề cha này, một nằm trong mệnh đề cha còn lại \Rightarrow chọn một cặp và xóa cả A và $\neg A$ ra khỏi mệnh đề kết quả.

+ Nếu mệnh đề kết quả là rỗng, thì xem như đã tìm được sự mâu thuẫn. Nếu không thì thêm mệnh đề kết quả đó vào trong tập hợp các mệnh đề hiện có.

B3.2: Nếu tất cả các mệnh đề mới được tạo ra không có mệnh đề mới thì kết thúc vòng lặp và trả về false.

II. Cấu trúc chương trình:

1. Cấu trúc tập tin nhập và tập tin xuất:

1.1. Dữ liệu đầu vào: Lưu dữ liệu trong tập tin **input.txt**

Tập tin có định dạng như sau:

- + Dòng đầu tiên chứa mệnh đề **alpha** cần chứng minh
- + Dòng thứ hai chứa số nguyên **N** – số mệnh đề cơ sở trong KB
- + N dòng tiếp theo biểu diễn các mệnh đề trong KB, mỗi mệnh đề được ghi trên một dòng. Với literal Dương được biểu diễn bằng ký tự đơn viết hoa (A-Z), literal Âm là literal Dương có dấu ‘-’ ngay trước ký tự. Từ khóa ‘OR’ nối các literal với nhau, có thể có một hoặc nhiều khoảng trắng giữa các literal và từ khóa

1.2. Dữ liệu đầu ra: Lưu dữ liệu trong tập tin output.txt

Tập tin có định dạng như sau:

- + Dòng đầu tiên chứa số nguyên M_1 – số mệnh đề mới được phát sinh trong vòng lặp đầu tiên. M_1 dòng tiếp theo biểu diễn các mệnh đề được phát sinh (kể cả mệnh đề rỗng ‘{}’), mỗi mệnh đề được biểu diễn trên một dòng
- + Các vòng lặp tiếp theo (lần lượt có M_2, \dots, M_n mệnh đề) được biểu diễn tương tự như trên.
- + Dòng cuối cùng trình bày câu kết luận, tức là trả lời câu hỏi “**KB entails \square ?**”. In YES nếu KB entails \square . Ngược lại, in NO

• Lưu ý:

- Các literal trong cùng mệnh đề (đối với cả dữ liệu đầu vào và đầu ra) được xếp theo thứ tự chữ cái
- Kiểm tra điều kiện suy dẫn ở cuối mỗi vòng lặp, tức là khi đã phát sinh hết câu mới từ KB hiện hành, chứ không kiểm tra sau mỗi lần phát sinh một câu
- Bỏ qua các mệnh đề trùng (xuất hiện trong cùng vòng lặp hay, KB ban đầu hay những vòng lặp trước đó).

2. Cấu trúc tổ chức dữ liệu:

2.1. Tổ chức lưu trữ dữ liệu đầu vào:

Đọc dữ liệu từ file Input, sau đó lưu vào Từ điển dữ liệu (Dictionary) có cấu trúc như sau:

- + ‘base knowledge’ chứa danh sách các mệnh đề cơ sở trong KB
- + ‘query’ chứa câu mệnh đề alpha cần chứng minh
- + ‘number of clauses’ chứa số nguyên – số các mệnh đề cơ sở đã cho

Ví dụ: Với file Input như sau:

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

- Cấu trúc dữ liệu đầu vào sẽ là:

```
{'base knowledge': ['A OR C', '-C OR D', 'A OR B', '-B OR C'], 'query': 'A OR -B OR D', 'number of clauses': 4}
```

2.2. Tổ chức lưu trữ dữ liệu đầu ra:

- Các dữ liệu được lưu trữ trong hai danh sách:

- + **number_clauses_step**: chứa các số nguyên, với mỗi số nguyên là số lượng mệnh đề mới được phát sinh ở mỗi vòng lặp
- + **list_clauses_step**: chứa danh sách các mệnh đề được suy diễn từ vòng lặp đầu tiên cho đến khi kết thúc vòng lặp. Từ đó có thể tính vị trí bắt đầu và vị trí kết thúc của các mệnh đề ở mỗi vòng lặp là:

$$\forall i \in (1, \text{len}(\text{number_clauses_step})) : \text{number_clauses_step}[i] = \text{number_clauses_step}[i-1] + \text{number_clauses_step}[i]$$

- Như vậy mệnh đề được phát sinh trong vòng lặp thứ i là các mệnh đề:

list_clauses_step[number_clauses_step[i-1]: number_clauses_step[i]]

- Dùng biến **result** dạng Boolean để lưu kết quả của **KB entails** □

- **Ngoài ra:**

Sử dụng một biến có cấu trúc dữ liệu set để lưu trữ tất cả các mệnh đề của hợp giải => Do đó, có thể dùng biến này để kiểm tra xem mệnh đề phát sinh ra là mới hay đã có. Ngoài ra, các mệnh đề có tính duy nhất giúp dễ dàng hơn trong việc in dữ liệu. Các trường hợp ví dụ

III. Báo cáo test case:

Input.txt	Output.txt	Ghi chú
Test 01		
<pre> A OR -K 5 -A OR B OR -C OR F -C E OR F C OR -D -F OR K </pre>	<pre> 4 -A OR B OR -D OR F -A OR B OR -C OR K -D E OR K 1 -A OR B OR -D OR K 0 NO </pre>	
<pre> A OR -K 5 -A OR B OR -C OR F -C E OR F C OR -D -F OR K </pre>	<pre> 4 -A OR B OR -D OR F -A OR B OR -C OR K -D E OR K 1 -A OR B OR -D OR K 0 NO </pre>	<pre> (-A OR B OR -C OR F) AND (C OR -D) (-A OR B OR -C OR F) AND (-F OR K) (-C) AND (C OR -D) (E OR F) AND (-F OR K) (C OR -D) AND (-A OR B OR -C OR K) </pre>
Test 02		

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

A OR -B OR D 4 A OR C -C OR D A OR B -B OR C	4 A OR D C -B OR D -C 4 A D -B {} YES	(A OR C) AND (-C OR D) (A OR C) AND (-A) (-C OR D) AND (-B OR C) (-C OR D) AND (-D) (A OR C) AND (-C) (-C OR D) AND (C) (-B OR C) AND (-C) (C) AND (-C)
---	---	--

Test 03

	A OR B 5 A OR -C OR -D -A OR -C B OR D E A OR B OR -E	7 -C OR -D A OR B OR -C B OR -C OR -E D A OR B B OR -E A OR -E 7 A OR -C B OR -C -C OR -E A B -E -C 1 {} YES	
--	---	--	--

A OR B 5 A OR -C OR -D -A OR -C B OR D E A OR B OR -E	7 -C OR -D A OR B OR -C B OR -C OR -E D A OR B B OR -E A OR -E	(A OR -C OR -D) AND (-A OR -C) (A OR -C OR -D) AND (B OR D) (-A OR -C) AND (A OR B OR -E) (B OR D) AND (-B) (E) AND (A OR B OR -E) (A OR B OR -E) AND (-A) (A OR B OR -E) AND (-B)
---	---	--

	7 A OR -C B OR -C -C OR -E A B -E -C 1 {} YES	(A OR -C OR -D) AND (D) (-A OR -C) AND (A OR B OR -C) (-A OR -C) AND (A OR -E) (E) AND (A OR -E) (E) AND (B OR -E) (-A) AND (A OR -E) (-C OR -D) AND (D) (E) AND (-E)
--	---	--

Test 04

File đính kèm

A OR B OR -C OR -D 6 A OR -E OR -K B OR -F C OR F -C OR E K -B OR E OR F	8 A OR -C OR -K A OR -E A OR -B OR F OR -K -E OR -K B OR C -F E OR F E 17 A OR -K A OR F OR -K B OR E -C OR -K A OR -C A OR -B OR F -E -B OR E C OR E OR F -B OR F OR -K -K A A OR B OR -K F OR -K A OR F A OR -B OR -K A OR C OR F OR -K 21 B OR -K E OR -F A OR B A OR B OR C OR -K B OR C OR E	(A OR -E OR -K) AND (-C OR E) (A OR -E OR -K) AND (K) (A OR -E OR -K) AND (-B OR E OR F) (A OR -E OR -K) AND (-A) (B OR -F) AND (C OR F) (B OR -F AND -B) (C OR F AND -C OR E) (-C OR E AND C) (A OR -E OR -K) AND (E) (A OR -E OR -K) AND (E OR F) (B OR -F) AND (E OR F) (-C OR E) AND (-E OR -K) (-C OR E) AND (A OR -E) (K) AND (A OR -B OR F OR -K) (K) AND (-E OR -K) (-B OR E OR F) AND (-F) (-B OR E OR F) AND (B OR C) (-B OR E OR F) AND (-E OR -K) (E) AND (-E OR -K) (A OR -E) AND (E) (A OR -C OR -K) AND (B OR C) (-E OR -K) AND (E OR F) (A OR -E) AND (E OR F) (-F) AND (A OR -B OR F OR -K) (B OR C) AND (A OR -B OR F OR -K) (B OR -F) AND (F OR -K) (B OR -F) AND (-B OR E) (B OR -F) AND (A OR F) (B OR -F) AND (A OR C OR F OR -K) (B OR -F) AND (C OR E OR F)
---	---	--

A OR -F OR -K	(B OR -F) AND (A OR -B OR -K)
A OR E OR F OR -K	(-C OR E) AND (A OR C OR F OR -K)
-C	(-C OR E) AND (-E)
F	(K) AND (F OR -K)
{ }	(K) AND (-K)
-B OR F	(K) AND (-B OR F OR -K)
A OR C OR F	(K) AND (A OR C OR F OR -K)
A OR -B	(K) AND (A OR -B OR -K)
C OR F OR -K	(-A) AND (A OR C OR F OR -K)
-B OR -K	(-A) AND (A OR -B OR -K)
A OR C OR -K	(-F) AND (A OR C OR F OR -K)
C OR E	(-F) AND (C OR E OR F)
E OR F OR -K	(B OR E) AND (-B OR F OR -K)
A OR E OR F	(B OR E) AND (A OR -B OR F)
A OR E OR -K	(B OR E) AND (A OR -B OR -K)
B	(B OR E) AND (-E)
YES	

Test 05

File đính kèm

A OR B OR -D	5	
6	A OR -B OR F OR -K	(A OR -E OR -K) AND (-B OR E OR F)
A OR -E OR -K	-E OR -K	(A OR -E OR -K) AND (-A)
B OR -F	B OR C	(B OR -F) AND (C OR F)
C OR F	-F	(B OR -F) AND (-B)
-C OR D	D OR F	(C OR F) AND (-C OR D)
-K	7	
-B OR E OR F	B OR D	(B OR -F) AND (D OR F)
	C	(C OR F) AND (-F)
	-B OR E	(-B OR E OR F) AND (-F)
	C OR E OR F	(-B OR E OR F) AND (B OR C)
	-B OR F OR -K	(-B OR E OR F) AND (-E OR -K)
	A OR -B OR -K	(A OR -B OR F OR -K) AND (-F)
	A OR C OR F OR -K	(A OR -B OR F OR -K) AND (B OR C)
	13	
	E OR -F	(B OR -F) AND (-B OR E)
	A OR B OR C OR -K	(B OR -F) AND (A OR C OR F OR -K)
	B OR C OR E	(B OR -F) AND (C OR E OR F)
	A OR -F OR -K	(B OR -F) AND (A OR -B OR -K)
	A OR D OR F OR -K	(-C OR D) AND (A OR C OR F OR -K)
	D OR E OR F	(-C OR D) AND (C OR E OR F)
	C OR F OR -K	(-A) AND (A OR C OR F OR -K)
	-B OR -K	(-A) AND (A OR -B OR -K)
	A OR C OR -K	(-F) AND (A OR C OR F OR -K)
	C OR E	(-F) AND (C OR E OR F)
	D OR F OR -K	(-B OR F OR -K) AND (B OR D)
	D OR E	(-B OR E) AND (B OR D)

A OR D OR -K	(B OR D) AND (A OR -B OR -K)
15	
A OR B OR D OR -K	(B OR -F) AND (A OR D OR F OR -K)
-F OR -K	(B OR -F) AND (-B OR -K)
B OR C OR -K	(B OR -F) AND (C OR F OR -K)
B OR D OR -K	(B OR -F) AND (D OR F OR -K)
B OR D OR E	(B OR -F) AND (D OR E OR F)
A OR -B OR E OR -K	(-B OR E OR F) AND (A OR -F OR -K)
A OR C OR E OR F OR -K	(-B OR E OR F) AND (A OR B OR C OR -K)
C OR -K	(-A) AND (A OR C OR -K)
D OR -K	(-A) AND (A OR D OR -K)
C OR E OR F OR -K	(-B OR F OR -K) AND (B OR C OR E)
-B OR E OR -K	(-B OR F OR -K) AND (E OR -F)
A OR C OR E OR -K	(-B OR E) AND (A OR B OR C OR -K)
A OR D OR E OR -K	(A OR -F OR -K) AND (D OR E OR F)
C OR E OR -K	(B OR C OR E) AND (-B OR -K)
D OR E OR -K	(E OR -F) AND (D OR F OR -K)
6	
A OR B OR C OR E OR -K	(B OR -F) AND (A OR C OR E OR F OR -K)
E OR -F OR -K	(B OR -F) AND (-B OR E OR -K)
A OR E OR -F OR -K	(B OR -F) AND (A OR -B OR E OR -K)
B OR C OR E OR -K	(B OR -F) AND (C OR E OR F OR -K)
A OR D OR E OR F OR -K	(-C OR D) AND (A OR C OR E OR F OR -K)
D OR E OR F OR -K	(-C OR D) AND (C OR E OR F OR -K)
2	
B OR D OR E OR -K	(B OR -F) AND (D OR E OR F OR -K)
A OR B OR D OR E OR -K	(B OR -F) AND (A OR D OR E OR F OR -K)
0	
NO	

IV. Nhận xét và đề xuất giải pháp cải tiến của giải thuật hợp giải:

1. Nhận xét về hợp giải

1.1. Ưu điểm:

- Có tính tổng quát và được sử dụng phổ biến
- Logic định đề có tính cấu tạo (kết cấu): Ngữ nghĩa của mệnh đề $(S1 \wedge S2)$ được suy ra từ ngữ nghĩa của $S1$ và ngữ nghĩa của $S2$
- Ngữ nghĩa trong logic định đề không phụ thuộc ngữ cảnh (context-independent)
- Khả năng áp dụng tốt với các dạng câu Horn $A \wedge B \wedge \dots \wedge Z \rightarrow \alpha$
- Cho ra khối lượng lớn các thông tin từ một số thông tin ban đầu

1.2. Nhược điểm:

- Khả năng diễn đạt (biểu diễn) của logic định đề là rất hạn chế. Trong khi chuyển các câu tri thức về dạng mệnh đề, có thể làm đánh mất các thông tin kinh nghiệm có giá trị chứa trong các câu ban đầu.
- Logic mệnh đề phải liệt kê (xét) mọi khả năng gán giá trị chân lý (đúng/sai) cho X và Y . Do liên tục phải bổ sung các câu mới vào KB và lặp lại tìm kiếm các cặp câu có thể phân giải với

nhau được nên không gian và thời gian của giải thuật sẽ tăng rất nhanh, giải thuật phân giải làm việc không hiệu quả.

- Sinh ra rất nhiều sự kiện mà không liên quan gì đến câu truy vấn (vì bản chất của giải thuật này là tìm kiếm theo chiều rộng)
- Luật suy diễn hợp giải có tính đúng đắn (sound), nhưng không có tính hoàn chỉnh (incomplete)

Ví dụ:

+ Tập giả thiết (cơ sở tri thức) KB chứa biểu thức $(p \wedge q)$

+ Cần chứng minh: $(p \vee q)$

⇒ Luật suy diễn hợp giải không thể suy ra được biểu thức cần chứng minh!

2. Đề xuất giải pháp cải tiến:

Một cách khác để thực hiện suy diễn phân giải KB $\models q$ là xuất phát từ clause $\neg q$ (coi như trạng thái đích) ta thực hiện phân giải với các clauses khác trong KB để sinh ra clauses mới, rồi từ các clauses mới này thực hiện tiếp với các clauses khác của KB để sinh ra clauses mới hơn, đến khi nào [] được sinh ra hoặc không sinh ra được clause mới thì dừng. Nói cách khác là chỉ thực hiện phân giải các clauses liên quan đến q . Giải thuật phân giải lùi sẽ làm việc hiệu quả hơn giải thuật phân giải tiến. Do tập trung vào đích đã cho. Nó tạo ra một loạt câu hỏi chỉ liên quan đến vấn đề đang xét, thuận tiện đối với người dùng. Vì thế, khi suy diễn một điều gì từ thông tin đã biết, nó chỉ tìm trên một phần của cơ sở tri thức thích đáng đối với bài toán đang xét.

V. Đánh giá mức độ hoàn thành của nhóm:

Tiêu chí	Đánh giá
Đọc dữ liệu đầu vào và lưu trong cấu trúc dữ liệu phù hợp	5%
Cài đặt giải thuật hợp giải trên logic mệnh đề	10%
Các bước suy diễn phát sinh đủ mệnh đề và kết luận đúng	30%
Tuân thủ mô tả định dạng của đề bài	5%
Báo cáo test case và đánh giá	10%

BÀI 2: BIỂU DIỄN TRI THỨC BẰNG LOGIC BẬC NHẤT VỚI PROLOG

I. Cây phả hệ gia đình Hoàng gia Anh:

1. Xây dựng cơ sở tri thức mô tả các quan hệ:

- parent(Person, Child)
- male(Person)
- female(Person)
- married(Person, Person)
- divorced(Person, Person)

2. Định nghĩa các vị từ sau dựa vào những vị từ đã có ở trên:

```
husband(Person, Wife) :- married(Person, Wife), male(Person).
wife(Person, Husband) :- married(Person, Husband), female(Person).
father(Parent, Child) :- parent(Parent, Child), male(Parent).
mother(Parent, Child) :- parent(Parent, Child), female(Parent).
child(Child, Parent) :- parent(Parent, Child).
son(Child, Parent) :- parent(Parent, Child), male(Child).
daughter(Child, Parent) :- parent(Parent, Child), female(Child).

grandparent(GP, GC) :- parent(GP, Parent), parent(Parent, GC).
grandmother(GM, GC) :- parent(GM, Parent), parent(Parent, GC), female(GM).
grandfather(GF, GC) :- parent(GF, Parent), parent(Parent, GC), male(GF).
grandchild(GC, GP) :- parent(GP, Parent), parent(Parent, GC).
grandson(GS, GP) :- parent(GP, Parent), parent(Parent, GS), male(GS).
granddaughter(GD, GP) :- parent(GP, Parent), parent(Parent, GD), female(GD).

sibling(Person1, Person2) :- parent(Parent, Person1), parent(Parent, Person2), Person1 \== Person2.
brother(Person, Sibling) :- sibling(Person, Sibling), male(Person).
sister(Person, Sibling) :- sibling(Person, Sibling), female(Person).
aunt(Person, NieceNephew) :- parent(Parent, NieceNephew), (sister(Person, Parent); (brother(Uncle, Parent), wife(Person, Uncle))).
uncle(Person, NieceNephew) :- parent(Parent, NieceNephew), (brother(Person, Parent); (sister(Aunt, Parent), husband(Person, Aunt))).
niece(Person, AuntUncle) :- (aunt(AuntUncle, Person); uncle(AuntUncle, Person)), female(Person).
nephew(Person, AuntUncle) :- (aunt(AuntUncle, Person); uncle(AuntUncle, Person)), male(Person).
```

3. Các câu hỏi xây dựng:

```
%Xu ly ket qua trung nhau: setof
%http://www.cse.unsw.edu.au/~billw/dictionaries/prolog/setof.html

%load KB
:- [hoangGiaAnh].

:- writeln('Cau 1: Ai la chong cua Diana Pricess of Wales ?').
:- writeln('Cau 2: Ai la vo cu Prince Phillip ?').
:- writeln('Cau 3: Ai la cha cua Princess Beatrice of York ?').
:- writeln('Cau 4: Ai la me cua Prince Charles ?').
:- writeln('Cau 5: Ai la con cua Prince Edward ?').
:- writeln('Cau 6: Ai la con trai cua Princess Anne ?').
:- writeln('Cau 7: Ai la con gai cua Kate Middleton ?').
:- writeln('Cau 8: Ai la ong/ba cua Archie Harrison Mountbatten-Windsor ?').
:- writeln('Cau 9: Ai la ba cua Zara Tindall ?').
:- writeln('Cau 10: Ai la ong cua Pricess Eugenie ?').
:- writeln('Cau 11: Ai la chau cua Queen Elizabeth II ?').
:- writeln('Cau 12: Ai la chay trai cua Prince Phillip ?').
:- writeln('Cau 13: Ai la chau gai cua Prince Charles ?').
:- writeln('Cau 14: Ai la anh/chi/em ruot cua Prince George ?').
:- writeln('Cau 15: Ai la anh/em trai cua Lady Louise Windsor ?').
:- writeln('Cau 16: Ai la chi/em gai cua Peter Phillips ?').
:- writeln('Cau 17: Ai la di cua Prince William ?').
:- writeln('Cau 18: Ai la cau cua James, Viscount Severn?').
:- writeln('Cau 19: Ai la chau gai cua Sarah Ferguson ?').
:- writeln('Cau 20: Ai la chau trai cua Camilla Parker Bowles ?').
```

4. Ví dụ chạy chương trình

```
hvtu@hvtu-Inspiron-3459:~/Desktop/hcmus_ai_p2_logic/Bai_2$ swipl hoangGiaAnh_test.pl
Cau 1: Ai la chong cua Diana Pricess of Wales ?
Cau 2: Ai la vo cu Prince Phillip ?
Cau 3: Ai la cha cua Princess Beatrice of York ?
Cau 4: Ai la me cua Prince Charles ?
Cau 5: Ai la con cua Prince Edward ?
Cau 6: Ai la con trai cua Princess Anne ?
Cau 7: Ai la con gai cua Kate Middleton ?
Cau 8: Ai la ong/ba cua Archie Harrison Mountbatten-Windsor ?
Cau 9: Ai la ba cua Zara Tindall ?
Cau 10: Ai la ong cua Pricess Eugenie ?
Cau 11: Ai la chau cua Queen Elizabeth II ?
Cau 12: Ai la chay trai cua Prince Phillip ?
Cau 13: Ai la chau gai cua Prince Charles ?
Cau 14: Ai la anh/chi/em ruot cua Prince George ?
Cau 15: Ai la anh/em trai cua Lady Louise Windsor ?
Cau 16: Ai la chi/em gai cua Peter Phillips ?
Cau 17: Ai la di cua Prince William ?
Cau 18: Ai la cau cua James, Viscount Severn?
Cau 19: Ai la chau gai cua Sarah Ferguson ?
Cau 20: Ai la chau trai cua Camilla Parker Bowles ?

Press cau_1, cau_2, cau_3,..., cau_20 to see question and answer.
Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- 
```

```
?- cau_1.  
Ai la chong cua Diana Pricess of Wales ?  
Khong biet  
true.  
  
?- cau_5.  
Ai la con cua Prince Edward ?  
[james_severn,lady_loulse]  
true.  
  
?- cau_15.  
Ai la anh/em trai cua Lady Louise Windsor ?  
[james_severn]  
true.  
  
?- cau_20.  
Ai la chau trai cua Camilla Parker Bowles ?  
[james_severn,peter_phillips]  
true.  
  
?- 
```

II. Quản lý trường đại học:

1. Xây dựng các quan hệ trong trường học:

Gồm:

- **Khoa** quản lý **Bộ môn**, **Phòng thí nghiệm**.
- **Bộ môn**, **Phòng thí nghiệm**, **Phòng ban** gồm **Giáo viên đứng đầu**, danh sách **Giáo viên** làm việc cho đơn vị, **Năm thành lập**.
- **Giáo viên** có **Họ tên**, **Quê quán**, **Giới tính**, **Ngày sinh**, **Lương**, **Học hàm**, **Học vị**.



2. Xây dựng mô hình:

2.1. Xây dựng các quan hệ:

- bo_mon, khoa, phong_thi_nghiem, phong_ban, giao_vien
- nam_thanh_lap(bo_mon, nam_tl)
nam_thanh_lap(khoa, nam_tl)
nam_thanh_lap(phong_thi_nghiem, nam_tl)
- truc_thuoc(bo_mon, khoa), truc_thuoc(phong_thi_nghiem, khoa)
truc_thuoc(giao_vien, bo_mon), truc_thuoc(giao_vien, phong_thi_nghiem),
truc_thuoc(giao_vien, phong_ban)
- nguoi_dung_dau(khoa, giao_vien)
nguoi_dung_dau(phong_thi_nghiem, giao_vien)
nguoi_dung_dau(phong_ban, giao_vien)
- que_quan(giao_vien, dia_chi)
- ngay_sinh(giao_vien, ngay_thang)
- luong(giao_vien, tien_luong)
- nam(giao_vien)
nu(giao_vien)
- thac_si(giao_vien)
tien_si(giao_vien)

- giao_su(giao_vien)
- pho_giao_su(giao_vien)

2.2. Xây dựng các quan hệ khác dựa trên các quan hệ đã có:

- giao_vien_thuoc_bo_mon(GV, BM)
- giao_vien_thuoc_khoa(GV, K)
- giao_vien_thuoc_phong_thi_nghiem(GV, PTN)
- giao_vien_thuoc_phong_ban(GV, PB)
- bo_mon_thuoc_khoa(BM, K)
- phong_thi_nghiem_thuoc_khoa(PTN, K)

- bo_mon_cung_khoa(BM1, BM2)
- phong_thi_nghiem_cung_khoa(PTN1, PTN2)
- khoa_thanh_lap_cung_nam(K1, K2)
- bo_mon_thanh_lap_cung_nam(BM1, BM2)
- phong_TN_thanh_lap_cung_nam(PTN1, PTN2)

- truong_khoa(TK, K)
- truong_phong_ban(TPB, PB)
- truong_phong_thi_nghiem(TPTN, PTN)
- truong_bo_mon(TBM, BM)
- tuoi(GV, Tuoi)

- giao_vien_cung_bo_mon(GV1, GV2)
- giao_vien_cung_khoa(GV1, GV2)
- giao_vien_cung_phong_thi_nghiem(GV1, GV2)
- giao_vien_cung_phong_ban(GV1, GV2)

- cung_gioi_tinh(GV1, GV2)
- bang_tuoi(GV1, GV2)
- cung_ngay_sinh(GV1, GV2)
- cung_thang_sinh(GV1, GV2)
- sinh_cung_ngay(GV1, GV2)
- lon_tuoi_hon(GV1, GV2)
- nho_tuoi_hon(GV1, GV2)
- cung_que_quan(GV1, GV2)
- luong_bang_nhau(GV1, GV2)
- luong_cao_hon(GV1, GV2)
- luong_thap_hon(GV1, GV2)
- cung_hoc_ham(GV1, GV2)
- cung_hoc_vi(GV1, GV2)

3. Các câu hỏi xây dựng:

```
%load KB
:- [truongDaiHoc].

:- writeln('Cau 1: PTN Tri tue nhan tao thuoc khoa CNTT phai khong?').
:- writeln('Cau 2: Khoa Toan hoc thanh lap nam 2000 phai khong?').
:- writeln('Cau 3: Khoa Vat ly co cac bo mon nao?').
:- writeln('Cau 4: PTN Ky thuat hat nhan thanh lap nam nao?').
:- writeln('Cau 5: Giao vien Tran Thai Son thuoc phong ban nao?').
:- writeln('Cau 6: Giao vien Le Hoai Bac thuoc bo mon Khoa hoc may tinh phai khong?').
:- writeln('Cau 7: Giao vien Tran Minh Triet la truong PTN Cong nghephan mem phai khong?').
:- writeln('Cau 8: Luong cua giao vien Dinh Ba Tien la bao nhieu?').
:- writeln('Cau 9: Luong cua giao vien Tran Vu lon hon cua giao vien Tran Trung Dung phai khong?').
:- writeln('Cau 10: Que cua giao vien Ly Quoc Ngoc o dau?').
:- writeln('Cau 11: Giao vien cung que voi giao vien Nguyen Chi Nhan?').
:- writeln('Cau 12: Giao vien Trinh Thanh Deo nam nay bao nhieu tuoi?').
:- writeln('Cau 13: Giao vien Nguyen Thi Thanh Mai thuoc bo mon nao?').
:- writeln('Cau 14: Giao vien cung tuoi voi giao vien Truong Thi Hong Loan?').
:- writeln('Cau 15: Ai la truong khoa Vat ly?').
:- writeln('Cau 16: Giao vien Le Vu Tuan Hung chung khoa voi giao vien Nguyen Chi Nhan phai khong?').
:- writeln('Cau 17: Giao vien Vu Hai Quan la Tien si phai khong?').
:- writeln('Cau 18: Liet ke danh sach thacsu.').
:- writeln('Cau 19: Liet ke danh sach giao su.').
:- writeln('Cau 20: Liet ke danh sach giao vien thuoc khoa CNTT.').
```

4. Ví dụ chạy chương trình:

```
hvtu@hvtu-Inspiron-3459:~/Desktop/hcmus_ai_p2_logic/Bai_2$ swipl truongDaiHoc_test.pl
Cau 1: PTN Tri tue nhan tao thuoc khoa CNTT phai khong?
Cau 2: Khoa Toan hoc thanh lap nam 2000 phai khong?
Cau 3: Khoa Vat ly co cac bo mon nao?
Cau 4: PTN Ky thuat hat nhan thanh lap nam nao?
Cau 5: Giao vien Tran Thai Son thuoc phong ban nao?
Cau 6: Giao vien Le Hoai Bac thuoc bo mon Khoa hoc may tinh phai khong?
Cau 7: Giao vien Tran Minh Triet la truong PTN Cong nghephan mem phai khong?
Cau 8: Luong cua giao vien Dinh Ba Tien la bao nhieu?
Cau 9: Luong cua giao vien Tran Vu lon hon cua giao vien Tran Trung Dung phai khong?
Cau 10: Que cua giao vien Ly Quoc Ngoc o dau?
Cau 11: Giao vien cung que voi giao vien Nguyen Chi Nhan?
Cau 12: Giao vien Trinh Thanh Deo nam nay bao nhieu tuoi?
Cau 13: Giao vien Nguyen Thi Thanh Mai thuoc bo mon nao?
Cau 14: Giao vien cung tuoi voi giao vien Truong Thi Hong Loan?
Cau 15: Ai la truong khoa Vat ly?
Cau 16: Giao vien Le Vu Tuan Hung chung khoa voi giao vien Nguyen Chi Nhan phai khong?
Cau 17: Giao vien Vu Hai Quan la Tien si phai khong?
Cau 18: Liet ke danh sach thacsu.
Cau 19: Liet ke danh sach giao su.
Cau 20: Liet ke danh sach giao vien thuoc khoa CNTT.

Press cau_1, cau_2, cau_3,..., cau_20 to see question and answer.
Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- 
```



```
?- cau_1.  
PTN Tri tue nhan tao thuoc khoa CNTT phai khong?  
Phai  
true.  
  
?- cau_5.  
Giao vien Tran Thai Son thuoc phong ban nao?  
Phong dao tao  
true.  
  
?- cau_15.  
Ai la truong khoa Vat ly?  
Truong Thi Hong Loan  
true.  
  
?- cau_20.  
Liet ke danh sach giao vien thuoc khoa CNTT.  
[Dinh Ba Tien,Le Hoai Bac,Ly Quoc Ngoc,Tran Minh Triet,Tran Trung Dung,Vu Hai Quan]  
true.  
  
?- 
```

BÀI 3: SUY DIỄN TRÊN LOGIC BẬC NHẤT

Áp dụng thuật toán suy diễn tiến

```
FOL-FC-Ask(KB,α) {
  repeat until new là rỗng
    new ← {}
    for each câu r trong KB // r ở dạng chuẩn hóa  $(p_1 \wedge \dots \wedge p_n \Rightarrow q)$ 
      for each phép thế  $\theta$  sao cho  $(p_1 \wedge \dots \wedge p_n)\theta = (p'_1 \wedge \dots \wedge p'_n)\theta$ 
        với  $p'_1, \dots, p'_n$  nào đó trong KB
           $q' \leftarrow \text{Subst}(\theta, q)$ 
          if q' không phải là một câu đã có trong KB hay new then
            thêm q' vào new
             $\phi \leftarrow \text{Unify}(q', \alpha)$  // nếu đạt tới kết luận  $\alpha$ 
            if  $\phi$  thành công then return  $\phi$ 
  thêm new vào KB
  return false
}
```

I. Một số thay đổi của hệ thống tự cài đặt so với Prolog:

1. Ký hiệu phép kết:

Phép “kéo theo” từ “:-” thay bằng “->”

phép “và” từ “,” thay bằng dấu “^”

2. Thứ tự câu suy được ghi từ trái sang phải (so với từ phải sang trái như prolog)

Ví dụ

$\text{grandparent}(\text{GP}, \text{GC}) \text{ :- } \text{parent}(\text{GP}, \text{Parent}), \text{parent}(\text{Parent}, \text{GC})$

Thì sẽ được ghi là

$\text{parent}(\text{Parent}, \text{GC}) \wedge \text{parent}(\text{GP}, \text{Parent}) \text{ -> } \text{grandparent}(\text{GP}, \text{GC})$

3. Khi trả lời câu hỏi nếu có nhiều đáp án ví dụ cha có nhiều con thì sẽ xuất hết tất cả đáp án

4. Kết quả trả về câu hỏi

Trong prolog nếu tìm được thì trả về kết quả, nếu không thì trả về false

Trong hệ thống tự cài đặt, nếu tìm được thì sẽ trả về tất cả câu trả lời dạng quan hệ (nếu có nhiều câu trả lời) nếu không có thì không trả về gì.

Ví dụ với câu hỏi

Ai là con của child(Child, prince_phillip)

	Hệ thống tự cài đặt	Prolog
Cú pháp đặt câu hỏi	Child(X, prince_phillip) X là từ bất kỳ viết hoa chữ đầu	setof(Person, child(Person, prince_philip), List).
Kết quả trả về	child(prince_charles, prince_philip) child(prince_andrew, prince_philip) child(princess_anne, prince_philip) child(prince_edward, prince_philip)	List = [prince_andrew, prince_charles,

		prince_edward, princess_anne].
--	--	-----------------------------------

II. Kiểm chứng kết quả của hệ thống tự cài đặt so với Prolog

1. Hoàng gia Anh

Kết quả chạy chương trình

File input chứa KB, các phép suy diễn và câu hỏi, file output chứa câu hỏi và câu trả lời

The image shows two Notepad windows side-by-side. The left window, titled 'hoang_gia_anh - Notepad', contains a list of Prolog facts defining the royal family of the United Kingdom. The right window, titled 'output - Notepad', shows the results of queries entered into the Prolog system.

Left Window (hoang_gia_anh - Notepad):

```
parent(prince_philip,prince_charles)
parent(prince_philip,prince_andrew)
parent(prince_philip,princess_anne)
parent(prince_philip,prince_edward)
parent(queen_elizabeth_II,prince_charles)
parent(queen_elizabeth_II,prince_andrew)
parent(queen_elizabeth_II,princess_anne)
parent(queen_elizabeth_II,prince_edward)
parent(prince_charles,prince_william)
parent(prince_charles,prince_harry)
parent(princess_diana,prince_william)
parent(princess_diana,prince_harry)
parent(prince_william,prince_george)
parent(prince_william,princess_charlotte)
parent(prince_william,prince_louis)
parent(kate_middleton,prince_george)
parent(kate_middleton,princess_charlotte)
parent(kate_middleton,prince_louis)
parent(prince_harry,archie_harrison)
parent(meghan_markle,archie_harrison)
parent(prince_andrew,princess_eugenie)
parent(prince_andrew,princess_beatrice)
parent(sarah_ferguson,princess_eugenie)
parent(sarah_ferguson,princess_beatrice)
parent(mark_phillips,peter_phillips)
parent(mark_phillips,zara_tindall)
parent(princess_anne,peter_phillips)
parent(princess_anne,zara_tindall)
parent(prince_edward,lady_loulse)
parent(prince_edward,james_severn)
parent(sophie_rhys_jones,lady_loulse)
parent(sophie_rhys_jones,james_severn)
male(prince_philip)
male(prince_charles)
male(prince_andrew)
male(prince_william)
male(prince_harry)
male(prince_george)
male(prince_louis)
male(archie_harrison)
male(mark_phillips)
male(timothy_laurence)
male(prince_edward)
male(peter_phillips)
male(mike_tindall)
male(james_severn)
female(queen_elizabeth_II)
female(princess_diana)
female(camilla_bowles)
female(sarah_ferguson)
female(kate_middleton)
female(meghan_markle)
female(princess_eugenie)
female(princess_beatrice)
female(princess_charlotte)
female(princess_anne)
female(sophie_rhys_jones)
female(autumn_phillips)
female(zara_tindall)
```

Right Window (output - Notepad):

```
Ai la chong cua Queen Elizabeth II ?
husband(prince_philip,queen_elizabeth_II)
Ai la chong cua Prince Diana ?
husband(prince_charles,princess_diana)
Ai la vo cua Prince Philip?
wife(queen_elizabeth_II,prince_philip)
Ai la vo cua Prince Harry?
wife(meghan_markle,prince_harry)
Ai la cha cua Prince Charles?
father(prince_philip,prince_charles)
Ai la cha cua Prince Andrew?
father(prince_philip,prince_andrew)
Ai la me cua Prince Charles?
mother(queen_elizabeth_II,prince_charles)
Ai la me cua Prince Andrew?
mother(queen_elizabeth_II,prince_andrew)
Ai la con cua Prince Phillip?
child(prince_charles,prince_philip)
child(prince_andrew,prince_philip)
child(princess_anne,prince_philip)
child(prince_edward,prince_philip)
Ai la con cua Prince Charles?
child(prince_william,prince_charles)
child(prince_harry,prince_charles)
Ai la con trai cua Prince William?
son(prince_william,princess_diana)
son(prince_harry,princess_diana)
Ai la con trai cua Prince Louis?
son(prince_george,prince_william)
son(prince_louis,prince_william)
Ai la con gái cua Prince Edward?
daughter(lady_loulse,prince_edward)
Ai la con gái cua Prince Phillip?
daughter(princess_anne,prince_philip)
Ai la ong ba cua Princess Eugenie?
grandparent(prince_philip,princess_eugenie)
grandparent(queen_elizabeth_II,princess_eugenie)
Ai la ong ba cua Princess Beatrice?
grandparent(prince_philip,princess_beatrice)
grandparent(queen_elizabeth_II,princess_beatrice)
Ai la ong ba ngoài của Archie Harrison?
grandmother(princess_diana,archie_harrison)
Ai la ong ba ngoài của Prince Louis?
grandmother(queen_elizabeth_II,lady_loulse)
Ai la ong ba nội của Archie Harrison?
grandfather(prince_charles,archie_harrison)
Ai la ong ba nội của Prince Louis?
grandfather(prince_philip,lady_loulse)
```

Kiểm tra lại với Prolog

Kết quả file output của hệ thống tự cài đặt	Kết quả trên prolog
Ai là chồng của Queen Elizabeth II ? husband(prince_philip, queen_elizabeth_II)	?- husband(Person, queen_elizabeth_II). Person = prince_philip.
Ai là chồng của Prince Diana ? husband(prince_charles, princess_diana)	?- husband(Person, princess_diana). false.
Ai là vợ của Prince Philip? wife(queen_elizabeth_II, prince_philip)	?- wife(Person, prince_philip). Person = queen_elizabeth_II.
Ai là vợ của Prince Harry? wife(meghan_markle, prince_harry)	?- wife(Person, prince_harry). Person = megan_markle.
Ai là cha của Prince Charles? father(prince_philip, prince_charles)	?- father(Person, prince_charles). Person = prince_philip
Ai là cha của Prince Andrew? father(prince_philip, prince_andrew)	?- father(Person, prince_andrew). Person = prince_philip .
Ai là mẹ của Prince Charles? mother(queen_elizabeth_II, prince_charles)	?- mother(Person, prince_charles). Person = queen_elizabeth_II.
Ai là mẹ của Prince Andrew? mother(queen_elizabeth_II, prince_andrew)	?- mother(Person, prince_andrew). Person = queen_elizabeth_II.
Ai là con của Prince Phillip? child(prince_charles, prince_philip) child(prince_andrew, prince_philip) child(princess_anne, prince_philip) child(prince_edward, prince_philip)	?- setof(Person, child(Person, prince_philip), List). List = [prince_andrew, prince_charles, prince_edward, princess_anne].
Ai là con của Prince Charles? child(prince_william, prince_charles) child(prince_harry, prince_charles)	?- setof(Person, child(Person, prince_charles), List). List = [prince_harry, prince_william].
Ai là con trai của Princess Diana? son(prince_william, princess_diana) son(prince_harry, princess_diana)	?- setof(Person, son(Person, princess_diana), List). List = [prince_harry, prince_william].

Ai là con trai của Prince William? son(prince_george,prince_william) son(prince_louis,prince_william)	?- setof(Person,son(Person,prince_william),List). List = [prince_george, prince_louis].
Ai là con gái của Princess Edward? daughter(lady_louise,prince_edward)	?- setof(Person,daughter(Person,prince_edward),List). List = [lady_louise].
Ai là con gái của Prince Phillip? daughter(princess_anne,prince_philip)	?- setof(Person,daughter(Person,prince_philip),List). List = [princess_anne]
Ai là ông bà của Princess Eugenie? grandparent(prince_philip,princess_eugenie) grandparent(queen_elizabeth_II,princess_eugenie)	?- setof(Person,grandparent(Person,princess_eugenie),List). List = [prince_philip, queen_elizabeth_II].
Ai là ông bà của Princess Beatrice? grandparent(prince_philip,princess_beatrice) grandparent(queen_elizabeth_II,princess_beatrice)	?- setof(Person,grandparent(Person,princess_beatrice),List). List = [prince_philip, queen_elizabeth_II].
Ai là ông bà ngoại của Archie Harrison? grandmother(princess_diana,archie_harrison)	?- setof(Person,grandmother(Person,archie_harrison),List). List = [princess_diana].
Ai là ông bà ngoại của Prince Louis? grandmother(queen_elizabeth_II,lady_louise)	?- setof(Person,grandmother(Person,lady_louise),List). List = [queen_elizabeth_II].
Ai là ông bà nội của Archie Harrison? grandfather(prince_charles,archie_harrison)	?- setof(Person,grandfather(Person,archie_harrison),List). List = [prince_charles].
Ai là ông bà nội của Prince Louis? grandfather(prince_philip,lady_louise)	?- setof(Person,grandfather(Person,lady_louise),List).

Luong bang voi Nguyen Thi Cam Loan? luong_bang_nhau(Truong Thi Hong Loan,Nguyen Thi Cam Loan) luong_bang_nhau(Nguyen Thi Thanh Mai,Nguyen Thi Cam Loan) luong_bang_nhau(Trinh Thanh Deo,Nguyen Thi Cam Loan)	?- setof(Person,luong_bang_nhau(Person,' Nguyen Thi Cam Loan'),List). List = ['Nguyen Thi Thanh Mai', 'Trinh Thanh Deo', 'Truong Thi Hong Loan'].
Luong bang nhau voi Nguyen Thi Thanh Mai? luong_bang_nhau(Truong Thi Hong Loan,Nguyen Thi Thanh Mai) luong_bang_nhau(Nguyen Thi Cam Loan,Nguyen Thi Thanh Mai) luong_bang_nhau(Trinh Thanh Deo,Nguyen Thi Thanh Mai)	?- setof(Person,luong_bang_nhau(Person,' Nguyen Thi Thanh Mai'),List). List = ['Nguyen Thi Cam Loan', 'Trinh Thanh Deo', 'Truong Thi Hong Loan'].
Cung que quan voi Tran Trung Dung? cung_que_quan(Ly Quoc Ngoc,Tran Trung Dung)	?- setof(Person,cung_que_quan(Person,'Tr an Trung Dung'),List). List = ['Ly Quoc Ngoc'].
Cung que quan voi Ly Quoc Ngoc? cung_que_quan(Tran Trung Dung,Ly Quoc Ngoc)	?- setof(Person,cung_que_quan(Person,'L y Quoc Ngoc'),List). List = ['Tran Trung Dung'].
Cung hoc vi voi Vu Hai Quan? cung_hoc_vi(Dinh Ba Tien,Vu Hai Quan) cung_hoc_vi(Tran Minh Triet,Vu Hai Quan) cung_hoc_vi(Trinh Thanh Deo,Vu Hai Quan)	?- setof(Person,cung_hoc_vi(Person,'Vu Hai Quan'),List). List = ['Dinh Ba Tien', 'Tran Minh Triet', 'Trinh Thanh Deo'].
Cung hoc vi voi Tran Minh Triet? cung_hoc_vi(Dinh Ba Tien,Tran Minh Triet) cung_hoc_vi(Vu Hai Quan,Tran Minh Triet) cung_hoc_vi(Trinh Thanh Deo,Tran Minh Triet)	?- setof(Person,cung_hoc_vi(Person,'Tran Minh Triet'),List). List = ['Dinh Ba Tien', 'Trinh Thanh Deo', 'Vu Hai Quan'].
Cung hoc ham voi Vu Hai Quan? cung_hoc_ham(Tran Minh Triet,vu Hai Quan)	?- setof(Person,cung_hoc_ham(Person,'V u Hai Quan'),List). List = ['Tran Minh Triet'].
Cung hoc ham voi Le Hoai Bac?	?- setof(Person,cung_hoc_ham(Person,'Le Hoai Bac'),List). false.
Truong khoa CNTT? truong_khoa(Dinh_Ba_Tien,CNTT)	?- truong_khoa(Person,CNTT).

	Person = 'Dinh Ba Tien', CNTT = 'CNTT'
Truong khoa Vat ly? truong_khoa(Le Vu Tuan Hung, vat ly)	?- truong_khoa(Person, 'Vat ly'). Person = 'Le Vu Tuan Hung'.
Truong khoa Toan hoc? truong_khoa(Huynh Quang Vu, Toan hoc)	?- truong_khoa(Person, 'Toan hoc'). Person = 'Huynh Quang Vu'.
Giao vien cua bo mon Hoa Ly? giao_vien_thuoc_bo_mon(Nguyen Thi Thanh Mai, Hoa ly)	?- setof(Person, giao_vien_thuoc_bo_mon(Person, 'Hoa ly'), List). List = ['Nguyen Thi Thanh Mai'].
Giao vien cua bo mon Khoa hoc may tinh? giao_vien_thuoc_bo_mon(Le Hoai Bac, Khoa hoc may tinh) giao_vien_thuoc_bo_mon(Tran Trung Dung, Khoa hoc may tinh)	?- setof(Person, giao_vien_thuoc_bo_mon(Person, 'Khoa hoc may tinh'), List). List = ['Le Hoai Bac', 'Tran Trung Dung'].
Giao vien cung bo mon voi Tran Trung Dung? giao_vien_cung_bo_mon(Le Hoai Bac, Tran Trung Dung)	?- setof(Person, giao_vien_cung_bo_mon(Person, 'Tran Trung Dung'), List). List = ['Le Hoai Bac'].
Giao vien cung bo mon voi Le Hoai Bac? giao_vien_cung_bo_mon(Tran Trung Dung, Le Hoai Bac)	<input type="checkbox"/> ?- setof(Person, giao_vien_cung_bo_mon(Person, 'Le Hoai Bac'), List). List = ['Tran Trung Dung'].
Truong phong ban ke hoach tai chinh? truong_phong_ban(Nguyen Thi Cam Loan, Phong ke hoach tai chinh)	?- truong_phong_ban(Person, 'Phong ke hoach tai chinh'). Person = 'Nguyen Thi Cam Loan'.
Cung gioi tinh voi Nguyen Thi Thanh Mai? cung_gioi_tinh(Truong Thi Hong Loan, Nguyen Thi Thanh Mai) cung_gioi_tinh(Nguyen Thi Cam Loan, Nguyen Thi Thanh Mai) cung_gioi_tinh(Trinh Thanh Deo, Nguyen Thi Thanh Mai)	?- setof(Person, cung_gioi_tinh(Person, 'Nguyen Thi Thanh Mai'), List). List = ['Nguyen Thi Cam Loan', 'Trinh Thanh Deo', 'Truong Thi Hong Loan'].

Giao vien thuoc phong dao tao? giao_vien_thuoc_phong_ban(Tran Thai Son,Phong dao tao)	?- setof(Person,giao_vien_thuoc_phong_ban(Person,'Phong dao tao'),List). List = ['Tran Thai Son'].
Giao vien thuoc phong ke hoach tai chinh? giao_vien_thuoc_phong_ban(Nguyen Thi Cam Loan,Phong ke hoach tai chinh)	?- setof(Person,giao_vien_thuoc_phong_ban(Person,'Phong ke hoach tai chinh'),List). List = ['Nguyen Thi Cam Loan'].

Nhận xét: Kết quả trên hệ thống tự cài đặt và prolog đều cho kết quả giống nhau, tuy nhiên ở hệ thống tự cài đặt vì sử dụng phép suy diễn tiến nên ra kết quả chậm so với prolog sử dụng suy diễn lùi (từ câu hỏi truy ngược trở lại chứ không tìm hết tất cả như suy diễn tiến).