

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN**

Đồ án 2
CƠ SỞ TRÍ TUỆ NHÂN TẠO
Nhóm: Ultra Team

LOGIC BẬC NHẤT

Học kỳ I
Năm học: 2018- 2019

1

1 MỞ ĐẦU.....	2
1.1 Giới thiệu nhóm và phân công công việc.....	2
1.2 Mô tả đồ án.....	2
2 GIỚI THIỆU ĐỒ ÁN.....	2
2.1 Thời gian thực hiện.....	2
2.2 Các bước thực hiện đồ án.....	2
2.3 Bảng phân công công việc:.....	2
3 NỘI DUNG ĐỒ ÁN.....	3
3.1 Làm quen với công cụ Prolog.....	3
3.1.1 Yêu cầu 1: <i>Tìm hiểu ngôn ngữ Prolog.</i>	3
3.1.2 Yêu cầu 2: <i>Tìm hiểu môi trường lập trình Prolog. Cách thức triển khai ngôn ngữ Prolog trên công cụ đã tìm hiểu.</i>	8
1.1 Giới thiệu chung.....	8
1.2 Cách hoạt động cơ bản.....	8
Bắt đầu chương trình.....	8
Thoát.....	12
1.3 Một số hàm hỗ trợ khác sử dụng trong đồ án.....	12
Thao tác với file.....	12
Chạy file có sẵn.....	12
3.1.3 Yêu cầu 2: <i>Giải quyết bài toán suy diễn bằng ngôn ngữ Prolog trên công cụ SWI-Prolog. Xây dựng cây phá hệ chỗ giá đình Hoàng giá Anh như minh họa ở hình bên dưới.</i>	14
3.2 Xây dựng cơ sở tri thức với công cụ Prolog.....	20
3.3 Cài đặt hệ thống suy diễn bằng Ngôn ngữ lập trình.....	20
4 TÀI LIỆU THAM KHẢO.....	20

1 MỞ ĐẦU

1.1 Giới thiệu nhóm và phân công công việc

STT	MSSV	Họ và tên
1	1612001	Hoàng Dân An
2	1612165	Nguyễn Đào Vinh Hải
3	1612831	Bùi Thúy Vy

1.2 Mô tả đồ án

- Tìm hiểu các đặc điểm của môi trường lập trình Prolog.
- Dùng Prolog để xây dựng cơ sở tri thức sao cho tồn tại nhiều quan hệ cấp bậc giữa các đối tượng.
- Xây dựng chương trình suy diễn logic bằng cả ba phương pháp suy diễn:
 - o suy diễn tiến
 - o suy diễn lùi
 - o hợp giải

2 GIỚI THIỆU ĐỒ ÁN

2.1 Thời gian thực hiện

- Đồ án bắt đầu từ ngày 14/11 đến ngày 27/11/2018.

2.2 Các bước thực hiện đồ án

- Tìm hiểu ngôn ngữ logic Prolog
- Tìm hiểu môi trường SWI-Prolog (cài đặt, chạy thử)
- Viết báo cáo về Prolog và SWI-Prolog
- Suy nghĩ, thực hiện bài tập cây phả hệ Hoàng Gia Anh, bài tập xây dựng cơ sở tri thức
- Viết bộ câu hỏi và test script tương ứng với 2 bài tập trên.
- Tìm phương pháp và cài đặt hệ thống suy diễn bằng Python.
- Test chương trình với 2 bộ test đã làm.

2.3 Bảng phân công công việc:

Công việc	Người phụ trách	Mức độ	
-----------	-----------------	--------	--

		Hoàn thành	
Tìm hiểu ngôn ngữ Prolog.	Bùi Thúy Vy	100%	40%
Tìm hiểu môi trường lập trình Prolog (SWI-Prolog). (40%)	Hoàng Dân An	100%	
Giải quyết bài toán cây phá hệ của Hoàng Gia Anh	Bùi Thúy Vy	100%	
Viết test script và bộ câu hỏi cho bài toán Cây phá hệ	Hoàng Dân An	100%	
Giải quyết bài toán xây dựng cơ sở tri thức (30%)	Nguyễn Đào Vinh Hải	100%	30%
Viết test script và bộ câu hỏi cho bài toán đã xây dựng	Nguyễn Đào Vinh Hải	30%	
Xây dựng hệ thống suy diễn giống Prolog bằng Python	Hoàng Dân An	30%	30%
Tổng cộng			70%

3 NỘI DUNG ĐỒ ÁN

3.1 Làm quen với công cụ Prolog

3.1.1 Yêu cầu 1: Tìm hiểu ngôn ngữ Prolog.

a. Tổng quan.

Prolog là một ngôn ngữ lập trình. Tên gọi Prolog được xuất phát từ cụm từ tiếng Pháp **Programmation en logique**, nghĩa là “lập trình theo logic”, xuất hiện từ năm 1972, do Alain Colmerauer và Robert Kowalski thiết kế, mục tiêu của Prolog là giúp người dùng mô tả lại bài toán trên ngôn ngữ logic, theo đó máy tính sẽ tiến hành suy diễn tự động dựa vào những cơ chế suy diễn có sẵn (hợp nhất, quay lui và tìm kiếm theo chiều sâu) để tìm câu trả lời cho người dùng.

Prolog được sử dụng nhiều trong các ứng dụng của trí tuệ nhân tạo và ngôn ngữ máy tính (đặc biệt là trong ngành xử lý ngôn ngữ tự nhiên vì đây là mục tiêu thiết kế ban đầu của nó). Cú pháp và ngữ nghĩa của Prolog đơn giản và sáng sủa, nó được người Nhật coi là một trong những nền tảng để xây dựng máy tính thế hệ thứ năm mà ở đó, thay vì phải mô tả cách giải quyết một bài toán trên máy tính, con người chỉ cần mô tả bài toán và máy tính sẽ hỗ trợ họ nốt phần còn lại.

b. Cấu trúc cơ bản của một chương trình Prolog

Trong một chương trình Prolog ta thường có các yếu tố sau đây: Đối tượng, dữ kiện, sự kiện và luật.

Trong đó, đối tượng bao gồm hằng và biến, hằng mang giá trị cho sẵn ở đầu chương trình hoặc trong quá trình viết ta đưa vào, biến có giá trị thay đổi sẽ được gán giá trị khi chạy chương trình, tên biến là một ký tự hoa hoặc một chuỗi ký tự,

bắt đầu bằng một ký tự hoa, có một loại biến đặc biệt gọi là biến tự do, biến này không có tên và người ta dùng ký hiệu _ (dấu gạch dưới) thay cho tên biến.

Một chương trình Prolog bao gồm các luật được biểu diễn dưới dạng **mệnh đề Horn**. Một mệnh đề Horn có dạng: **Head:- Body**.

Trong đó: Head là một vị từ logic, còn Body có thể rỗng hoặc là một tập các vị từ logic.

Ví dụ:

```
child(X,Y) :- parent(Y,X).
```

Phần lớn các trình biên dịch dành cho các chương trình Prolog đều yêu cầu vị từ logic ở phần đầu của một mệnh đề Horn là một vị từ dương (không có dấu phủ định đi kèm), còn các vị từ trong phần Body có thể có dấu phủ định đi kèm. Chương trình logic mà không có sự xuất hiện của dấu phủ định đi kèm gọi là chương trình logic xác định, còn không thì được gọi là chương trình logic thông thường.

Những mệnh đề Horn mà phần Body rỗng được gọi là dữ kiện. Kiểu mệnh đề này thường được sử dụng để mô tả các dữ kiện của bài toán.

Ví dụ:

```
female('Princess Elsa').
```

Khi ta truy vấn dữ kiện:

```
(Princess Elsa có phải female không?)  
?- female('Princess Elsa').  
True.
```

```
(Trong dữ kiện đã cho, ai là female?)
```

```
?- female(X).  
X = 'Princess Elsa'.
```

Phần còn lại của các mệnh đề trong một chương trình Prolog được gọi là **luật (Rules)**. Nó thường thể hiện những phát biểu logic trong bài toán.

Ví dụ:

```
mother(X,Y) :- parent(X,Y), female(X).
```

Toán tử “:-” được dịch là “nếu”, trong logic thì nó đại diện cho toán tử “ \leftarrow ”. Phát biểu trên được phát biểu dưới dạng văn xuôi là “Nếu X là ‘parent’ của Y và X là ‘female’ thì X là ‘mother’ của Y”. Dấu phẩy “,” trong mệnh đề trên được dịch là toán tử “và”.

Trong một ví dụ khác:

```
parent(X,Y) :- mother(X,Y); father(X,Y).
```

Dấu “;” trong mệnh đề trên được dịch là toán tử “hoặc”. Mệnh đề trên có thể được phát biểu như sau: “X là ‘parent’ của Y nếu X là ‘mother’ hoặc ‘father’ của Y”.

Giống như nhiều ngôn ngữ lập trình khác, Prolog cũng có quy ước cho **chú thích (comments)**, cụ thể như sau:

```
+      %: Chú thích trên dòng
```

+ /*...*/: Chú thích trên một hoặc nhiều dòng

Prolog chủ yếu hỗ trợ xử lý các ký hiệu, tuy nhiên nó cũng hỗ trợ một số phép toán 2 ngôi chuẩn như sau:

Phép toán 2 ngôi	Ý nghĩa	Phép so sánh 2 ngôi	Ý nghĩa
+	Cộng	>	Lớn hơn
-	Trừ	>=	Lớn hơn hoặc bằng
*	Nhân	<	Bé hơn
/	Chia	<=	Bé hơn hoặc bằng
Mod	Chia lấy phần dư	<> hoặc ><	Khác
Div	Chia lấy phần nguyên	:=	Bằng

c. Ngữ nghĩa của chương trình Prolog

Một chương trình logic có ngữ nghĩa của riêng nó. Ngữ nghĩa quyết định những kết luận "đúng" nào có thể rút ra được từ một chương trình Prolog. Ví dụ một chương trình Prolog gồm một dữ kiện:

female("Princess Elsa").

Khi đó, ta có thể rút ra duy nhất một dữ kiện đúng là "Princess Elsa là 'female'".

Trong một ứng dụng Prolog, bạn có thể hỏi câu hỏi sau để có được trả lời "true":

?- female("Princess Elsa").

Mọi câu hỏi khác đều cho trả lời là "false". Điều này có nghĩa là trong một chương trình Prolog, mọi thứ bạn khai báo là đúng, nếu không thì nó là sai. Vì vậy trong ví dụ trên, khi bạn hỏi "Queen Ellizabeth II có phải là 'female'", bạn sẽ nhận được câu trả lời "false".

Với một chương trình Prolog xác định, ngữ nghĩa của nó được định nghĩa là ngữ nghĩa tối thiểu của nó. Với một chương trình Prolog thông thường, có nhiều loại ngữ nghĩa được sử dụng như ngữ nghĩa đầy đủ, ngữ nghĩa tối thiểu, ngữ nghĩa hoàn chỉnh,...

Đa số các chương trình biên dịch Prolog phổ thông (SWI-Prolog, GNU-Prolog) sử dụng ngữ nghĩa đầy đủ mà đi kèm là thủ tục suy diễn SLDNF.

d. Các kiểu dữ liệu của Prolog

Prolog có 2 dạng kiểu dữ liệu: Sơ cấp và có cấu trúc

- + Kiểu dữ liệu sơ cấp bao gồm: char, string, integer, unsigned, real
- + Kiểu dữ liệu có cấu trúc bao gồm tất cả các kiểu dữ liệu do người dùng tự định nghĩa dựa trên các kiểu dữ liệu sơ cấp được cung cấp sẵn nêu trên.

Việc khai báo kiểu dữ liệu tùy thuộc vào trình biên dịch mà bạn sử dụng.

e. Ứng dụng của Prolog

Như đã biết ở phần 1. Sơ lược về Prolog, Prolog được ứng dụng nhiều nhất trong trí tuệ nhân tạo và xử lý ngôn ngữ tự nhiên. Cụ thể, Prolog hỗ trợ để máy có thể suy luận và đưa ra đáp án khi có những dữ liệu cần thiết.

Tên User	Giới tính	Đặc điểm		Tiêu chuẩn kết bạn		
An	Nữ	Chăm chỉ	Giỏi	Giàu	Tốt bụng	Đẹp
Lam	Nữ	Tốt bụng	Khoẻ mạnh	Khoẻ mạnh	Đẹp	Giỏi
Chi	Nữ	Đẹp	Giàu	Giàu	Khoẻ mạnh	Giỏi
Huân	Nam	Khoẻ mạnh	Giàu	Giàu	Chăm chỉ	Đẹp
Liêm	Nam	Giỏi	Chăm chỉ	Đẹp	Tốt bụng	Giỏi
Quý	Nam	Đẹp	Tốt bụng	Khoẻ mạnh	Tốt bụng	Chăm ch

Xét một ví dụ, ta có một website (hoặc ứng dụng) tìm kiếm bạn bè theo quy chuẩn được đưa ra bởi các user (khách hàng) sử dụng dịch vụ của trang web đó. Ta có bảng dữ liệu đầu vào (dữ liệu do các user đưa ra) như sau:

Với yêu cầu kết bạn là:

+ Hai người khác phái

+ Mỗi người phải đáp ứng được đúng 2 trong 3 yêu cầu mà người còn lại đưa ra.

Bằng việc nhập vào các dữ kiện nêu trên và đưa ra các luật, ta dễ dàng tìm được các cặp có thể kết bạn với nhau (yêu cầu ban đầu).

3.1.2 Yêu cầu 2: *Tìm hiểu môi trường lập trình Prolog. Cách thức triển khai ngôn ngữ Prolog trên công cụ đã tìm hiểu.*

1.1 Giới thiệu chung

SWI-Prolog là một chương trình cung cấp **môi trường thực thi** cho ngôn ngữ **Prolog**. Có rất nhiều và đầy đủ các thư viện cung cấp các **logic ràng buộc, đa nhiệm, unit testing, GUI**(Giao diện đồ họa), và cái API khác cho **Java, ODBC, Web server, ...** Với lượng document (tài liệu) rộng lớn.

SWI-Prolog miễn phí, có thể chạy được trên các nhóm hệ điều hành Windows, Linux và Unix (MacOS).

SWI-Prolog được phát triển từ năm 1987. Tác giả phát triển chính là **Jan Wielemaker**.

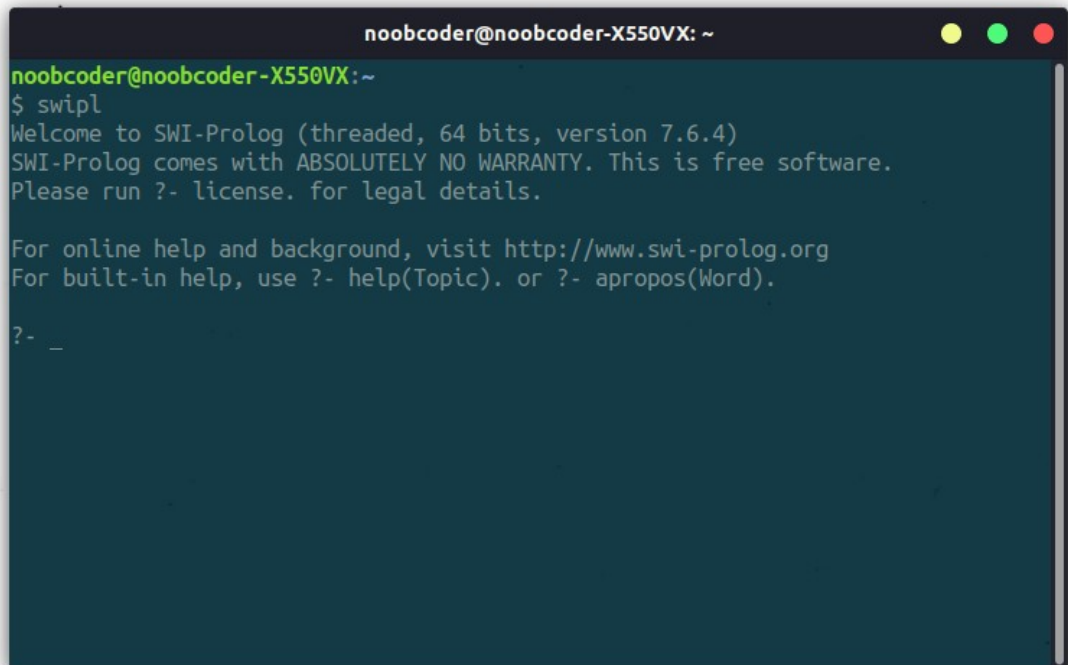
Các thông tin khác:

- **Viết trên:** C, Prolog
- **OS:** Cross-platform
- **Loại:** Logic programming
- **Bản quyền:** Simplified BSD, LGPL prior to version 7.3.33
- **Website:** swi-prolog.org

1.2 Cách hoạt động cơ bản

Bắt đầu chương trình

SWI-Prolog là chương trình hoạt động trên giao diện **command line**. Tương tác trực tiếp với người dùng thông qua command line. Người dùng có thể trực tiếp tạo các facts và rules, và đồng thời truy vấn các Goal đã được định nghĩa sẵn khi chương trình bắt đầu.

A screenshot of a terminal window with a dark background. The window title is 'noobcoder@noobcoder-X550VX: ~'. The prompt is 'noobcoder@noobcoder-X550VX:~'. The user enters '\$ swipl'. The output shows the SWI-Prolog welcome message: 'Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)', a disclaimer 'SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software. Please run ?- license. for legal details.', and instructions for online help and built-in help. The prompt changes to '?- _'.

CommandLineInterface

Người dùng có thể nạp tri thức (rules & facts) từ file 'pl' lên theo cú pháp:

```
?- consult('tenfile.pl').
```

% hoặc nếu tên file viết liền

```
?- consult(tenfile).
```

% hoặc

```
?- ['tenfile'].
```

% hoặc

```
?- [tenfile].
```

Lưu ý: câu lệnh prolog kết thúc bằng dấu .

Truy vấn

Lấy một ví dụ: Ta có một bộ tri thức sau trong file `family.pl`

```
% family.pl
```

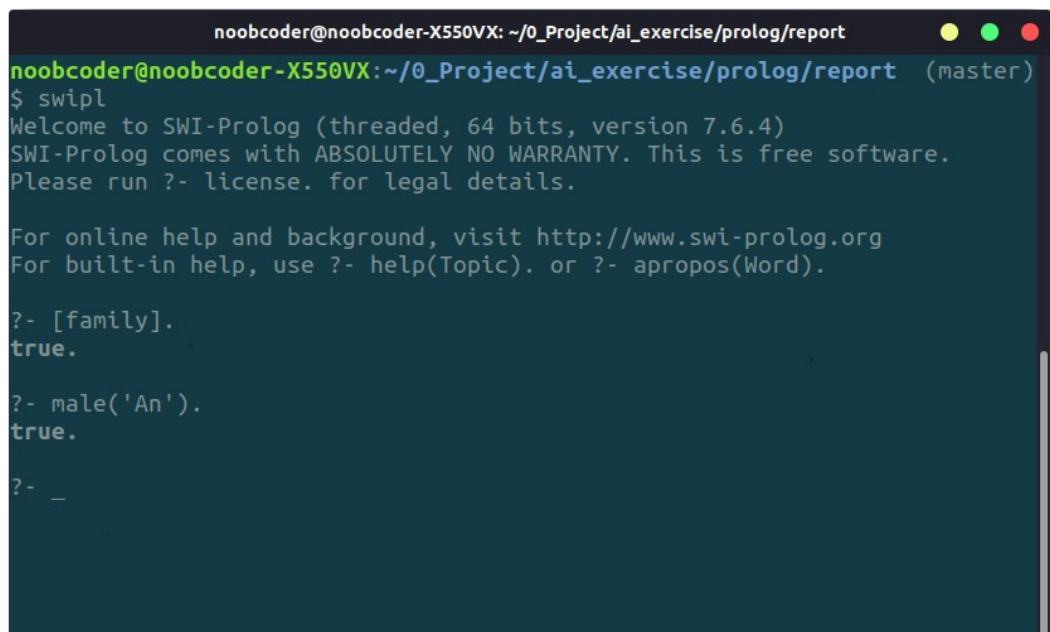
```
male('An').
male('Khang').
male('Dang').
female('Linh').
female('Chau').

parent('Linh', 'An').
parent('Khang', 'An').

friendOf('An', 'Dang').
friendOf('Chau', 'Dang').
friendOf('An', 'Chau').

friend(A, B):- friendOf(A, B), friendOf(B, A).
father(F, C):- parent(F, C), male(F).
mother(M, C):- parent(M, C), female(M).
```

- Ta có thể truy vấn các facts:
An có phải giới tính nam hay không?



```
noobcoder@noobcoder-X550VX: ~/0_Project/ai_exercise/prolog/report
noobcoder@noobcoder-X550VX:~/0_Project/ai_exercise/prolog/report (master)
$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
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).

?- [family].
true.

?- male('An').
true.

?- _
```

Kết quả: Phải!

- Truy vấn các rules:
Khang có phải cha của An hay không?
Dang có phải cha của An không?

```
noobcoder@noobcoder-X550VX: ~/0_Project/ai_exercise/prolog/report
$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
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).

?- [family].
true.

?- male('An').
true.

?- father('Khang', 'An').
true.

?- father('Dang', 'An').
false.

?- _
```

rules-query

- Để truy vấn các nhiều kết quả theo rules hoặc facts, ta để biến nào đó X (Y , Z ...) vào vị trí mà ta cần truy vấn. Khi truy vấn, để xem kết quả tiếp theo, ta ấn phím ; (Ấn phím . sẽ không hiện thêm kết quả tiếp theo).

Liệt kê những ai có giới tính nam

```
noobcoder@noobcoder-X550VX: ~/0_Project/ai_exercise/prolog/report
For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- [family].
true.

?- male('An').
true.

?- father('Khang', 'An').
true.

?- father('Dang', 'An').
false.

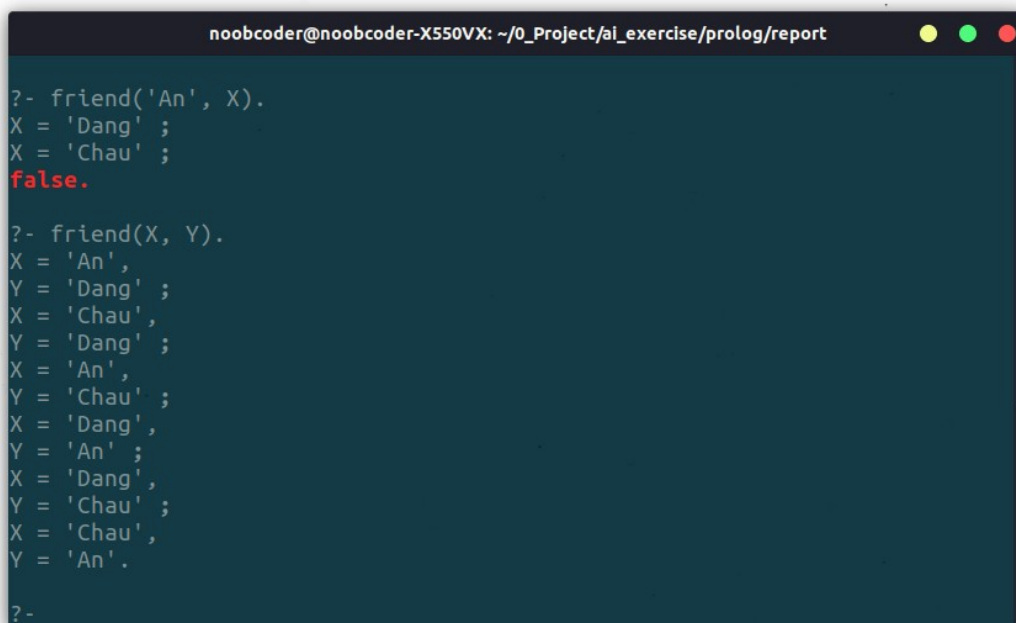
?- male(X).
X = 'An' ;
X = 'Khang' ;
X = 'Dang'.

?- _
```

Kết quả: An, Khang, Dang.

Liệt kê bạn của An

Liệt kê các cặp bạn bè của nhau



```
noobcoder@noobcoder-X550VX: ~/0_Project/ai_exercise/prolog/report
?- friend('An', X).
X = 'Dang' ;
X = 'Chau' ;
false.

?- friend(X, Y).
X = 'An',
Y = 'Dang' ;
X = 'Chau',
Y = 'Dang' ;
X = 'An',
Y = 'Chau' ;
X = 'Dang',
Y = 'An' ;
X = 'Dang',
Y = 'Chau' ;
X = 'Chau',
Y = 'An'.
?-
```

friend-query

Kết quả:

- Dang, Chau
- (An, Dang), (An, Chau), (Chau, Dang), (Chau, An), ...

Thoát

Ta có thể gõ lệnh `halt.` hoặc tổ hợp phím `Ctrl+D` để thoát SWI-Prolog.

1.3 Một số hàm hỗ trợ khác sử dụng trong đồ án

Thao tác với file

SWI-Prolog cũng như Prolog cũng có thể thao tác đọc / ghi file.

?-

```
open('filename.txt', write, Stream),
writeln(Stream, 'Hello World'),
close(Stream).
```

Ta sẽ có ở file `filename.txt`

Hello World

[Xem chi tiết](#)

Chạy file có sẵn

Ta có thể chạy các truy vấn trong file `.pl` bằng câu lệnh:

```
swipl -s filename.pl
```

Giả sử ta có file `truyvan.pl`:

```
% truyvan.pl
```

```
% BAT DAU SCRIPT
```

```
:- consult('family'). % load tri thức từ file
```

family.pl

% nếu không có syntax :-, đoạn script sẽ xem như tri thức và không chạy

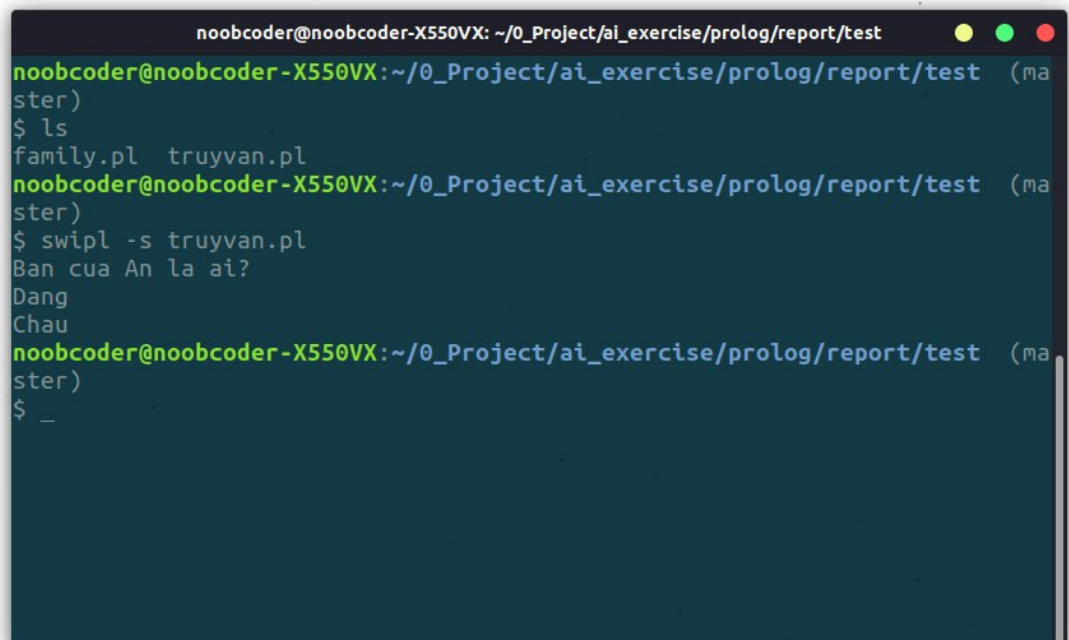
:-

```
writeln('Ban cua An la ai?'),  
forall( % ham ho tro liet ke tat ca ket qua  
        friend(X, 'An'),  
        writeln(X)  
    ).
```

:- halt.

% Nếu không có halt, chương trình SWI-Prolog sẽ chưa được thoát và trở lại giao diện tương tác.

Chạy script: swipl -s truyvan.pl

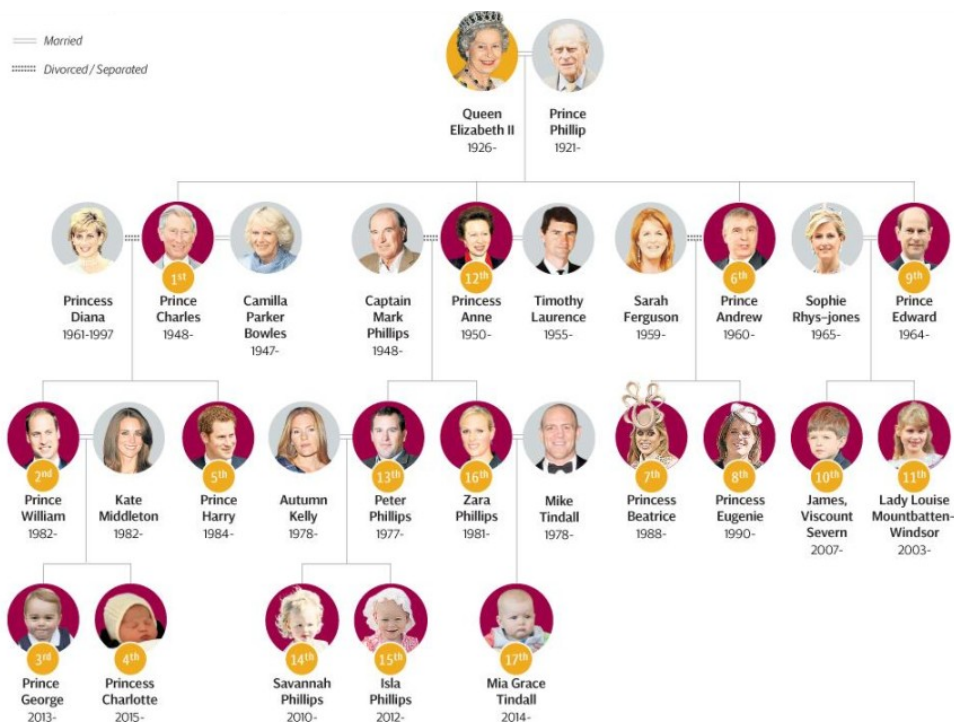


```
noobcoder@noobcoder-X550VX: ~/0_Project/ai_exercise/prolog/report/test  
noobcoder@noobcoder-X550VX:~/0_Project/ai_exercise/prolog/report/test (ma  
ster)  
$ ls  
family.pl  truyvan.pl  
noobcoder@noobcoder-X550VX:~/0_Project/ai_exercise/prolog/report/test (ma  
ster)  
$ swipl -s truyvan.pl  
Ban cua An la ai?  
Dang  
Chau  
noobcoder@noobcoder-X550VX:~/0_Project/ai_exercise/prolog/report/test (ma  
ster)  
$ _
```

Chạy Script

[Xem chi tiết](#)

3.1.3 Yêu cầu 2: Giải quyết bài toán suy diễn bằng ngôn ngữ Prolog trên công cụ SWI-Prolog. Xây dựng cây phả hệ chôn giá đình Hoàng gia Anh như minh họa ở hình bên dưới.



Lược đồ phả hệ Hoàng Gia Anh.

Các vị từ có trong cơ sở tri thức:

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

Xét theo ngữ cảnh phương Tây, quan hệ parent xét cho cha mẹ với con cái ruột, không tính tới con rể, con dâu. Ngoài ra có các quan hệ có tính đối xứng như married, divorced, brother, sister,... đều được trình bày cả hai chiều quan hệ, ví dụ married(A, B) thì sẽ bao gồm 2 fact A kết hôn với B và B kết hôn với A.

- Bảng liệt kê các quan hệ được viết trong file prolog-script/cay-pha-he/tri-thuc.pl

```
%Married
married('Queen Elizabeth II', 'Prince Phillip').
married('Prince Phillip', 'Queen Elizabeth II').

married('Prince Charles', 'Camilla Parker Bowles').
married('Camilla Parker Bowles', 'Prince Charles').

married('Princess Anne', 'Timothy Laurence').
married('Timothy Laurence', 'Princess Anne').

married('Sophie Rhys-jones', 'Prince Edward').
married('Prince Edward', 'Sophie Rhys-jones').

married('Prince William', 'Kate Middleton').
married('Kate Middleton', 'Prince William').

married('Autumn Kelly', 'Peter Phillips').
married('Peter Phillips', 'Autumn Kelly').

married('Zara Phillips', 'Mike Tindall').
married('Mike Tindall', 'Zara Phillips').

%Divorced
divorced('Princess Diana', 'Prince Charles').
divorced('Prince Charles', 'Princess Diana').

divorced('Captain Mark Phillips', 'Princess Anne').
divorced('Princess Anne', 'Captain Mark Phillips').

divorced('Sarah Ferguson', 'Prince Andrew').
divorced('Prince Andrew', 'Sarah Ferguson').

%Parent
parent('Queen Elizabeth II', 'Prince Charles').
parent('Queen Elizabeth II', 'Princess Anne').
parent('Queen Elizabeth II', 'Prince Andrew').
parent('Queen Elizabeth II', 'Prince Edward').
parent('Prince Phillip', 'Prince Charles').
parent('Prince Phillip', 'Princess Anne').
parent('Prince Phillip', 'Prince Andrew').
parent('Prince Phillip', 'Prince Edward').

parent('Princess Diana', 'Prince William').
parent('Princess Diana', 'Prince Harry').
```



```
parent('Prince Charles', 'Prince William').
parent('Prince Charles', 'Prince Harry').

parent('Captain Mark Phillips', 'Peter Phillips').
parent('Captain Mark Phillips', 'Zara Phillips').
parent('Princess Anne', 'Peter Phillips').
parent('Princess Anne', 'Zara Phillips').

parent('Sarah Ferguson', 'Princess Beatrice').
parent('Sarah Ferguson', 'Princess Beatrice').
parent('Prince Andrew', 'Princess Eugenie').
parent('Prince Andrew', 'Princess Eugenie').

parent('Sophie Rhys-jones', 'James Viscount Severn').
parent('Sophie Rhys-jones', 'Lady Louise Mountbatten-Windsor').
parent('Prince Edward', 'James Viscount Severn').
parent('Prince Edward', 'Lady Louise Mountbatten-Windsor').

parent('Prince William', 'Prince George').
parent('Prince William', 'Princess Charlotte').
parent('Kate Middleton', 'Prince George').
parent('Kate Middleton', 'Princess Charlotte').

parent('Autumn Kelly', 'Savannah Phillips').
parent('Autumn Kelly', 'Isla Phillips').
parent('Peter Phillips', 'Savannah Phillips').
parent('Peter Phillips', 'Isla Phillips').

parent('Zara Phillips', 'Mia Grace Tindall').
parent('Mike Tindall', 'Mia Grace Tindall').

%male
male('Prince Phillip').
male('Prince Charles').
male('Captain Mark Phillips').
male('Timothy Laurence').
male('Prince Andrew').
male('Prince Edward').
male('Prince William').
male('Prince Harry').
male('Peter Phillips').
male('Mike Tindall').
male('James Viscount Severn').
male('Prince George').
```

```
%female
female('Queen Elizabeth II').
female('Prince Diana').
female('Camilla Parker Bowles').
female('Princess Anne').
female('Sarah Ferguson').
female('Sophie Rhys-jones').
female('Kate Middleton').
female('Autumn Kelly').
female('Zara Phillips').
female('Princess Beatrice').
female('Princess Eugenie').
female('Lady Louise Mountbatten-Windsor').
female('Princess Charlotte').
female('Savannah Phillips').
female('Isla Phillips').
female('Mia Grace Tindall').
```

- Xây dựng các vị từ:

Vị từ	Giải nghĩa
<code>husband(Person, Wife) :-</code> <code>male(Person), female(Wife),</code> <code>married(Person, Wife),</code> <code>not(divorced(Person, Wife)).</code>	Một Person là husband của Wife thỏa mã Person cưới Wife và Person là đàn ông, Wife là nữ và họ chưa ly dị nhau.
<code>wife(Person, Husband) :-</code> <code>female(Person), male(Husband),</code> <code>married(Person, Husband),</code> <code>not(divorced(Person, Husband)).</code>	Một Person là wife của Husband thỏa mã Person cưới Husband và Person là phụ nữ, Husband là đàn ông và họ chưa ly dị nhau.
<code>father(Parent, Child):- parent(Parent,</code> <code>Child), male(Parent).</code>	Một Parent là father của Child nếu Parent là cha mẹ của Child và Parent có giới tính là nam.
<code>mother(Parent, Child):- parent(Parent,</code> <code>Child), female(Parent).</code>	Một Parent là mother của Child nếu Parent là cha mẹ của Child và Parent có giới tính là nữ.
<code>child(Child, Parent):- parent(Parent,</code> <code>Child).</code>	Một Child là child của Parent nếu Parent thỏa mãn là cha mẹ của Child (ngược lại so với vị từ parent).
<code>son(Child, Parent) :- parent(Parent,</code> <code>Child), male(Child).</code>	Một Child là son của Parent nếu Child là con (child) của Parent và có giới tính là nam.

<code>daughter(Child, Parent) :- parent(Parent, Child), female(Child).</code>	Một Child là daughter của Parent nếu Child là con (child) của Parent và có giới tính là nữ.
<code>grandparent(GP, GC) :- parent(GP, X), parent(X, GC).</code>	Một GP là grandparent của GC nếu GC là con của X và X là con của GP.
<code>grandmother(GP, GC) :- grandparent(GP, GC), female(GP).</code>	Một GP là grandmother của GC nếu GP là grandparent của GC và là nữ.
<code>grandfather(GP, GC) :- grandparent(GP, GC), male(GP).</code>	Một GP là grandfather của GC nếu GP là grandparent của GC và là nam.
<code>grandchild(GC, GP) :- grandparent(GP, GC).</code>	Một GC là grandchild của GP nếu GP là ông bà (grandparent) của GC.
<code>grandson(GC, GP) :- grandchild(GC, GP), male(GC).</code>	GC là grandson khi GC là grandchild và là nam.
<code>granddaughter(GC, GP) :- grandchild(GC, GP), female(GC).</code>	GC là granddaughter khi GC là grandchild và là nữ.
<code>sibling(X, Y) :- mother(Mother, X), mother(Mother, Y), father(Father, X), father(Father, Y), X \= Y.</code>	Sibling (anh/chị/em ruột) là khi 2 người cùng cha cùng mẹ, và tất nhiên hai người phải khác nhau.
<code>brother(X, Y) :- sibling(X, Y), male(X). sister(X, Y) :- sibling(X, Y), female(X).</code>	Sibling là nam thì gọi là brother, là nữ thì gọi là sister.
<code>aunt(Person, NieceNephew) :- parent(X, NieceNephew), sibling(X, Person), female(Person). uncle(Person, NieceNephew) :- parent(X, NieceNephew), sibling(X, Person), male(Person).</code>	Cô/dì/chú/bác Person là anh em ruột của X, là cha mẹ của Person. Cô/dì là nữ, chú/bác là nam.
<code>niece(Person, AuntUncle) :- aunt(AuntUncle, Person), female(Person); uncle(AuntUncle, Person), female(Person). nephew(Person, AuntUncle) :- aunt(AuntUncle, Person), male(Person); uncle(AuntUncle, Person), male(Person).</code>	Person là cháu của cô/dì/chú/bác AuntUncle nếu AuntUncle là Cô/dì hoặc Chú/Bác của Person. Nếu nữ thì gọi là niece, nếu nam thì gọi là Nephew.

--	--

- Bộ các câu hỏi để hỏi hệ tri thức vừa xây dựng:

01. Ai là mẹ của Prince Andrew? (mother)
02. Ai là vợ của Prince Charles? (wife)
03. Camilla có phải là mẹ của Prince William không? (mother)
04. Princess Diana có phải vợ của Prince Edward không? (wife)
05. Liệt kê cháu gái của Queen Elizabeth II? (granddaughter)
06. Liệt kê con trai của Prince Phillip? (son)
07. Prince William có là anh em của Prince Harry không? (sibling)
08. Prince Andrew có phải là cậu/bác của Peter Phillips? (uncle)
09. Kate Middleton có phải là Dì của Isla Phillips? (aunt)
10. Sarah Ferguson có đang là vợ của Prince Andrew? (wife)
11. Captain Mark Phillips có những đứa cháu là ai? (grandchild)
12. Ai là ông nội của Savannah Phillips? (grandfather)
13. Cha của James Viscount Severn là ai? (father)
14. Em gái/chị của Peter Phillips là ai? (sister)
15. Anh/em trai của Prince Andrew là ai? (brother)
16. Ông/bà của Princess Charlotte gồm những ai? (grandparent)
17. Mia Grace Tindall có phải là con trai? (male)
18. Vợ hiện tại Prince Andrew là ai? (wife)
19. Ai chưa có gia đình? (not married)
20. Bà mẹ nào có đúng 2 con? (female, children).

- Bộ test được viết thành script trong file `prolog-script/cay-pha-he/test-script.pl`

Chạy test bằng câu lệnh:

```
swipl -s test-script.pl
```

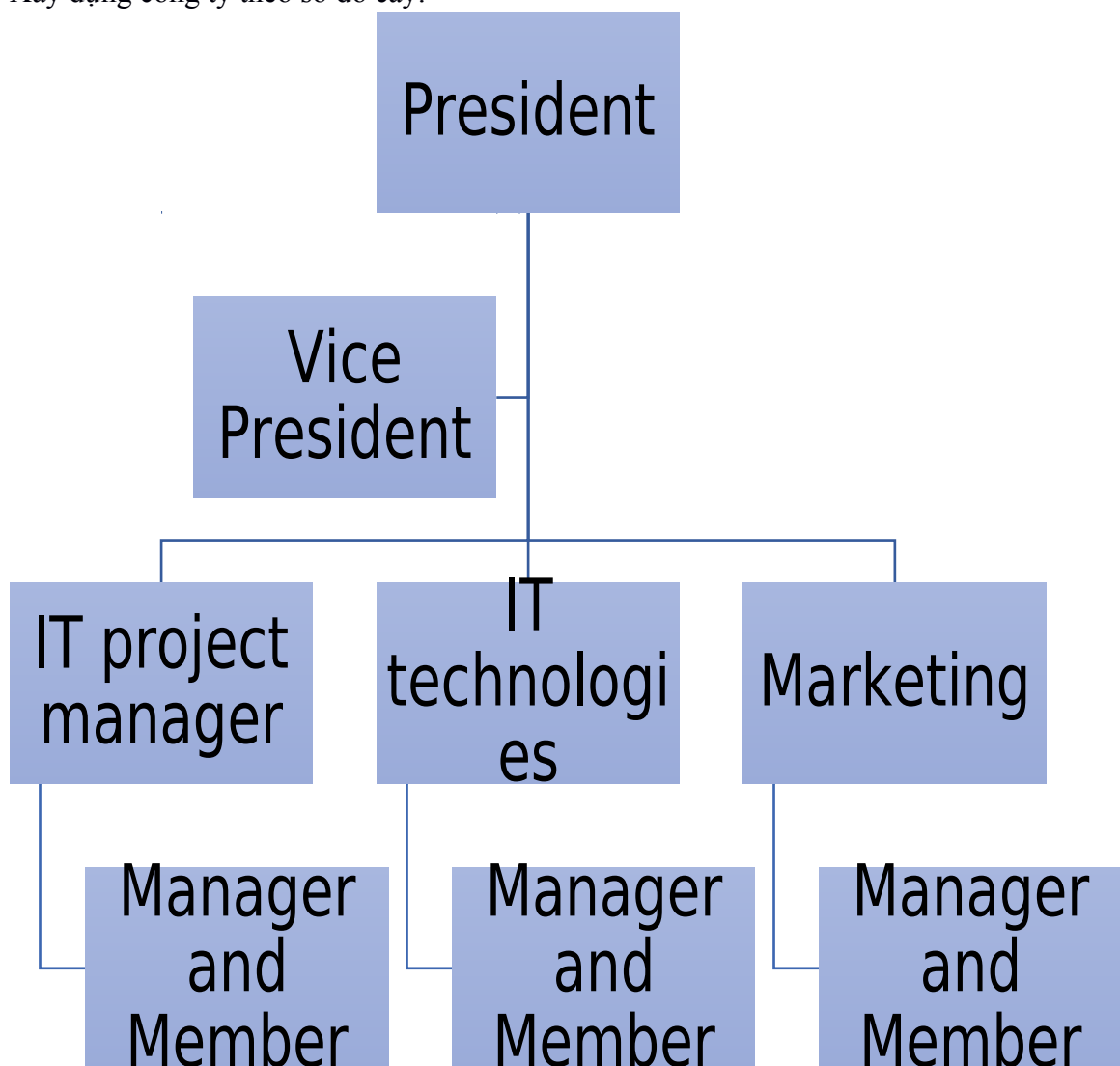
Câu trả lời sẽ được lưu vào file `test-output.txt`

```
01: Queen Elizabeth II
02: Camilla Parker Bowles
03: false
04: false
05: Zara Phillips, Princess Eugenie, Princess Eugenie, Lady Louise
Mountbatten-Windsor,
06: Prince Charles, Prince Andrew, Prince Edward,
07: false
08: true
09: false
10: false
11: Savannah Phillips, Isla Phillips, Mia Grace Tindall,
12: Captain Mark Phillips
13: Prince Edward
```

```
14: Zara Phillips,  
15: Prince Charles, Prince Edward,  
16: Princess Diana, Prince Charles,  
17: false  
18:  
19: Prince Harry, James Viscount Severn, Prince George, Prince Diana,  
Princess Beatrice, Princess Eugenie, Lady Louise Mountbatten-Windsor,  
Princess Charlotte, Savannah Phillips, Isla Phillips, Mia Grace  
Tindall,  
20: Princess Anne, Sarah Ferguson, Sophie Rhys-jones, Kate Middleton,  
Autumn Kelly,
```

3.2 Xây dựng cơ sở tri thức với công cụ Prolog

- Xây dựng công ty theo sơ đồ cây:



- Các vị từ của Sơ đồ:

%gender member:

male('Alex').

male('David').

male('John').

male('Mason').

male('Buck').

male('Caulder').

male('Burn').

male('Kane').

male('Mark').

male('William').

male('Todd').

male('Billy').

male('Jacob').

male('James').

male('Daniel').

male('Adam').

male('Riley').

male('Henry').

male('Max').

male('Edward').

female('Jean').

female('Jenny').

female('Grace').

female('Arica').

female('Chloe').

female('Victoria').

female('Camila').

female('Layla').

female('Mia').

female('Sofia').

female('Avery').

female('Amelia').

%company

company('Kagat').

company('Mining').

company('Vining').

```
%department
department('IT Technologies').
department('IT Project manager').
department('Marketing').

% CEO
president('Alex').
vicePresident('David').

% manager
manager('John').
manager('William').
manager('Arica').

% members:
member('John','IT Technologies').
member('Mason','IT Technologies').
member('Buck','IT Technologies').
member('Caulder','IT Technologies').
member('Burn','IT Technologies').
member('Kane','IT Technologies').
member('Mark','IT Technologies').
member('Jean','IT Technologies').
member('Jenny','IT Technologies').
member('Grace','IT Technologies').

member('William','IT Project manager').
member('Todd','IT Project manager').
member('Billy','IT Project manager').
member('Jacob','IT Project manager').
member('James','IT Project manager').
member('Daniel','IT Project manager').
member('Adam','IT Project manager').
member('Chloe','IT Project manager').
member('Sofia','IT Project manager').
member('Layla','IT Project manager').

member('Arica','Marketing').
member('Victoria','Marketing').
```

```
member('Camila','Marketing').  
member('Mia','Marketing').  
member('Avery','Marketing').  
member('Amelia','Marketing').  
member('Riley','Marketing').  
member('Henry','Marketing').  
member('Max','Marketing').  
member('Edward','Marketing').
```

```
member('IT Technologies','John').  
member('IT Technologies','Mason').  
member('IT Technologies','Buck').  
member('IT Technologies','Caulder').  
member('IT Technologies','Burn').  
member('IT Technologies','Kane').  
member('IT Technologies','Mark').  
member('IT Technologies','Jean').  
member('IT Technologies','Jenny').  
member('IT Technologies','Grace').
```

```
member('IT Project manager','William').  
member('IT Project manager','Todd').  
member('IT Project manager','Billy').  
member('IT Project manager','Jacob').  
member('IT Project manager','James').  
member('IT Project manager','Daniel').  
member('IT Project manager','Adam').  
member('IT Project manager','Chloe').  
member('IT Project manager','Sofia').  
member('IT Project manager','Layla').
```

```
member('Marketing','Arica').  
member('Marketing','Victoria').  
member('Marketing','Camila').  
member('Marketing','Mia').  
member('Marketing','Avery').  
member('Marketing','Amelia').  
member('Marketing','Riley').  
member('Marketing','Henry').  
member('Marketing','Max').
```



```
member('Marketing','Edward').

presidentComp('Alex','Kagat').
presidentComp('Kagat','Alex').

vicePresidentComp('David','Kagat').
vicePresidentComp('Kagat','David').

%Company co-operation:
co_op('Kagat','Mining').
co_op('Mining','Kagat').

co_op('Kagat','Vining').
co_op('Vining','Kagat').

%Department of company
departmentOfCompany('Kagat','IT Project manager').
departmentOfCompany('Kagat','IT Technologies').
departmentOfCompany('Kagat','Marketing').

departmentOfCompany('IT Project manager','Kagat').
departmentOfCompany('IT Technologies','Kagat').
departmentOfCompany('Marketing','Kagat').

%Relationship:
memberCompany(X,Y) :- departmentOfCompany(X,Z),member(Y,Z).

companyMember(X,Y) :- departmentOfCompany(Y,Z),member(X,Z).

departmentCompany(X,Y) :- departmentCompany(X,Y).

memberDepartMent(X,Y) :- member(X,Y).

managerDepartment(X,Y) :- member(X,Y),manager(X).

managerToEmployees(X,Y) :- member(X,Z),member(Y,Z),manager(X).

presidentToEmployees(X,Y) :-
presidentComp(X,Z),memberCompany(Y,Z),president(X).
```

<p>vicePresidentToEmployees(X,Y) :- vicePresidentComp(X,Z),memberCompany(Y,Z),vicePresident(X).</p> <p>presidentToManager(X,Y) :- presidentComp(X,Z),memberCompany(Y,Z),president(X),manager(Y).</p> <p>vicePresidentToManager(X,Y) :- vicePresidentComp(X,Z),memberCompany(Y,Z),vicePresident(X),manager(Y).</p> <p>cooperation(X,Y) :- co_op(X,Y).</p> <p>companyPresident(X,Y) :- presidentComp(X,Y).</p> <p>companyVicePresident(X,Y) :- vicePresidentComp(X,Y).</p>

- Cài đặt các mối quan hệ:

memberCompany(X,Y) :- departmentOfCompany(X,Z),member(Y,Z).	Người X là nhân viên của công ty Y khi X thuộc về phòng ban Z và phòng Z thuộc công ty X.
departmentCompany(X,Y) :- departmentCompany(X,Y).	Phòng X thuộc về công ty Y (hoặc ngược lại).
memberDepartMent(X,Y) :- member(X,Y).	Nhân viên X có thuộc về phòng Y (hoặc ngược lại).
managerDepartment(X,Y) :- member(X,Y),manager(X).	Nhân viên X là trưởng phòng Y khi nhân viên X thuộc phòng Y và X giữ chức trưởng phòng.
managerToEmployees(X,Y) :- member(X,Z),member(Y,Z),manager(X).	X là trưởng phòng Y khi X và Y cùng phòng và X là trưởng phòng.
presidentToEmployees(X,Y) :- presidentComp(X,Z),memberCompany(Y,Z),president(X).	X là giám đốc Y khi X và Y cùng thuộc công ty và X giữ chức giám đốc.
presidentToManager(X,Y) :- presidentComp(X,Z),memberCompany(Y,Z),president(X),manager(Y).	X là giám đốc của trưởng phòng Y khi X và Y cùng thuộc công ty và X giữ chức giám đốc, Y giữ chức trưởng phòng.
cooperation(X,Y) :- co_op(X,Y).	Công ty X và Y hợp tác.
companyPresident(X,Y) :- presidentComp(X,Y).	X là giám đốc của công ty Y.

- Bộ câu hỏi và test được ghi trong file *congty-test.pl*

- Việc test làm giống như trên bài tập cây phả hệ:

swipl -s congty-test.pl

- Kết quả: **Lỗi một số bộ test và một số test không có kết quả**

3.3 Cài đặt hệ thống suy diễn bằng Ngôn ngữ lập trình

Ngôn ngữ sử dụng: Python

Mã nguồn tham khảo: <http://www.openbookproject.net/py4fun/prolog/prolog3.html>

Kết quả: chỉ có mã nguồn vài một vài test lỗi, nhóm chưa có báo cáo cụ thể về bài tập này

Thực thi:

4 TÀI LIỆU THAM KHẢO

- <http://www.cs.toronto.edu/~sheila/324/f05/tuts/swi.pdf>
- <https://en.wikipedia.org/wiki/SWI-Prolog>
- http://www.swi-prolog.org/pldoc/doc_for?object=manual
- <http://radio-weblogs.com/0106756/gems/391report.pdf>
- <http://www.openbookproject.net/py4fun/prolog/prolog3.html>

--Hết--