



**ĐẠI HỌC QUỐC GIA TP.HCM**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**CƠ SỞ TRÍ TUỆ NHÂN TẠO**  
**PROJECT 02**

Sinh viên thực hiện : 21127043 - Lư Trung Hậu

21127083 - Hoàng Đức Kiên

21127285 - Phan Thanh Hoàng

21127553 - Lê Gia Quốc Tỉ

Lớp : 21CLC03

Giáo viên hướng dẫn: Nguyễn Trần Duy Minh  
Lê Ngọc Thành  
Nguyễn Ngọc Thảo  
Nguyễn Hải Đăng

## Mục lục

|  |    |
|--|----|
| <b>I. Thông tin nhóm:</b> .....  | 3  |
| <b>II. Phân công nhiệm vụ:</b> .....   | 3  |
| <b>III. Kết quả tổng quan:</b> .....   | 3  |
| <b>1.1 Làm việc với công cụ Prolog</b> .....                                 | 3  |
| <b>1.2 Xây dựng cơ sở tri thức với Prolog</b> .....                          | 14 |
| <b>1.3 Thực hiện hệ thống suy luận logic trong ngôn ngữ lập trình.</b> ..... | 19 |
| <b>IV. Nguồn tham khảo</b> .....   | 22 |

## **I. Thông tin nhóm:**

- + Lư Trung Hậu – 21127043 – 21CLC03
- + Hoàng Đức Kiên – 21127083 – 21CLC03
- + Phan Thanh Hoàng – 21127285 – 21CLC03
- + Lê Gia Quốc Tỉ - 21127553 – 21CLC03

## **II. Phân công nhiệm vụ:**

| STT | Họ và Tên        | MSSV     | Nhiệm vụ  | Mức độ hoàn thành |
|-----|------------------|----------|---|-------------------|
| 1   | Lư Trung Hậu     | 21127043 | Thực hiện các yêu cầu mục 1.1 trong file Project02.pdf và viết báo cáo                  | 100%              |
| 2   | Hoàng Đức Kiên   | 21127083 | Cài đặt chương trình prolog mục 1.1 trong file Project02.pdf và viết báo cáo            | 100%              |
| 3   | Phan Thanh Hoàng | 21127285 | Làm mục 1.3 trong file Project02.pdf  | 100%              |
| 4   | Lê Gia Quốc Tỉ   | 21127553 | Làm mục 1.2 trong file Project02.pdf<br>Làm câu hỏi 1 phần 1.1 trong file Project02.pdf | 100%              |

## **III. Kết quả tổng quan:**

- Các thành viên hoàn thành tốt công việc được giao, hoàn thành các hạng mục được yêu cầu, có sự nhất quán với nhau trong công việc.
- Hoàn thành 100% project được giao.

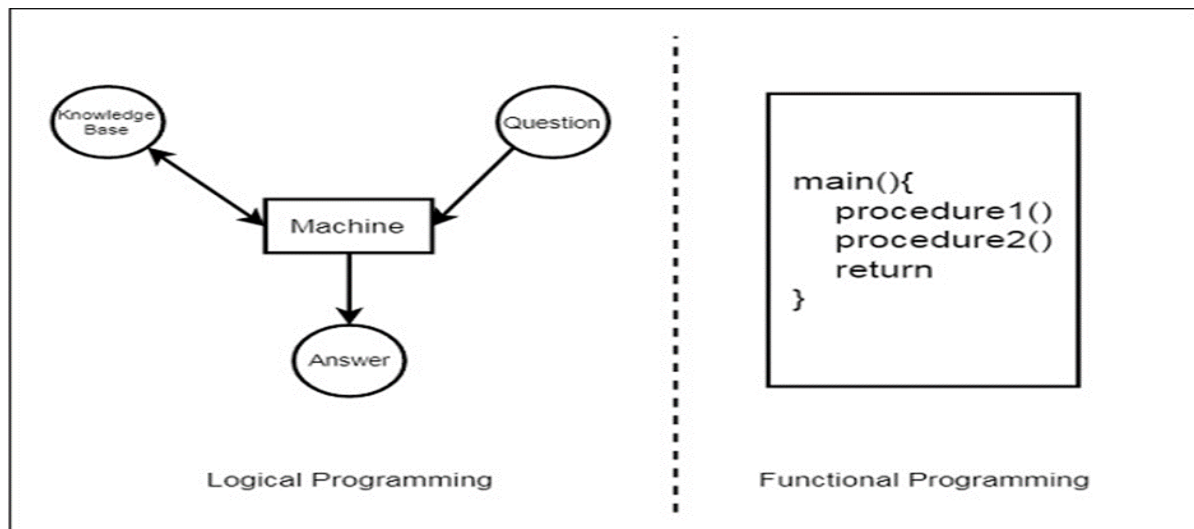
### **1.1 Làm việc với công cụ Prolog**

#### **1.1.1**

## I. Những nét riêng biệt của PROLOG

- PROLOG lập trình logic sẽ dựa trên cơ sở tri thức và câu hỏi để đưa ra câu trả lời
- Cơ sở tri thức bao gồm: sự thật và các mối quan hệ giữa các đối tượng
  - Là ngôn ngữ lập trình khai báo.  
⇒ Giải thích: Các sự thật và mối quan hệ cần được tìm hiểu sẽ được mã hóa bằng cách làm cho thông tin này trở thành một phần của cơ sở tri thức của hệ thống. Sau đó, người dùng có thể truy vấn trong hệ thống để biết thông tin được nêu rõ ràng trong cơ sở tri thức hoặc được ngụ ý từ thông tin này.
  - Thuật ngữ tính toán vị từ  
⇒ Giải thích: Trong phép tính vị từ, các đối tượng được biểu diễn bằng các thuật ngữ có thể một trong các dạng sau:
    - a/ Một biểu tượng hằng
    - b/ Một biểu tượng biến
    - c/ Một thuật ngữ hợp chất
  - Trình xử lý danh sách và truy hồi tự nhiên
  - Có công cụ suy luận tích hợp, cũng như là quay lui tự động để giải quyết vấn đề suy luận logic.

⇒ Khả năng đọc hiểu vào code dễ hiểu hơn, dễ dàng đọc hiểu bởi vì ngôn ngữ dùng để miêu tả sự thật, mối quan hệ, nó hướng dẫn người đọc phải làm gì hoặc là không phải làm gì.

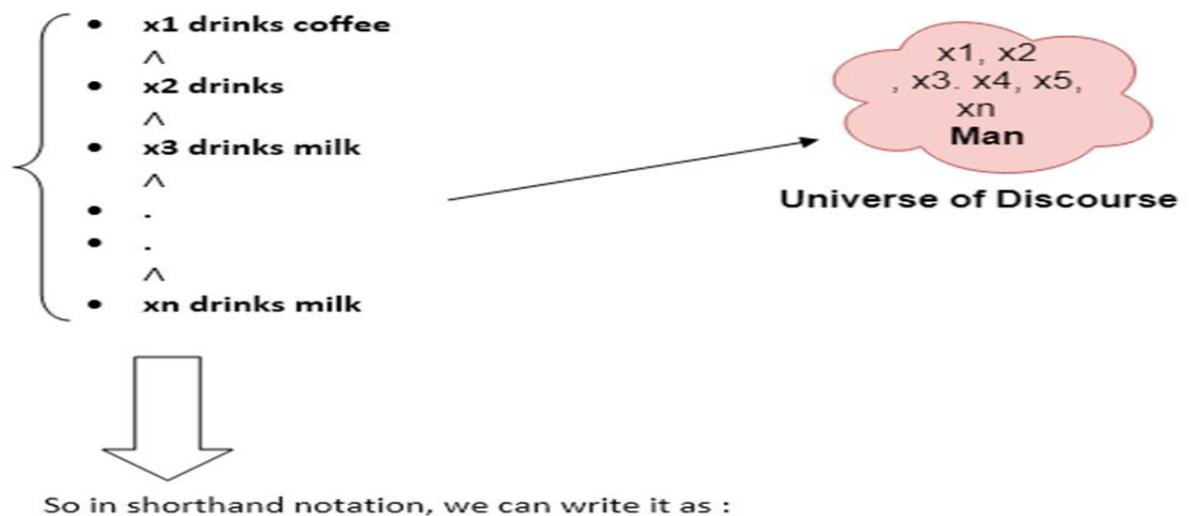


- Logic bậc nhất còn được gọi là logic vị từ hoặc là logic vị từ bậc nhất
- Giống như là một ngôn ngữ tự nhiên, có hai thành phần chính là cú pháp và ngữ nghĩa.

- Logic bậc nhất là một cách khác để biểu diễn tri thức trong trí tuệ nhân tạo. Nó là một phần mở rộng của logic mệnh đề
- FOL đủ biểu cảm để trình bày các câu lệnh ngôn ngữ tự nhiên một cách ngắn gọn
- Logic bậc nhất giúp phát triển thông tin về các đối tượng theo cách dễ dàng hơn và cũng có thể biểu thị mối quan hệ giữa các đối tượng đó.
- Logic bậc nhất không chỉ giả định rằng thế giới chứa đựng các sự kiện như logic mệnh đề mà còn giả định những điều sau đây trên thế giới: đối tượng, quan hệ, chức năng.

## II. Ví dụ minh họa liên quan FOL

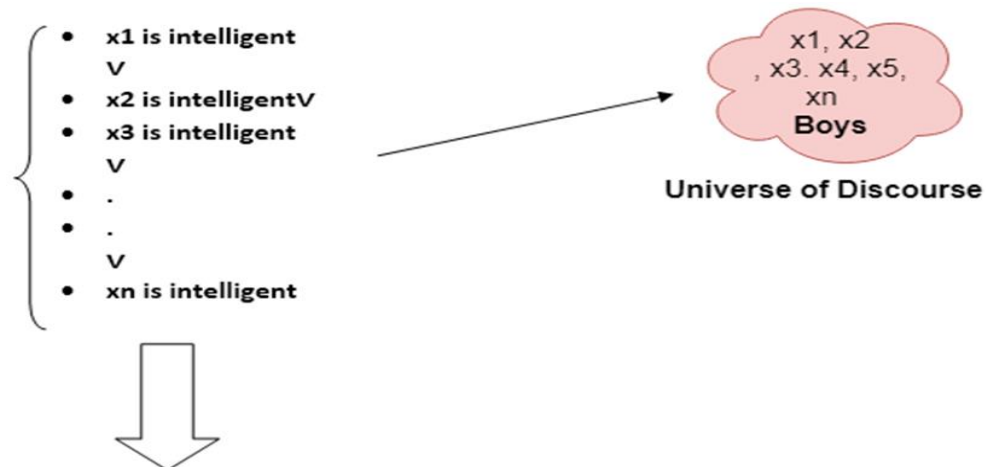
### 1/ All man drink coffee.



So in shorthand notation, we can write it as :

$$\forall x \text{ man}(x) \rightarrow \text{drink}(x, \text{coffee}).$$

### 2/ Some boys are intelligent



$\exists x: \text{boys}(x) \wedge \text{intelligent}(x)$

3/ Sasuke and Itachi are brothers  $\implies$  brothers (Sasuke, Itachi).



4/ All dogs bark.

$\forall x \text{ dog}(x) \rightarrow \text{bark}(x).$



5/ Elon Musk is the richest person.

$\forall x (\text{is-Elon-Musk}(x)) \rightarrow \text{is-richest}(x).$



### 1.1.2

**a. Khái quát chung về swi prolog:**

- **SWI Prolog** hoạt động dựa trên máy ảo Prolog định nghĩa các lệnh, không chỉ có khả năng biên dịch ngôn ngữ lập trình nhanh mà còn hỗ trợ chế độ khử lỗi, **SWI Prolog** cho phép người dùng xử lý các mã C và C++ trong giao diện linh hoạt cũng như hỗ trợ nhiều ứng dụng lớn và cung cấp các giá trị nguyên vô hạn.

- **SWI-Prolog** là một trong những môi trường lập trình Prolog phổ biến nhất. Nó là một hệ thống mã nguồn mở và có sẵn cho nhiều nền tảng, bao gồm Windows, macOS và Linux.
- **SWI Prolog** tích hợp chương trình tham khảo chéo Prolog, trình tạo dạng thi hành, thống kê lệnh đầu ra và một công cụ kiểm tra lỗi. Bên cạnh đó, gói SWI Prolog còn cung cấp nhiều thư viện hữu ích với các lập trình viên như các thành phần thao tác theo quy tắc ép buộc, thư viện giao diện, bộ phân tích cú pháp HTML, RDF và XML, thư viện server web quản lý các phiên, v.v.

**b. Để triển khai các chương trình Prolog trên SWI-Prolog, bạn cần thực hiện các bước sau:**

- Tải xuống và cài đặt SWI-Prolog trên máy tính của bạn: Bạn có thể tải SWI-Prolog từ trang web chính thức (<http://www.swi-prolog.org>). Làm theo hướng dẫn để cài đặt nó trên máy tính của bạn.
- Khởi chạy môi trường SWI-Prolog: Sau khi đã cài đặt SWI-Prolog, bạn có thể khởi chạy nó bằng cách nhấp đôi vào biểu tượng SWI-Prolog trên màn hình desktop của bạn hoặc bằng cách gõ "swipl" trong terminal.
- Viết mã code Prolog của bạn: Bạn có thể viết mã code Prolog của mình trong bất kỳ trình soạn thảo văn bản nào, chẳng hạn như Notepad, Sublime Text hoặc Visual Studio Code. Lưu tệp với phần mở rộng ".pl".
- Tải chương trình Prolog vào môi trường SWI-Prolog: Trong môi trường SWI-Prolog, gõ "[tên\_tệp]." (không bao gồm dấu ngoặc kép) để tải chương trình Prolog của bạn. Ví dụ, nếu chương trình Prolog của bạn được lưu trong tệp có tên "program.pl", bạn sẽ gõ "[program]."
- Kiểm tra chương trình Prolog của bạn: Sau khi chương trình Prolog của bạn được tải, bạn có thể kiểm tra nó bằng cách nhập các truy vấn vào môi trường SWI-Prolog. Ví dụ, nếu chương trình Prolog của bạn chứa một quy tắc để thêm hai số, bạn có thể nhập một truy vấn như "add(2,3,X)." để kiểm tra quy tắc.
- Sử dụng các công cụ SWI-Prolog: SWI-Prolog cung cấp một số công cụ hữu ích để làm việc với các chương trình Prolog. Ví dụ, bạn có thể sử dụng lệnh "trace" để điều chỉnh chương trình của bạn và xem cách nó thực thi hoặc sử dụng lệnh "profile" để đo hiệu suất của chương trình của máy tính.

**c. Các ví dụ cơ bản:**

**1. Gia Phả:**



- parent(john, mary).
- parent(john, lisa).
- parent(john, tom).
- parent(mary, anne).
- parent(mary, pat).
- parent(pat, jim).
- male(john).
- male(tom).
- male(pat).
- male(jim).
- female(mary).
- female(lisa).
- female(anne).
- sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.
- brother(X, Y) :- sibling(X, Y), male(X).

2. Tính giai thừa của 1 số nguyên dương:

- factorial(0, 1).
- factorial(N, F) :- N > 0, N1 is N - 1, factorial(N1, F1), F is N \* F1.

Để tính toán giai thừa của một số nguyên dương, ta có thể nhập truy vấn "factorial(5, F)." trong môi trường SWI-Prolog, và kết quả sẽ trả về giá trị F là 120.

3. Tính dãy fibonacci:

- fib(0, 0).
- fib(1, 1).
- fib(N, F) :- N > 1, N1 is N - 1, N2 is N - 2, fib(N1, F1), fib(N2, F2), F is F1 + F2.

Để tính toán số Fibonacci thứ n, ta có thể nhập truy vấn "fib(6, F)." trong môi trường SWI-Prolog, và kết quả sẽ trả về giá trị F là 8.

4. Phân biệt màu sắc hoa quả:

# Định nghĩa các màu sắc

- color(red).
- color(orange).

- color(yellow).
- color(green).
- color(blue).
- color(purple).

# Xác định màu sắc của từng loại hoa quả

- fruit\_color(apple, red).
- fruit\_color(apple, green).
- fruit\_color(orange, orange).
- fruit\_color(banana, yellow).
- fruit\_color(grape, purple).
- fruit\_color(blueberry, blue).

# Xác định một hoa quả có màu sắc nào

- has\_color(Fruit, Color) :- fruit\_color(Fruit, Color).

# Xác định một loại hoa quả có màu sắc nào

- is\_color(Fruit, Color) :- fruit\_color(Fruit, Color).

# Phân loại một loại hoa quả dựa trên màu sắc

classify\_fruit(Fruit, Result) :-

(is\_color(Fruit, red), Result = "This fruit is red.");

(is\_color(Fruit, orange), Result = "This fruit is orange.");

(is\_color(Fruit, yellow), Result = "This fruit is yellow.");

(is\_color(Fruit, green), Result = "This fruit is green.");

(is\_color(Fruit, blue), Result = "This fruit is blue.");

(is\_color(Fruit, purple), Result = "This fruit is purple.");

(Result = "I don't know what color this fruit is.").

=> Trong chương trình này, chúng ta định nghĩa các màu sắc có thể của hoa quả và xác định màu sắc của từng loại hoa quả. Sau đó, chúng ta xác định một hoa quả có màu sắc nào

và xác định một loại hoa quả có màu sắc nào. Cuối cùng, chúng ta phân loại một loại hoa quả dựa trên màu sắc.

Ta có thể nhập truy vấn "classify\_fruit(apple, Result)." trong môi trường SWI-Prolog, và kết quả sẽ trả về là "This fruit is red." hoặc "This fruit is green." tùy thuộc vào màu sắc của loại quả táo.

#### 5. Xác định động vật đẻ con hay đẻ trứng:

- # Xác định các động vật đẻ con
- mammal(dog).
- mammal(cat).
- mammal(elephant).
- #Xác định các động vật đẻ trứng
- reptile(snake).
- reptile(crocodile).
- reptile(chicken).

#Xác định một động vật là đẻ con hay đẻ trứng

- gives\_birth\_to(Animal, live) :- mammal(Animal).
- gives\_birth\_to(Animal, egg) :- reptile(Animal).

#Phân loại một động vật dựa trên cách sinh sản

classify\_animal(Animal, Result) :-

(gives\_birth\_to(Animal, live), Result = "This animal gives birth to live young.");

(gives\_birth\_to(Animal, egg), Result = "This animal lays eggs.");

(Result = "I don't know how this animal reproduces.").

=> Trong chương trình này, chúng ta định nghĩa các động vật đẻ con và các động vật đẻ trứng. Sau đó, chúng ta xác định một động vật là đẻ con hay đẻ trứng và phân loại một động vật dựa trên cách sinh sản.

Ta có thể nhập truy vấn "classify\_animal(dog, Result)." hoặc "classify\_animal(snake, Result)." trong môi trường SWI-Prolog, và kết quả sẽ trả về là "This animal gives birth to live young." hoặc "This animal lays eggs." tùy thuộc vào động vật.

### Câu hỏi:

- husband(prince\_Charles,X). (Prince Charles là chồng của ai?).

```
?- husband(prince_Charles,X).
X = camilla_Parker_Bowles.
```

- father(X,princess\_Anne). (Ai là cha của princess Anne?)

```
?- father(X,princess_Anne).
X = prince_Phillip.
```

- grandmother(princess\_Anne, isla\_Phillips). (Có phải Isla Phillips's là bà của princess Anne?)

```
?- grandmother(princess_Anne, isla_Phillips).
true
```

- sibling(prince\_George,princess\_Charlotte). (Có phải prince George và princess Charlotte là anh em ruột?)

```
?- sibling(prince_George,princess_Charlotte).
true
```

- niece(mia\_Grace\_Tindall,X). (mia Grace Tindall là cháu gái của ai?).

```
?- niece(mia_Grace_Tindall,X).
false.
```

- grandson(prince\_George,prince\_Charles). (Có phải prince George là cháu trai của prince Charles's grandson?)

```
?- grandson(prince_George,prince_Charles).
true
```

- aunt(zara\_Phillips,savannah\_Phillips). (Có phải Zara Phillips là dì của Savannah Phillips?).

```
?- aunt(zara_Phillips,savannah_Phillips).
true.
```

- grandson(X,prince\_Phillip). (Ai là cháu trai của Prince Phillip?)

```
?- grandson(X,prince_Phillip).
X = prince_William
```

- wife(zara\_Phillips,prince\_William). (Có phải Zara Phillips là vợ của Prince William?)

```
?- wife(zara_Phillips,prince_William).
false.
```

- grandchild(peter\_Phillips,X). (Ai là cháu của peter Phillips's?).  

```
?- grandchild(peter_Phillips,X).
X = prince_Phillip
```
- uncle(prince\_Edward,X). (Ai gọi Prince Edward là chú?).  

```
?- uncle(prince_Edward,X).
X = prince_William
```
- grandparent(princess\_Diana,X). (Ai là ông bà của queen Princess Diana?).  

```
?- grandparent(princess_Diana,X).
X = prince_George
```
- granddaughter(savannah\_Phillips,X). (Savannah Phillips là cháu gái của ai?).  

```
?- granddaughter(savannah_Phillips,X).
X = captain Mark Phillips
```
- brother(prince\_Harry,X). (Ai là anh trai của Prince Harry?).  

```
?- brother(prince_Harry,X).
X = prince_William
```
- child(X,prince\_Edward). (Ai là cháu của prince Edward?).  

```
?- child(X,prince_Edward).
X = james_Viscount_Severn
```
- sister(isla\_Phillips,X). (Ai là chị của Isla Phillips?).  

```
?- sister(isla_Phillips,X).
X = savannah_Phillips
```
- niece(zara\_Phillips,prince\_Beatrice). (Có phải Zara Phillips là niece của Prince Beatrice?).  

```
?- niece(zara_Phillips,prince_Beatrice).
false.
```
- mother(princess\_Anne,princess\_Beatrice). (Có phải Princess Anne là mẹ của princess Beatrice?).  

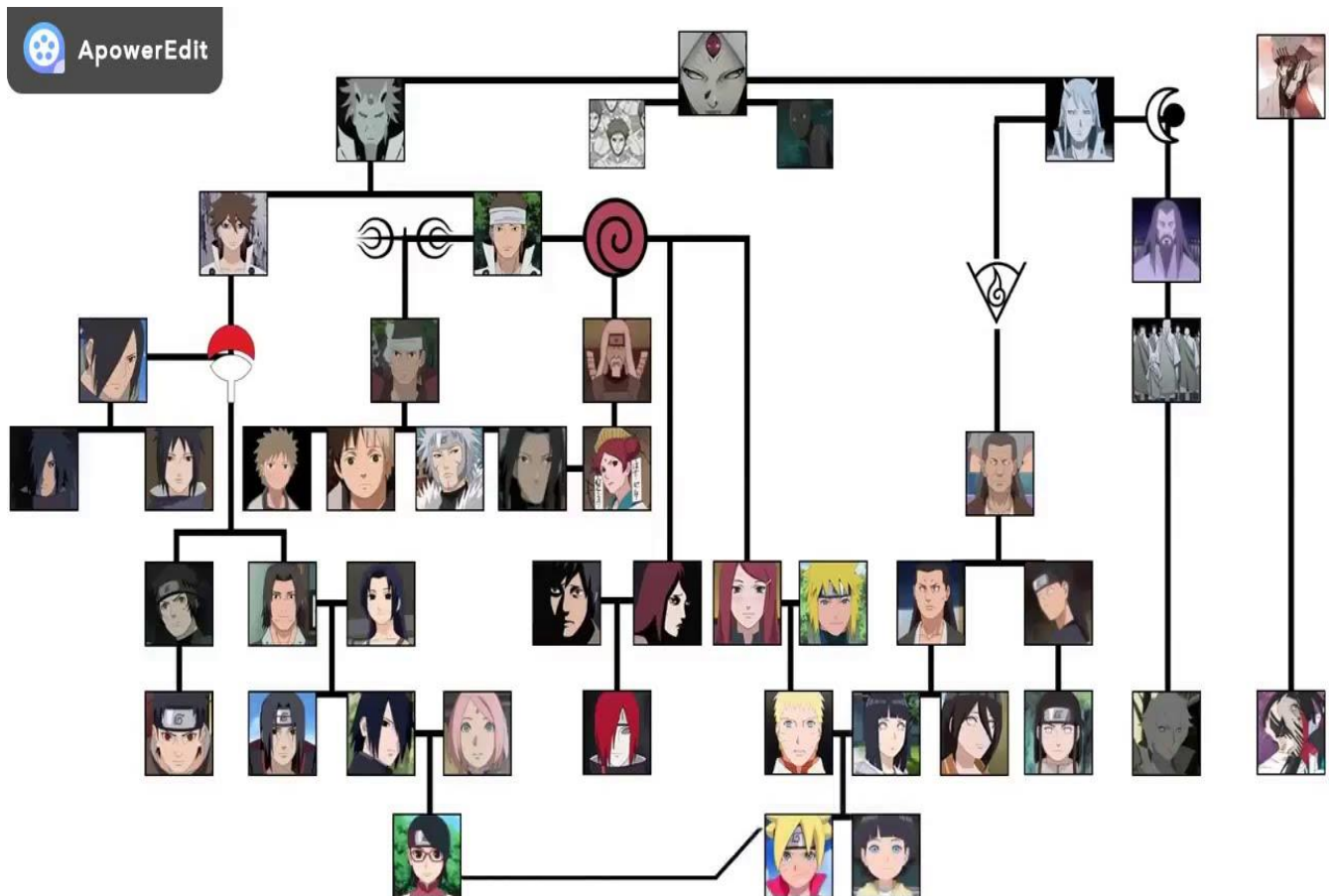
```
?- mother(princess_Anne,princess_Beatrice).
false.
```
- father(mike\_Tindall,kate\_Middleton) (Có phải Mike Tindall là cha của Kate Middleton?).  

```
?- father(mike_Tindall,kate_Middleton).
false.
```

- wife(queen\_Elizabeth\_II, mia\_Grace\_Tindall). (Có phải Queen Elizabeth là vợ của Mia Grace Tindall?)

```
?- wife(queen_Elizabeth_II, mia_Grace_Tindall).  
false.
```

## 1.2 Xây dựng cơ sở tri thức với Prolog



Cây gia phả của tất cả các tộc trong thế giới shinobi của phim Naruto

Nguồn : <https://www.youtube.com/watch?v=MRJyJ3MMByQ>

**Câu hỏi:**

1/ Sasuke có bạch nhãn không ?

Syntax: whiteeyes(sasuke\_uchiha).

?-

| whiteeyes(sasuke\_uchiha).

**false.**

2/ Naruto có mồ côi không ?

Syntax: orphanage(naruto\_uzumaki).

?- orphanage(naruto\_uzumaki).

**true**

3/ Kaguya có phải lục đạo tiên nhân không ?

Syntax:lucdaotiennhan(kaguya).

?- lucdaotiennhan(kaguya).

**true.**

4/ Cháu trai của Uchiha Fukagu là ai ?

Syntax: nephew(X,fugaku\_uchiha).

?- nephew(X,fugaku\_uchiha).

X = madara\_uchiha ;

X = izuna\_uchiha ;

X = shisui\_uchiha ;

5/ Itachi có sharingan hay không ?

Điều kiện để có sharinga: ám sát người thân, thuộc dòng họ Uchiha.

Syntax: sharingan(itachi\_uchiha,shisui\_uchiha,itachi).

Với: X: là đối tượng cần xác định có sharingan hay không?

Y: là người thân gia đình hoặc bạn thân.

Z: tên của thành viên thuộc dòng họ Uchiha.

?- `sharingan(itachi_uchiha,shisui_uchiha,itachi).`  
**true .**

6/ Minato có phải chồng của Uzumaki Kushina không ?

Syntax: `husband(minato_namikaze,kushina_uzumaki).`

?- `husband(minato_namikaze,kushina_uzumaki).`  
**true.**

7/ Ba của Sasuke là ai ?

Syntax: `father(X,sasuke_uchiha).`

?- `father(X,sasuke_uchiha).`  
**X = fugaku\_uchiha .**

8/ Mẹ của Itachi là ai ?

Syntax: `mother(X,itachi_uchiha).`

?-  
| `mother(X,itachi_uchiha).`  
**X = mikoto\_uchiha.**

9/ Itachi có phải anh trai của Sasuke?

Syntax: `brother(itachi_uchiha,sasuke_uchiha).`

?- `brother(itachi_uchiha,sasuke_uchiha).`  
**true |**

10/ Ba của sarada là ai ?



Syntax: father(X,sarada\_uchiha).

```
?- father(X,sarada_uchiha).  
X = sasuke_uchiha .
```

11/ Anh em họ Uchiha Itachi có những ai?

Syntax: cousin(X, itachi\_uchiha).

```
?- cousin(X, itachi_uchiha).  
X = shisui_uchiha ;  
X = madara_uchiha ;  
X = izuna_uchiha ;
```

12/ Ai là cậu của Sarada?

Syntax: uncle(X,sarada\_uchiha).

```
?-  
|   uncle(X,sarada_uchiha).  
X = itachi_uchiha .
```

13/ Mẹ/Dì của Uchiha Shisui ?

Syntax: aunt(X,shisui\_uchiha).

```
?- aunt(X,shisui_uchiha).  
X = mikoto_uchiha .
```

14/ Chồng của Uzumaki Kushina là ai ?

Syntax: husband(X,kushina\_uzumaki).

?- husband(X,kushina\_uzumaki).  
X = minato\_namikaze.

15/ Cháu trai của Uchiha Kagami là ai ?

Syntax: nephew(X,kagami\_uchiha).

?- nephew(X,kagami\_uchiha).  
X = madara\_uchiha ;  
X = izuna\_uchiha ;  
X = itachi\_uchiha .

16/ Con rể của Hyuga Hiashi là ai ?

Syntax: conre(X,hiashi\_hyuga).

?- conre(X,hiashi\_hyuga).  
X = naruto\_uzumaki ;

17/ Con dâu của Uzumaki Kushina là ai ?

Syntax: condau(X,kushina\_uzumaki).

?- condau(X,kushina\_uzumaki).  
X = hinata\_hyuga .

18/ Ông nội của Uzumaki Boruto là ai ?

Syntax: husgrandpa(X,boruto\_uzumaki).

?- husgrandpa(X,boruto\_uzumaki).  
X = minato\_namikaze .

19/ Bà nội của Uzumaki Himawari ?

Syntax: `husgrandma(X,himawari_uzumaki).`

```
?- husgrandma(X,himawari_uzumaki).  
X = kushina_uzumaki .
```

20/ Con trai của Hyuga Hizashi là ai ?

Syntax: `son(X,hizashi_hyuga).`

```
?- son(X,hizashi_hyuga).  
X = neji_hyuga.
```

Test datasets backup link: <http://codepad.org/zUYZxidC>

### 1.3 Thực hiện hệ thống suy luận logic trong ngôn ngữ lập trình.

Ngôn ngữ lập trình : python

Phương pháp suy luận : forward reasoning

Các file thuộc phần input data :

- `input.txt` : là file chứa cơ sở tri thức của cây gia phả British Royal Family
- `inputquestion.txt` : là những câu hỏi dành cho cây gia phả British Royal Family giống với những câu hỏi (dataset) trong phần 1.1
- `input2.txt` : là file chứa cơ sở tri thức của cây gia phả Naruto
- `input2question.txt` : là những câu hỏi dành cho cây gia phả của tất cả các tộc trong thế giới shinobi của phim Naruto giống với những câu hỏi (dataset) trong phần 1.2

File source python : `src.py`

Mã giả:

Hàm tạo mối quan hệ cha-con , mẹ-con trong gia đình:

```
func createFamilyBacktrack()
```

```

For each person registered:
    Current = person Gap[person, current] = 0
    While Current is not Null
        Gap[person, current.parent] = Gap[person, current] + 1
        Current = current.parent

```

Hàm kiểm tra các mối quan hệ trong gia đình

Check\_relationship\_family(person1, person2):

```

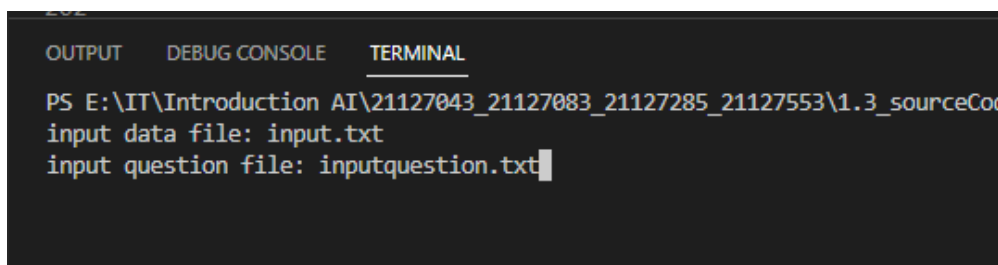
    State = 0
    Distance = 250
    Middleman = None
    For individual in family tree
        If not having relationship between person1 or person 2 and individual
            Continue
        Caculate max distance between person1 or person 2 and individual
        If max distance >= distance : continue
        Else : update distance , state , middleman
    If not find middleman , check person 1 and person2 is husband-wife together , or ex-
    husband-wife or not having relationship
    If middleman is person1 or person2
        Return the relationship base on the value of state

```

Cách thức chương trình chạy và kết quả sau khi chạy :

Khi chạy chương trình thì sẽ được yêu cầu nhập input data file và input question file

Ứng với input data file : input.txt thì chỉ có thể nhập input question file : inputquestion.txt



```

PS E:\IT\Introduction AI\21127043_21127083_21127285_21127553\1.3_sourceCod
input data file: input.txt
input question file: inputquestion.txt

```

Kết quả sau khi chạy :

```
input data file: input.txt
input question file: inputquestion.txt
camilla_parker_bowles
peter_phillips
True
True
Can't find!
True
True
Can't find!
Can't find!
Can't find!
Can't find!
Can't find!
Can't find!
prince_william
prince_phillip
savannah_phillips
False
False
prince_george
False
```

Ứng với input data file : input2.txt thì chỉ có thể nhập input question file : input2question.txt

```
PS E:\IT\Introduction AI\2112\043_2112\083_2112\285_
input data file: input2.txt
input question file: input2question.txt
```

Kết quả sau khi chạy :

```

input data file: input2.txt
input question file: input2question.txt
True
True
False
True
True
False
Can't find!
patriarch_clan_otsutsuki
Can't find!
Can't find!
Can't find!
boruto_uzumaki
himawari_uzumaki
fugaku_uchiha
sarada_uchiha
True
patriarch_clan_otsutsuki
True
Can't find!
Can't find!
True
Can't find!
madara_uchiha
Can't find!
sarada_uchiha
minato_namikaze
patriarch_clan_otsutsuki
hyuga_elder

```

- Ngoài ra code cũng chạy được kiểu : python scr.py [tên file data] [tên file câu hỏi]

#### IV. Nguồn tham khảo

<https://web.engr.oregonstate.edu/~walkiner/teaching/cs381wi21/slides/6.LogicProgramming.pdf>

<https://eclipseclp.org/doc/tutorial/tutorial021.html>

<https://masterwed.files.wordpress.com/2010/07/ltprolog123.pdf>