

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN – ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**KHOA CÔNG NGHỆ THÔNG TIN**



## **BÁO CÁO ĐỒ ÁN 2**

**ĐỒ ÁN:**

## **LOGIC BẬC NHẤT**

**MÔN HỌC: CƠ SỞ TRÍ TUỆ NHÂN TẠO**

**LỚP: 18\_22**

**GVHD: DƯƠNG NGUYỄN THÁI BẢO**

Mục lục:

<b>1 THÔNG TIN THÀNH VIÊN</b>	<b>1</b>
<b>I. Ngôn ngữ Prolog:</b>	<b>2</b>
1) Giới thiệu:	2
2) Cú pháp:	2
□ Các kí hiệu bao gồm:	2
- :- (điều kiện nếu).	2
- , (điều kiện và).	2
- ; (điều kiện hoặc).	2
- . (kết thúc vị từ).	2
3) Các phép toán trong prolog:	3
3.1) Số học:	3
3.1) Các vị từ xác định:	4
3.1) Các vị từ hàm toán học:	4
4) Các hàm nhập xuất chuẩn:	6
4.1) Xuất ra màn hình:	6
4.1) Nhập từ bàn phím:	6
5) Các nguyên tắc trong prolog:	6
<b>II. SWI Prolog:</b>	<b>8</b>
1) Giới thiệu chung về SWI Prolog:	8
2) Ví dụ sử dụng SWI Prolog:	8
2.a) Cách truy vấn	8
2.b) Luật , vị từ và biến trong SWI Prolog:	9
2.c) Cách suy diễn trong Prolog:	11
3) Các ví dụ:	11
<b>III. Ví dụ về cơ sở tri thức</b>	<b>15</b>
<b>IV. Xây dựng chương trình suy diễn</b>	<b>16</b>

1 THÔNG TIN THÀNH VIÊN

MSSV	HỌ VÀ TÊN	EMAIL	ĐÓNG GÓP
18120488	Trần Phúc Nguyên	18120488@students.hcmus.edu.vn	30%
18120477	Đỗ Trọng Nghĩa	18120477@students.hcmus.edu.vn	30%
18120466	Trần Hoàng Nam	18120466@students.hcmus.edu.vn	30%

## I. Ngôn ngữ Prolog:

### 1) Giới thiệu:

- 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 của logic, dựa trên 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 học trong khoa học 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.

### 2) Cú pháp:

#### • Các kí hiệu bao gồm:

- :- (điều kiện nếu).
- , (điều kiện và).
- ; (điều kiện hoặc).
- . (kết thúc vị từ).
- <...>:-<...>(luật).
- ?-<...>(câu hỏi).
- /\* chú thích nhiều dòng.
- % chú thích theo từng dòng.

#### • Kiểu hằng số: số nguyên, số thực:

- vd: 1                5.2                -1.123                -80

#### • Kiểu hằng logic: true hoặc false

#### • Kiểu hằng chuỗi ký tự: chuỗi được đặt trong ngoặc kép

- vd: "Công nghệ thông tin",    "" (rỗng)

#### • Biến: Là 1 chuỗi ký tự gồm chữ cái, chữ số, bắt đầu bởi chữ hoa hoặc dấu gạch dưới dòng.

- vd: A, X, List\_members, X.

#### • Mệnh đề Horn:

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

Head:- Body.

- Head là một vị từ logic, còn Body có thể là rỗng hoặc là một tập các vị từ logic. Ví dụ như sau:

chẵn(X):- X chia\_dư 2 = 0.

- Phần lớn các bộ dịch của 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 thường.

### **Luật:**

#### **P**

- Phần còn lại của các mệnh đề trong một chương trình Prolog được gọi là luật. Nó thường thể hiện những phát biểu logic trong bài toán, ví dụ như cha là người phụ huynh và là nam:

cha(X,Y):- phu\_huynh(X,Y),nam(X).

- Dấu phẩy "," trong mệnh đề trên được dịch là toán tử "và"; biến trong Prolog được quy ước bắt đầu là một chữ cái hoa.

### **Câu hỏi:**

- Dùng để truy vấn đến cơ sở tri thức. Ví dụ: muốn tìm những người là bạn của nghĩa:

?- friendNghia(X).

- Prolog sẽ tiến hành truy vấn đến cơ sở tri thức tìm ra kết quả thỏa câu truy vấn, lần lượt hiện ra những kết quả thỏa yêu cầu(sử dụng dấu ";" ) đến khi nào không còn kết quả lời giải nào nữa thì dừng lại. Như sau:

?- friendNghia(X).

X = nguyen;

X = nam;

- Để kết thúc luồng trả lời nhấn enter hoặc dấu chấm.

- Để hỏi các câu hỏi trong 1 lần gọi ta dùng dấu ";" để ngăn cách giữa các câu hỏi(";" tương đương phép and). Ví dụ nguyên và nam có bạn chung không:

?-friend(X,nguyen),friend(X,nam).

- Prolog sẽ truy vấn đến cơ sở tri thức tìm xem có ai vừa là bạn của nguyên vừa là bạn của nam không, nếu không phải là bạn của 1 trong 2 người câu trả lời sẽ trả ra không tức là không tìm ra

### **Đệ qui trong prolog:**

- Dùng để xác định quan hệ của các ông bà tổ tiên, yêu cầu xác định nhiều luật trong cây phả hệ

- Để thuận tiện cho việc đọc chương trình Prolog, ta có thể viết một luật trên nhiều dòng, dòng đầu tiên là phần đầu của luật, các dòng tiếp theo là phần thân của luật, mỗi đích trên một dòng phân biệt. Bây giờ quan hệ ongba được viết lại như sau :

ongba(X,Y):- bome(X,Z),bome(Z,Y)

### **3) Các phép toán trong prolog:**

#### **3.1) Số học:**

Phép cộng: +.

Phép trừ: -.

Phép nhân: \*.

Phép chia số nguyên: //

Phép chia số thực: /

Phép chia lấy dư: mod

Biểu thức số học được xây dựng bằng vị từ is. Đối số nằm bên trái là một đối tượng sơ cấp. Đối số bên phải là một biểu thức toán học.

- Ngoài ra prolog còn cung cấp các phép toán so sánh như >(lớn), <(nhỏ), <=(nhỏ hơn hoặc bằng), >=(lớn hơn hoặc bằng), <=>(khác), =(bằng), between(a,b,Z) ( $a < Z < b$ ), succ(Int1,Int2)

### 3.1) Các vị từ xác định:

Integer(I): I là một số nguyên?

float(R) R là một số thực (dấu chấm động) ?

number(N) N là một số (nguyên hoặc thực) ?

Nonvar(X) : X không phải là biến?

Atom(A): A là một nguyên tử?

compound(X) X là một hạng có cấu trúc ?

ground(X) X là một hạng đã hoàn toàn ràng buộc ?

atomic(A) A là một nguyên tử hoặc một số ?

### 3.1) Các vị từ hàm toán học:

Sin(X): Tính  $\sin(x)$ .

Tan(X): Tính  $\tan(x)$ .

Arctan(X): Tính  $\arctan(x)$ .

Ln(X): Tính logarit cơ số e của X.

Log(X): Tính logarit cơ số 10 của X.

Round(X): Cho ta số nguyên là số X được làm tròn, dấu là dấu của X.

Trunc(X): Cho phần nguyên của số X dấu là dấu của X.

Abs(X): Cho ta trị tuyệt đối của X.

Random(X): Cho ta số thực X nằm trong khoảng từ  $[0,1)$ .

Random(Y,X): Cho ta số nguyên X nằm trong khoản từ  $[0,Y)$ .

#### 4) Các hàm nhập xuất chuẩn:

##### 4.1) Xuất ra màn hình:

Write (Arg1, Arg2, ..., Argn) in ra màn hình giá trị của các đối số.

Writef (Formatstring, Arg1, Arg2, ..., Argn) in ra màn hình giá trị của các đối số theo định dạng được chỉ định trong Formatstring.

- Trong đó Formatstring là một chuỗi có thể là:

“%d”: In số thập phân bình thường; đối số phải là char hoặc integer.

“%c”: Đối số là một số integer, in ký tự có mã Ascci là đối số đó, chẳng hạn writef(“%c”,65) được A.

“%e”: In số thực dưới dạng lũy thừa của 10.

“%x”: In số Hexa; đối số phải là char hoặc integer.

“%s”: In một chuỗi hoặc một symbol.

##### 4.1) Nhập từ bàn phím:

Readln(X): Nhập một chuỗi ký tự vào biến X.

ReadInt(X): Nhập một số nguyên vào biến X.

ReadReal(X): Nhập một số thực vào biến X.

ReadChar(X): Nhập vào một ký tự vào biến X.

#### 5) Các nguyên tắc trong prolog:

- Giải quyết vấn đề trong prolog sử dụng chủ yếu dựa vào 2 nguyên tắc : Đồng nhất, Quay lui.

##### **ĐỒNG NHẤT:**

- Một quan hệ có thể đồng nhất với một quan hệ nào đó cùng tên, cùng số lượng tham số, các đại lượng con cũng đồng nhất theo từng cặp.

- Một hằng có thể đồng nhất với một hằng.

- Một biến có thể đồng nhất với một hằng nào đó và có thể nhận luôn giá trị hằng đó.

- Chẳng hạn trong ví dụ nếu ta sử dụng goal dep(lan) thì có kết quả là Yes. Nếu ta dùng goal dep(X) thì sẽ có 3 kết quả: X=lan, X=hong và X=binh.

- Khi ta dùng goal dep(lan) thì dep(lan) sẽ đồng nhất với sự kiện dep(lan) trong phần clauses và do hai vị từ đồng nhất với nhau và hai đối số hằng đồng nhất nhau nên kết quả là Yes.

- Khi dùng goal dep(X) thì dep sẽ được đồng nhất với dep và biến X đồng nhất với hằng lan, do đó ta có kết quả X=lan. Tương tự X=hong và X=binh.

##### **QUAY LUI:**

- Giả sử hệ thống đang chứng minh goal g, trong đó g được mô tả như sau:

g:- g1, g2, ..., gj-1, gj, ..., gn.

- Khi các gi kiểm chứng từ trái sang phải, đến gj là sai thì hệ thống sẽ quay lui lại gj-1 để tìm lời giải khác.

- Chẳng hạn trong ví dụ 2 nói trên, khi ta yêu cầu Goal: thích(lan,X), ta được X=binh.



- Vị từ  $\text{thích}(\text{lan}, X)$  sẽ được đồng nhất với  $\text{thích}(\text{lan}, X)$  trong phần clauses, theo đó hệ thống phải chứng minh  $\text{thích}(\text{lan}, X) :- \text{khoe}(X), \text{dep}(X), \text{thong\_minh}(X)$ .
- Trước hết đồng nhất  $\text{khoe}(X)$  với  $\text{khoe}(\text{thuy}) \Rightarrow X = \text{thuy}$ .
- Do  $\text{dep}(\text{thuy})$  sai nên quay lui đồng nhất  $\text{khoe}(X)$  với  $\text{khoe}(\text{lan}) \Rightarrow X = \text{lan}$ .
- Do  $\text{dep}(\text{lan})$  đúng nên tiếp tục kiểm tra  $\text{thong\_minh}(\text{lan})$ .
- Do  $\text{thong\_minh}(\text{lan})$  sai nên quay lui để đồng nhất  $\text{khoe}(X)$  với  $\text{khoe}(\text{binh})$  để có  $X = \text{binh}$ , sau đó kiểm tra thấy  $\text{dep}(\text{binh})$  và  $\text{thong\_minh}(\text{binh})$  đều đúng nên  $X = \text{binh}$  là một nghiệm.

### **BỘ KÝ TỰ, TỪ KHOÁ:**

- Prolog dùng bộ ký tự sau: các chữ cái và chữ số ( $A - Z, a - z, 0 - 9$ ); các toán tử ( $+, -, *, /, <, =, >$ ) và các ký hiệu đặc biệt.

- Một số từ khoá:

**Trace:** Khi có từ khoá này ở đầu chương trình, thì chương trình được thực hiện từng bước để theo dõi; dùng phím F10 để tiếp tục.

**Fail:** Khi ta dùng goal nội, chương trình chỉ cho ta một kết quả (mặc dù có thể còn những kết quả khác), để nhận về tất cả các kết quả khi chạy goal nội, ta dùng toán tử Fail.

! hay còn gọi là nhất cắt, goal ngoại luôn cho ta mọi kết quả, muốn nhận chỉ một kết quả từ goal ngoại, ta dùng ký hiệu !.

## II. SWI Prolog:

### 1) Giới thiệu chung về SWI Prolog:

- 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 logic.
- Prolog được sử dụng nhiều trong các ứng dụng trí tuệ nhân tạo và ngôn ngữ học trong khoa học máy tính. SWI
- Prolog là một môi trường lập trình Prolog rất phổ biến, có các phiên bản chạy trên các hệ điều hành khác nhau như Windows, MacOS, Linux. SWI Prolog là một môi trường mã nguồn mở và thường được sử dụng trong dạy học.
- SWI Prolog còn được sử dụng như là một ngôn ngữ nhúng và dùng như là các cơ sở dữ liệu suy diễn

### 2) Ví dụ sử dụng SWI Prolog:

Dùng trình soạn thảo do SWI Prolog để tạo ra cơ sở tri thức. Ghi lại chương trình trong một tập tin định dạng văn bản sử dụng đuôi của tập tin là .pl

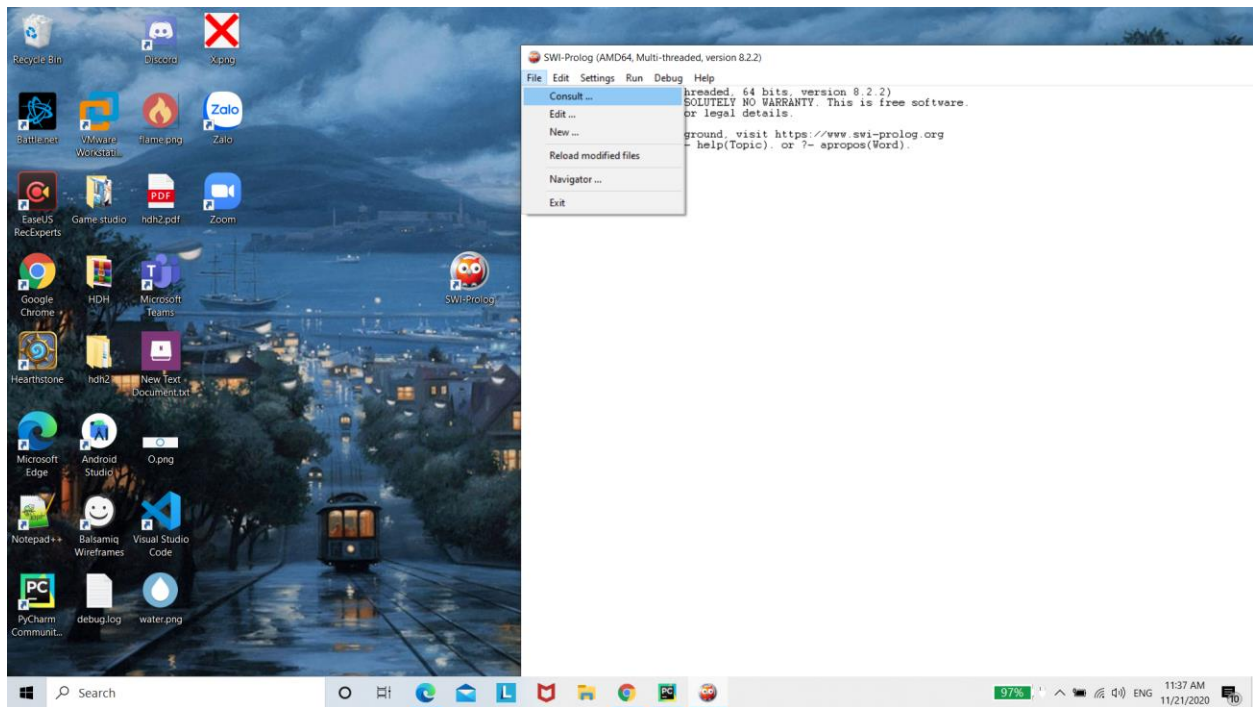
Mở phần mềm SWI Prolog và consult tới tập tin của chương trình. Sau đó đưa ra câu hỏi mong muốn.

Ví dụ: Cho các suy diễn logic như sau

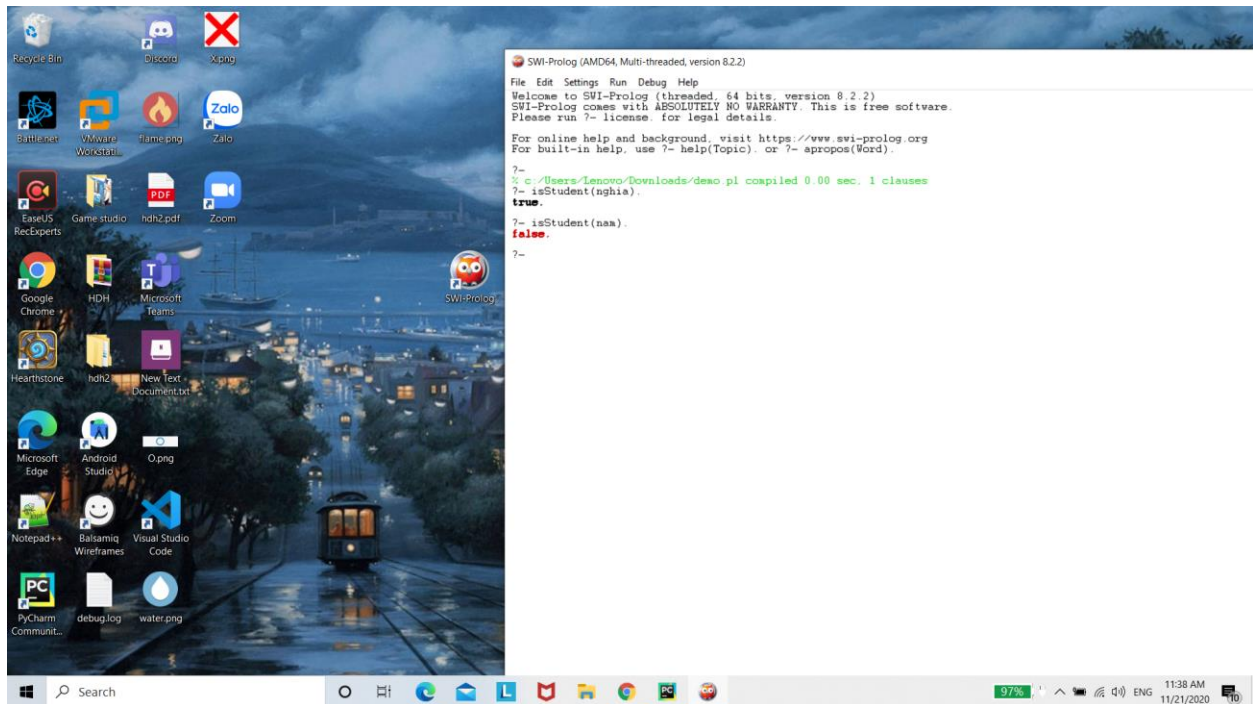
- Nguyên là sinh viên.
- Mọi sinh viên phải tốt nghiệp phổ thông.
- Hỏi: Nguyên có tốt nghiệp phổ thông không ?

#### 2.a) Cách truy vấn

Consult tới tập tin chứa cơ sở tri thức:



Truy vấn cơ sở tri thức:



Thực hiện 2 câu truy vấn isStudent(nghia),isStudent(nam) cho 2 kết quả true và false.

Prolog trả về true nếu chứng minh được, trả về false nếu không chứng minh.

2.b) Luật ,vị từ và biến trong SWI Prolog:

- **Luật:**

Một luật được biểu diễn bao gồm:

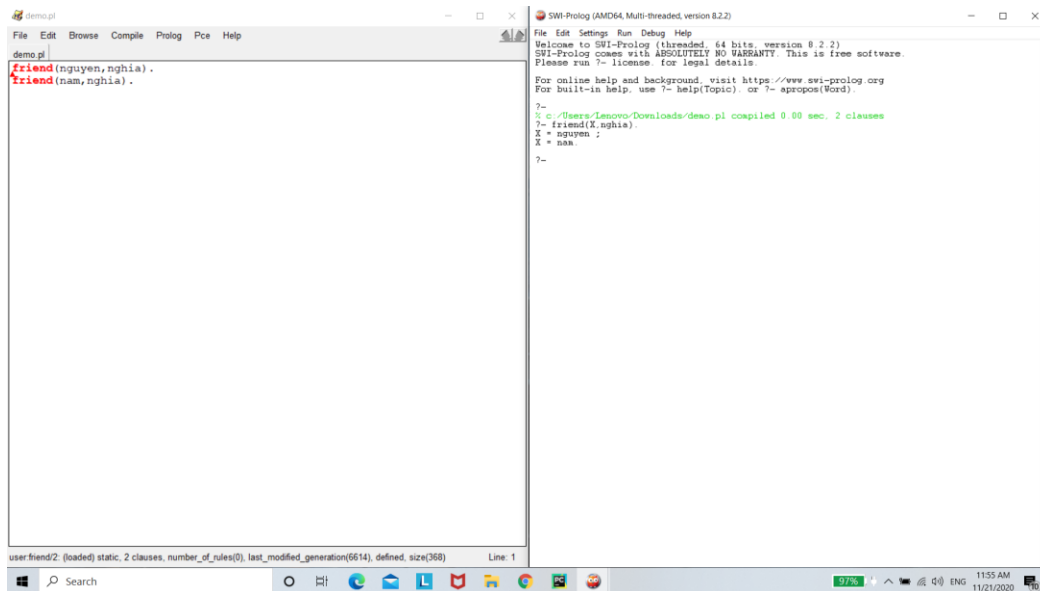
- Một cấu trúc biểu diễn mệnh đề kết luận của luật

- Ký hiệu: :-
- Một danh sách các cấu trúc biểu diễn mệnh đề giả thiết của luật ngăn cách bởi dấu ‘,’.  
Dấu ‘,’ giữa các cấu trúc có nghĩa như toán tử logic AND
- **Vị từ:**

Là tập hợp các mệnh đề cùng tên và một số tham số:

vd:

- friend(nguyen,nghia).
- friend(nghia,nam).



## - **Biến:**

Tên biến là một chuỗi ký tự gồm chữ cái, chữ số, bắt đầu bởi chữ hoa hoặc dấu gạch dưới dòng

Vd:

X, Y, A

Result, List\_of\_members \_x23, \_X, \_, ...

- **Hằng:**
- Hằng Logic:

Prolog sử dụng hai hằng Logic có giá trị là true và fail. Thông thường các hằng Logic không được dùng như tham số mà được dùng như các mệnh đề. Hằng fail thường được dùng để tạo sinh lời giải bài toán.

- Hằng chuỗi ký tự:

Các hằng là chuỗi (string) các ký tự được đặt giữa hai dấu nháy kép.

"Toto \#{ @ tata" chuỗi có tùy ý ký tự

"" chuỗi rỗng (empty string)

"\" chuỗi chỉ có một dấu nháy kép.

- Hằng nguyên tử:

Các hằng nguyên tử Prolog là chuỗi ký tự ở một trong ba dạng như sau :

(1) Chuỗi gồm chữ cái, chữ số và ký tự \_ luôn luôn được bắt đầu bằng một chữ cái in thường.

newyork      a\_

nil            x\_\_y

x25           tom\_cruise

(2) Chuỗi các ký tự đặc biệt :

<--->        ...

=====>      ::=

...

(3) chuỗi đặt giữa hai dấu nháy đơn (quote) được bắt đầu bằng chữ in hoa, dùng phân biệt với các tên biến :

'Jerry'        'Tom SMITH'

- Hằng kiểu số:

Prolog sử dụng cả số nguyên và số thực. Cú pháp của các số nguyên và số thực rất đơn giản, chẳng hạn như các ví dụ sau :

1        1515   0        -97

3.14    -0.0035        100.2

2.c) Cách suy diễn trong Prolog:

Prolog thực hiện suy diễn lùi

Giả sử chúng ta có cơ sở tri thức:

- friend(nguyen,X):-male(X),idiot(X).

- male(nghia).

- male(nam).

- idiot(nghia).

Prolog thực hiện suy diễn như sau:

1.male(X) = male(nam).      X=nam

idiot(nam). (False)

2.male(X) = male(nghia).      X=nghia

idiot(nghia). (True)

### 3)Các ví dụ:

- 3.1)

Có cơ sở tri thức sau:

parent(nam,nhan).

parent(lan,nhan).

parent(nhan,thuy).

parent(nhan,ngoc).

married(nam,lan).

grandParent(X,Y):-parent(Z,Y),parent(X,Z).

The screenshot shows the SWI-Prolog IDE with two windows. The left window, titled 'demo.pl', contains the following Prolog code:

```
parent(nam,nhan).
parent(lan,nhan).
parent(nhan,thuy).
parent(nhan,ngoc).
married(nam,lan).
grandParent(X,Y):-parent(Z,Y),parent(X,Z).
```

The right window shows the Prolog prompt and the execution of the query `?- grandParent(X,Y).`. The output is:

```
?- grandParent(X,Y).
X = nam,
Y = nhan ;
X = lan,
Y = nhan ;
X = nhan,
Y = thuy ;
X = nhan,
Y = ngoc ;
X = _G1,
Y = _G2,
fail.
```

- 3.2)

Làm toán trên Prolog

Cho công thức sau: compare(X,Y):- Y is (X+1)\*2.

The screenshot shows the SWI-Prolog IDE with two windows. The left window, titled 'demo.pl', contains the following Prolog code:

```
compare(X,Y):- Y is (X+1)*2.
```

The right window shows the Prolog prompt and the execution of the query `?- compare(1,X).`. The output is:

```
?- compare(1,X).
ERROR: Unknown procedure: compare/2
ERROR: However, there are definitions for:
ERROR: compare/3
false.
```

- 3.3)

Tính giai thừa:

fac(0, 1).

fac(N, F) :-

N > 0,

M is N - 1,

fac(M, Fm),

F is N \* Fm.

```
demo.pl
File Edit Browse Compile Prolog Pce Help
demo.pl
fac(0, 1).
fac(N, F) :-
N > 0,
M is N - 1,
fac(M, Fm),
F is N * Fm.

user:fac/2: (loaded) static, 2 clauses, number_of_rules(1), last_modified_generation(6614), defined, size(536) Line: 1
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.2.2)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?-
% c:/Users/Lesovo/Downloads/demo.pl compiled 0.00 sec, 2 clauses
?- fac(0,1).
true;
false.

?- fac(10,F).
F = 3628800;
false.

?-
```

- 3.4) Có cơ sở tri thức sau:

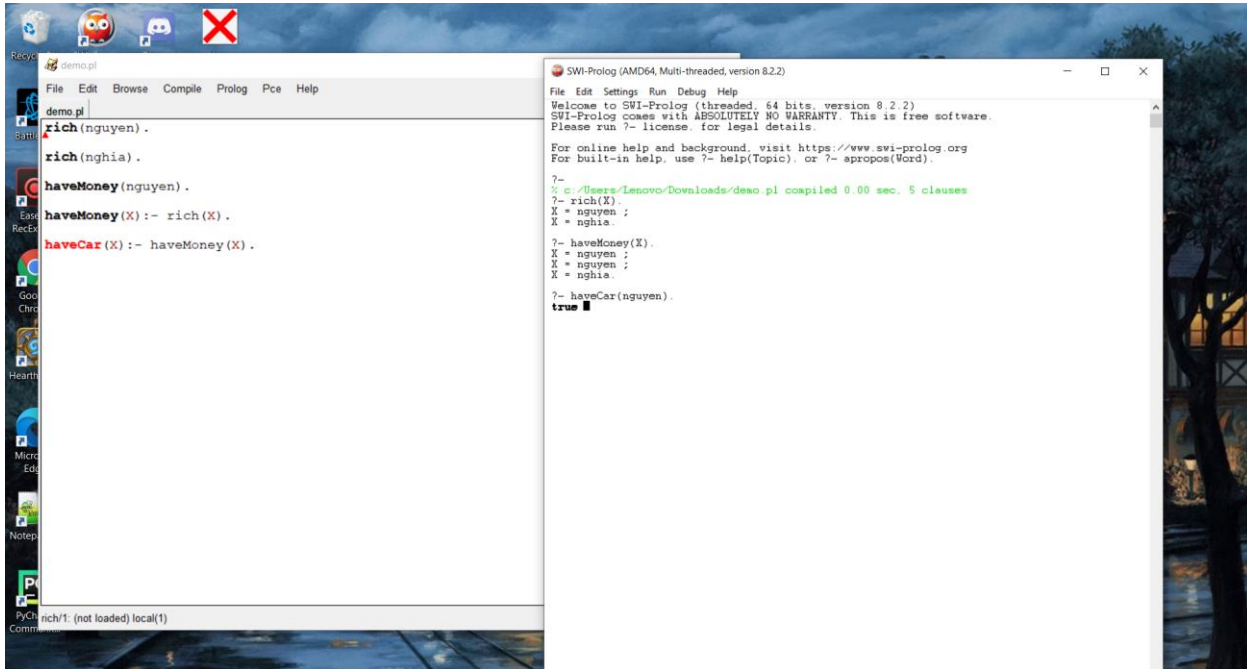
rich(nguyen).

rich(nghia).

haveMoney(nguyen).

haveMoney(X):- rich(X).

haveCar(X):- haveMoney(X).



- 3.5)

Cho cơ sở tri thức:

student(nguyen).

student(nam).

student(nghia).

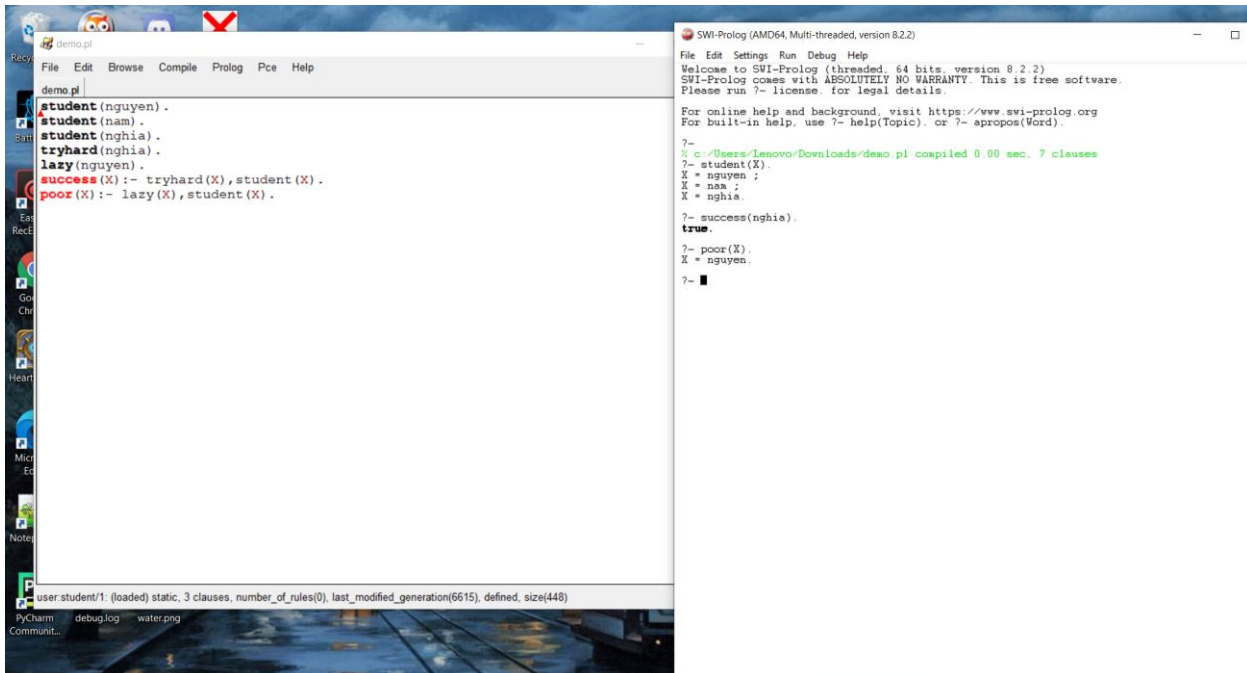
tryhard(nghia).

lazy(nguyen).

success(X):- tryhard(X),student(X).

poor(X):- lazy(X),student(X).





### III. Ví dụ về cơ sở tri thức

- File cơ sở tri thức: kiemhiepl.pl.
- File test: questions.txt.
- Giới thiệu về cơ sở tri thức:
  - + Dựa trên quan hệ giữa các thành viên trong một “bang phái” theo cách xưng hô của phim kiểm hiệp.
- 1. Mô tả các quan hệ trong phả hệ môn phái.
  - Phụ nữ được gọi là nữ nhân.
  - Đàn ông được gọi là nam nhân.
  - Nam nhân và nữ nhân lấy nhau gọi là đã thành hôn.
  - Người đứng đầu môn phái gọi là sư tổ.
  - Huynh đệ (Anh, em) là hai người đàn ông không nhất thiết cùng dòng máu, người em là sư đệ, người anh là sư huynh và hai người này có cùng một người thầy, gọi là sư phụ.
  - Sư đệ gọi vợ của sư huynh là tẩu tẩu.
  - Sư huynh gọi vợ của sư đệ là đệ muội.
  - Sư huynh của sư phụ gọi là sư bá, em của sư phụ gọi là sư thúc.
  - Vợ được chồng gọi là nương tử, vợ gọi chồng là tướng công.

- Vợ của sư phụ gọi là sư mẫu, vợ của sư bá hay sư thúc gọi là sư nương.
- Người đứng đầu môn phái gọi là sư tổ.
- Vợ của sư tổ gọi là tổ sư bà.

## 2. Các thành viên trong phả hệ.

- Thế hệ 1.
  - Đình Thục: Sư tổ
  - Tổ sư bà: Unknown
- Thế hệ 2:
  - Trọng Nghĩa: Đệ tử của Đình Thục.
  - Trung Nhân: Nương tử của Trọng Nghĩa.
- Thế hệ 3:
  - Thành Long, Trung Nam, Minh Nguyên, Lương Nhân theo thứ tự người đầu là sư huynh của những người sau, tất cả đều là đệ tử của Trọng Nghĩa. Nương tử của 4 sư huynh đệ trên lần lượt là: Bùi Nguyên, Phúc Nguyên, Cao Phúc, Đức Năng.
- Thế hệ 4:
  - Đại Long, Danh Lưu, Công Minh theo thứ tự người đầu là sư huynh của những người sau, tất cả đều là đệ tử của Minh Nguyên. Nương tử của 3 sư huynh đệ trên lần lượt là: Trần Nam, Phương Nam, Đại Nghĩa.

## IV. Xây dựng chương trình suy diễn

### 1. Ý tưởng

- Sử dụng thuật toán suy diễn lùi tìm kiếm chứng minh bằng cách đệ quy theo chiều sâu: không gian tuyến tính theo kích thước của chứng minh.
- Được dùng nhiều trong ngôn ngữ lập trình logic.

### 2. Cách thức thực hiện

- **Không đầy đủ do lặp vô tận => Giải pháp: kiểm tra giải pháp hiện tại với mọi trạng thái đang có trong Stack (Là một danh sách chứa các câu hỏi đích)**
- **Không hiệu quả do mục tiêu con bị lặp lại (cả khi thất bại cũng như khi thành công) => Giải pháp: dùng bộ nhớ tạm lưu lại các mục tiêu con đã duyệt qua.**
- **Trong cơ sở tri thức các phép hội(';') trong câu thì được tách ra thành các câu đơn(các dòng) chỉ gồm các phép tuyển(',') có dạng chuẩn  $(p1 \wedge \dots \wedge pn \Rightarrow q)$ .**
- **Cơ sở tri thức được lưu dưới dạng danh sách liên kết, kết luận của cơ sở tri thức có con trỏ trỏ đến tiền đề suy ra nó.**
- **Các vị từ được định nghĩa là các object(thuộc class PredicateLevel2).**

### TÀI LIỆU THAM KHẢO

<https://voer.edu.vn/c/lap-trinh-logic/7a0aeef9/d427e176>.

<https://vi.wikipedia.org/wiki/Prolog>.