

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH



BÁO CÁO ĐỒ ÁN
MÔN NGUYÊN LÝ VÀ PHƯƠNG PHÁP LẬP TRÌNH
Đề tài: Lập trình Logic với Prolog

GVHD: ThS. Trịnh Quốc Sơn

Nhóm sinh viên thực hiện: Nhóm 7

- | | |
|---------------------|----------------|
| 1. Huỳnh Trọng Khoa | MSSV: 18520918 |
| 2. Nguyễn Quan Huy | MSSV: 19521622 |

☞ Tp. Hồ Chí Minh, 7/2021 ☞

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

[illegible]

....., ngày.....tháng.....năm 2021

Người nhận xét

(Ký tên và ghi rõ họ tên)

BẢNG PHÂN CÔNG, ĐÁNH GIÁ THÀNH VIÊN:*Bảng 1: Bảng phân công, đánh giá thành viên*

Họ và tên	MSSV	Phân công	Đánh giá
Huỳnh Trọng Khoa	18520918	<ul style="list-style-type: none"> - Tìm hiểu nội dung về các cú pháp trong Prolog. - Tìm hiểu nguyên lý hoạt động của Prolog. - Thực hiện demo chương trình. - Thiết kế slide. - Viết báo cáo. - Thuyết trình. 	Hoàn thành
Nguyễn Quan Huy	19521622	<ul style="list-style-type: none"> - Tìm hiểu lịch sử, giới thiệu về ngôn ngữ Prolog. - Tìm hiểu nội dung về cú pháp trong Prolog. - Viết báo cáo. 	Hoàn thành

DANH MỤC CÁC BẢNG, HÌNH ẢNH

Danh mục các bảng:

Bảng 1. Bảng phân công, đánh giá thành viên.....	3
Bảng 2. Danh sách các vị từ hỗ trợ cấu trúc danh sách	10
Bảng 3. Các phép toán sử dụng trong Prolog.....	11

Danh mục hình ảnh:

Hình 1. Logo chính thức của ngôn ngữ Prolog	6
Hình 2. Mô tả quá trình so khớp ví dụ 1a	13
Hình 3. Mô tả quá trình so khớp ví dụ 1b	14
Hình 4. Giao diện chính của phần mềm SWI-Prolog	15
Hình 5. Giao diện lập trình Prolog	16
Hình 6. Giao diện trên trình duyệt web.....	16
Hình 7. Chọn nơi lưu trữ file prolog	17
Hình 8. Cách viết các vị từ trên prolog	18
Hình 9. Thông dịch prolog thành công	20
Hình 10. Truy vấn trên Prolog	20

MỤC LỤC

DANH MỤC CÁC BẢNG, HÌNH ẢNH.....	4
Chương 1: GIỚI THIỆU VỀ NGÔN NGỮ PROLOG	6
1.1.Vài nét sơ lược về lịch sử hình thành ngôn ngữ Prolog.....	6
1.2.Nguyên lý lập trình của Prolog	6
Chương 2: CÚ PHÁP TRONG PROLOG	7
2.1.Biểu diễn các đối tượng	7
2.2.Quan hệ giữa các đối tượng	7
2.3.Sự kiện.....	8
2.4.Luật	8
2.5.Truy vấn	9
2.6.Cấu trúc danh sách	9
2.7.Các thành phần khác	10
Chương 3: CÁC NGUYÊN TẮC TRONG PROLOG	12
3.1.Quy tắc đồng nhất (UNIFICATION).....	12
3.2.Đệ quy (RECURSION).....	14
Chương 4: CÁCH THỰC THI CHƯƠNG TRÌNH	15
4.1.Giới thiệu phần mềm.....	15
4.2.Xây dựng một chương trình.....	16
Chương 5: KẾT LUẬN	22
5.1.Uưu điểm.....	22
5.2.Nhược điểm.....	22
TÀI LIỆU THAM KHẢO	22

Chương 1: GIỚI THIỆU VỀ NGÔN NGỮ PROLOG

1.1 Vài nét sơ lược về lịch sử hình thành ngôn ngữ Prolog

Prolog là ngôn ngữ lập trình được xuất phát từ cụm từ tiếng Pháp “*Programmation en logique*” có nghĩa là lập trình theo logic. Xuất hiện từ năm 1972 do Alain Colmerauer và Ropert 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.



Hình 1. Logo chính thức của ngôn ngữ Prolog

1.2 Nguyên lý lập trình của Prolog

Prolog được sử dụng phổ biến trong lĩnh vực trí tuệ nhân tạo, đặc biệt là xử lý ngôn ngữ tự nhiên. Nguyên lý lập trình logic dựa trên các mệnh đề Horn. Một mệnh đề Horn biểu diễn một sự kiện hay một sự việc nào đó là đúng hoặc không đúng, xảy ra hoặc không xảy ra (có hoặc không có, v.v...).

Ví dụ: Một số câu tiếng Việt về mệnh đề Horn:

1. Nếu Nam đạt điểm cao thì Nam sẽ nhận học bổng.
2. Tất cả mọi người sẽ chết.
3. Hoa là người hạnh phúc.

Ở các mệnh đề trên, mệnh đề (1),(2) được gọi là luật (rule), mệnh đề (3) được gọi là sự kiện (fact).

Một chương trình Prolog có thể được xem như là một cơ sở dữ liệu gồm các mệnh đề Horn, hoặc dạng luật hoặc dạng sự kiện. Người sử dụng sẽ chạy chương trình logic bằng cách đặt câu hỏi (question/query) truy vấn trên cơ sở dữ liệu này.

Ví dụ: Một số câu hỏi tiếng Việt:

1. Ai là người hạnh phúc ?
2. Hoa có phải là người hạnh phúc ?

Chương trình logic sẽ tìm kiếm, chứng minh câu hỏi là một khẳng định là đúng hay sai hoặc kết quả cụ thể nào đó. Qua những thông tin trên, chúng ta cũng có thể biết được cách một chương trình logic được thực hiện như thế nào.

Chương 2: CÚ PHÁP TRONG PROLOG

2.1 Biểu diễn các đối tượng

Các đối tượng trong Prolog bao gồm các hằng và biến, hằng mang giá trị cho sẵn ở đầu chương trình, các biến có giá trị thay đổi sẽ được gán giá trị khi chạy chương trình. Tên biến là một ký tự hoa hoặc chuỗi ký tự bắt đầu bằng chữ cái in hoa. Biến không cần khai báo kiểu dữ liệu.

Ví dụ: Biểu diễn một số vị từ trong ngôn ngữ prolog như sau:

hocsinh(nam).

hocsinh(X).

Diễn giải: Trong 2 vị từ trên, “nam” được gọi là hằng, “X” được gọi là biến.

2.2 Quan hệ giữa các đối tượng

Quan hệ giữa các đối tượng được dùng dưới hình thức vị từ, các vị từ bao gồm tên của vị từ và các đối số. Các đối số được đặt trong ngoặc và phân cách nhau bởi dấu phẩy.

Ví dụ: Biểu diễn một số vị từ trong ngôn ngữ prolog như sau:

quen(X,Y).

hocsinh(nam).

Diễn giải: Ở vị từ đầu tiên, ta có thể định nghĩa cho câu là “X quen Y”, trong đó X,Y là 2 đối tượng và quen vị từ được xem là quan hệ giữa 2 đối tượng đó, cụ thể có thể dịch sang ngôn ngữ tự nhiên là “X quen Y”. Đối với vị từ thứ 2, nam là hằng và vị từ hocsinh là quan hệ của đối tượng “nam”, cụ thể có thể dịch sang ngôn ngữ tự nhiên là “nam là học sinh”.

2.3 Sự kiện

Là các phát biểu chân lý về một vấn đề, thay cho các chỉ thị để giải quyết vấn đề. Hệ thống thực thi của Prolog sẽ sử dụng các phát biểu này để tìm lời giải của bài toán bằng cách kiểm tra từng trường hợp của bài toán. Được định nghĩa bằng một định danh và một bộ các hằng. Một định danh quan hệ được gọi là một vị từ.

Cú pháp khai báo:

- Tên quan hệ hay đối tượng phải bắt đầu bằng ký tự thường.
- Tên quan hệ viết trước (thông thường quan hệ chính là vị từ của câu).
- Các đối tượng được viết cách nhau bằng dấu phẩy và được bao bởi cặp dấu ngoặc đơn.
- Khai báo phải kết thúc bằng dấu chấm.

Ví dụ biểu diễn trong Prolog:

hoa.	Đối tượng: ký hiệu là hoa
minh.	Đối tượng: ký hiệu là minh
quen(hoa,minh).	Quan hệ: hoa quen với minh
Ngân đẹp.	Biểu diễn: đẹp(ngân).
Minh có một quyển sách	Biểu diễn: có(minh,quyển_sách).

2.4 Luật

Cho biết một vị từ sẽ thỏa trong những điều kiện nào, luật được cấu tạo bởi các vị từ, gọi là hạng tử, theo dạng:

A :- hạng_tử_1, hạng_tử_2, ..., hạng_tử_n

Ý nghĩa:

Nếu các hạng tử hạng_tử_1, ..., hạng_tử_n thỏa thì hạng tử A sẽ thỏa.

Cú pháp của luật:

- <Đầu luật> :- <Thân luật>
- :- đọc là nếu

- Một luật được kết thúc bằng dấu chấm.

Một số dạng phổ biến của luật:

- **Điều kiện if-then:**

Nếu Hoa quen Minh thì Hoa là bạn của Minh

Biểu diễn trong prolog: bạn(minh, hoa) :- quen(hoa, minh).

- **Tổng quát hóa:**

Mọi người đều phải chết

Biểu diễn trong prolog: chết(X) :- người(X).

- **Định nghĩa:**

Chim là động vật có cánh và chân

Biểu diễn trong prolog: chim(X):- động_vật(X), có(X, cánh), có(X, chân).

2.5 Truy vấn

Dùng để yêu cầu Prolog giải quyết một vấn đề nào đó. Có hai dạng truy vấn:

- Truy vấn cơ sở: là truy vấn không chứa biến, kết quả trả về là một câu trả lời đúng/sai.

Ví dụ: quen(hoa, minh). → true/false

- Truy vấn phi cơ sở: là truy vấn có chứa biến, kết quả là giá trị của các biến trong truy vấn đó.

Ví dụ: quen(X, Y). → X = ?, Y = ?

2.6 Cấu trúc danh sách

Danh sách là kiểu cấu trúc dữ liệu được sử dụng rộng rãi trong các ngôn ngữ lập trình phi số. Một danh sách là một dãy bất kỳ các đối tượng. Khác với kiểu dữ liệu tập hợp, các đối tượng của danh sách có thể trùng nhau (xuất hiện nhiều lần) và mỗi vị trí xuất hiện của đối tượng đều có ý nghĩa.

Danh sách là cách diễn đạt ngắn gọn của kiểu dữ liệu hạng phức hợp trong Prolog. Hàm tử của danh sách là dấu chấm “.”. Do việc biểu diễn danh sách bởi hàm tử này có thể tạo

CS111.L21.KHCL - Nguyên lý và phương pháp lập trình

ra những biểu thức mập mờ, nhất là khi xử lý các danh sách gồm nhiều phần tử lồng nhau, cho nên Prolog quy ước đặt dãy các phần tử của danh sách giữa các cặp móc vuông “[]”.

Cách khai báo khác: [**head** | **tail**], Thành phần thứ nhất, được gọi là đầu (head) của danh sách. 2. Thành phần thứ hai, phần còn lại của danh sách (trừ ra phần đầu), được gọi là đuôi (tail) của danh sách, cũng là một danh sách.

Một số vị từ xử lý danh sách khác:

<i>Vị từ</i>	<i>Ý nghĩa</i>
append(List1, List2, List3)	Ghép hai danh sách List1 và List2 thành List3.
member(Elem, List)	Kiểm tra Elem có là phần tử của danh sách List hay không, nghĩa là Elem hợp nhất được với một trong các phần tử của List.
nextto(X, Y, List)	Kiểm tra nếu phần tử Y có đứng ngay sau phần tử X trong danh sách List hay không.
delete(List1, Elem, List2)	Xoá khỏi danh sách List1 những phần tử hợp nhất được với Elem để trả về kết quả List2.
select(Elem, List, Rest)	Lấy phần tử Elem ra khỏi danh sách List để trả về những phần tử còn lại trong Rest, có thể dùng để chèn một phần tử vào danh sách.
sumlist(List, Sum)	Tính tổng các phần tử của danh sách List chứa toàn số để trả về kết quả Sum
numlist(Low, High, List)	Nếu Low và High là các số sao cho $Low \leq High$, thì trả về danh sách $List = [Low, Low+1, ..., High]$.

Bảng 2. Danh sách các vị từ hỗ trợ cấu trúc danh sách

2.7 Các thành phần khác

a) Phép toán logic

Trong Prolog, phép toán và (AND) được dùng để kết hợp các vị từ được biểu diễn bằng một dãy các vị từ cách nhau bằng dấu phẩy “,”.

Ví dụ: biểu diễn ngôn ngữ tự nhiên sang một vị từ

CS111.L21.KHCL - Nguyên lý và phương pháp lập trình

Hoa hậu là phụ nữ đẹp \rightarrow hoa_hậu(X):- phụ_nữ(X),đẹp(X).

Đối với phép hoặc (OR) các vị từ được biểu diễn bằng một dãy các vị từ đó tách nhau bằng dấu chấm phẩy “;”.

Ví dụ: Minh biết Thúy nếu Thúy là bạn của Hải hoặc Hiếu.

Được biểu diễn dưới dạng logic vị từ: biết(minh, thúy) :- bạn(thúy, hải); bạn(thúy, hiếu)

b) Cú pháp chú thích

Nội dung cần được chú thích trong chương trình có thể được đặt trong cặp câu /* ... */ hoặc sau ký tự %.

c) Phép phủ định

Phép phủ định được sử dụng để thay đổi chân trị của một vị từ từ true sang false hoặc ngược lại.

Ví dụ: not(quen(hoa,hải)) được định nghĩa là hoa không quen hải.

d) Phép gán

Phép gán được sử dụng thông qua từ “is”, được sử dụng để tính toán các phép tính số học.

Ví dụ: cong(A,B):- D is A+B,write(D).

Khi truy vấn cong(3,4) thì lúc này biến D được gán là giá trị tổng của A và B là bằng 7.

e) Phép so sánh

Các phép so sánh được sử dụng trong prolog bao gồm:

Phép toán	Ý nghĩa
<	Nhỏ hơn
<=	Nhỏ hơn hoặc bằng
>	Lớn hơn
>=	Lớn hơn hoặc bằng
=\=	Khác nhau
::=	Giống nhau

Bảng 3. Các phép toán sử dụng trong Prolog

f) Phép hợp nhất

Phép hợp nhất được sử dụng thông qua kí hiệu "=", cho phép thay đổi giá trị thành một chuỗi các giá trị bên phải của phép hợp nhất.

Ví dụ: $\text{cong}(A,B):-D = A+B, \text{write}(D).$

Ở đây ta có 1 vị từ "cong" như ở ví dụ phép gán, tuy nhiên khi dùng phép hợp nhất, kết quả trả về $\text{cong}(3,4)$ là $3+4$.

Chương 3: CÁC NGUYÊN TẮC TRONG PROLOG

3.1 Quy tắc đồng nhất (UNIFICATION)

Hợp nhất (đồng nhất) là một cơ chế cơ bản trong Prolog trong quá trình tìm kiếm kết quả. Kết quả của truy vấn được thực hiện nhờ quá trình so khớp. Hợp nhất là cách so khớp trong Prolog hai sự kiện khớp với nhau nếu vị từ và các tham số của chúng giống nhau. Nếu so khớp được, prolog sẽ trả kết quả là yes, ngược lại là no. "no" có nghĩa là không chứng minh được từ những sự kiện đã cho.

Ví dụ: cho các sự kiện được biểu diễn dưới dạng Prolog như sau:

$\text{bạn}(\text{hoa}, \text{nam}).$

$\text{bạn}(\text{nam}, \text{hải}).$

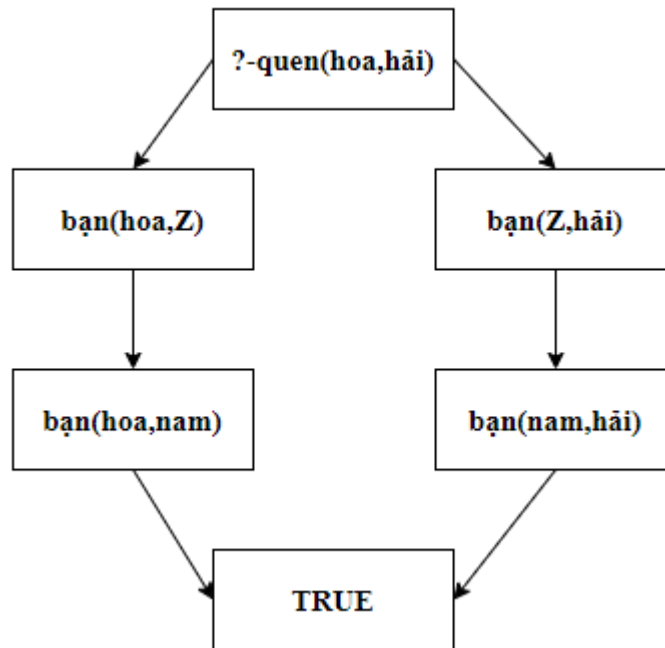
$\text{quen}(X, Y) :- \text{bạn}(X, Z), \text{bạn}(Z, Y).$

Thực hiện truy vấn các truy vấn:

a) $\text{quen}(\text{hoa}, \text{hải}).$

b) $\text{quen}(\text{hoa}, X).$

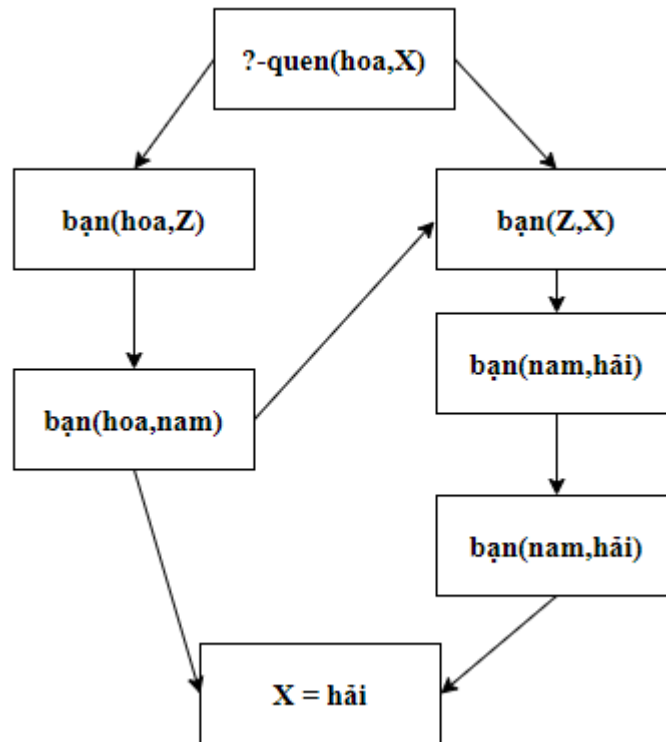
Xét Trường hợp truy vấn $\text{quen}(\text{hoa}, \text{hải})$, kết quả trả về sẽ là True/False:



Hình 2. Mô tả quá trình so khớp ví dụ 1a

Đầu tiên chương trình sẽ sử dụng vị từ “quen” để so khớp với vị từ “quen” trong các sự kiện nếu có. Tiếp theo là đồng nhất các biến và hằng với nhau, cụ thể “hoa” tương đương với “X” và “hải” tương đương với “Y”. Tiếp đó, khi bắt gặp vị từ “bạn”, chương trình tiếp tục thực hiện so khớp với các vị từ “bạn” để tìm đồng nhất biến “Z”. Khi đã tìm được “Z” là “nam” thì 2 vị từ bạn sẽ “AND” với nhau và cho ra kết quả cuối cùng (ở đây 2 vị từ “bạn(hoa,nam) và bạn(nam,hải) có giá trị True).

Xét trường hợp truy vấn quen(hoa,X), kết quả sẽ trả về giá trị của X.



Hình 3. Mô tả quá trình so khớp ví dụ 1b

Ở trường hợp này, vấn đề so khớp tương tự ở ví dụ trên, nhưng đối với vị từ “bạn(Z,X)”, vị từ này tồn tại 2 biến, đồng nghĩa với việc chương trình sẽ phải so khớp với toàn bộ vị từ bạn có xuất hiện trong tập sự kiện, tuy nhiên khi vị từ bạn(hoa,Z) tìm được Z = nam thì chương trình sẽ thay thế biến Z cho vị từ bạn(Z,X) để tìm ra kết quả cuối cùng X = hải.

3.2 Độ quy (RECURSION)

Đệ quy trong ngôn ngữ Prolog cũng là một cơ chế tự động trong quá trình tìm kiếm kết quả. Sử dụng đệ quy khi một luật được định nghĩa nhờ vào chính luật đó. Một thao tác đệ quy bao gồm 2 phần:

- Điều kiện dừng: Một thao tác sơ cấp (đã biết cách thực hiện trực tiếp)
- Một lời gọi đệ quy, sao cho sau một số hữu hạn lần gọi đệ quy sẽ phải dẫn đến điều kiện dừng.

Ví dụ: Tính giai thừa (Factor(N)):

- Điều kiện dừng: $N = 0$: factor(N) = 1.
- Lời gọi đệ quy: $N > 0$: factor(N) = Factor(N-1).N

Biểu diễn trong prolog:

`gt(0,1):-!.`

`gt(N,K):-N1 is N-1, gt(N1,K1), K is N*K1.`

Đối với điều kiện dừng, chúng ta sử dụng kí tự (!) hay được gọi là lát cắt để tạo ra điều kiện dừng cho một biểu thức đệ quy. Chương trình sẽ tính giai thừa theo thứ tự từ N giảm dần về 0, và khi $N = 0$ thì chương trình sẽ dừng lại.

Ngoài ra lát cắt (!) còn được dùng đối với bài toán có nhiều đáp án nhưng chỉ lấy đáp án đầu tiên.

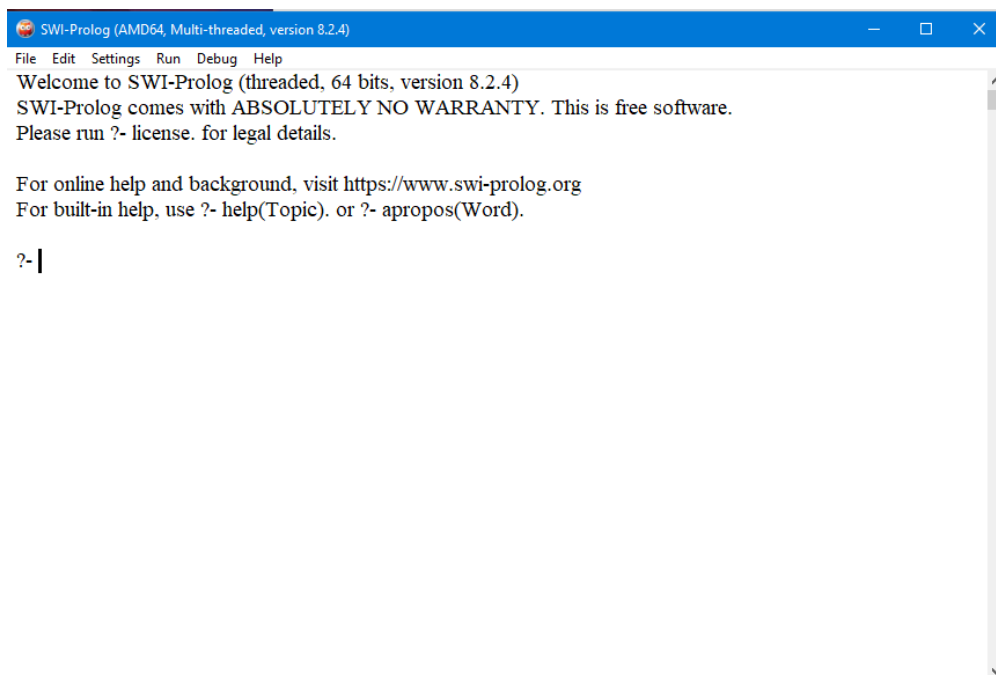
Chương 4: CÁCH THỰC THI CHƯƠNG TRÌNH

4.1 Giới thiệu phần mềm

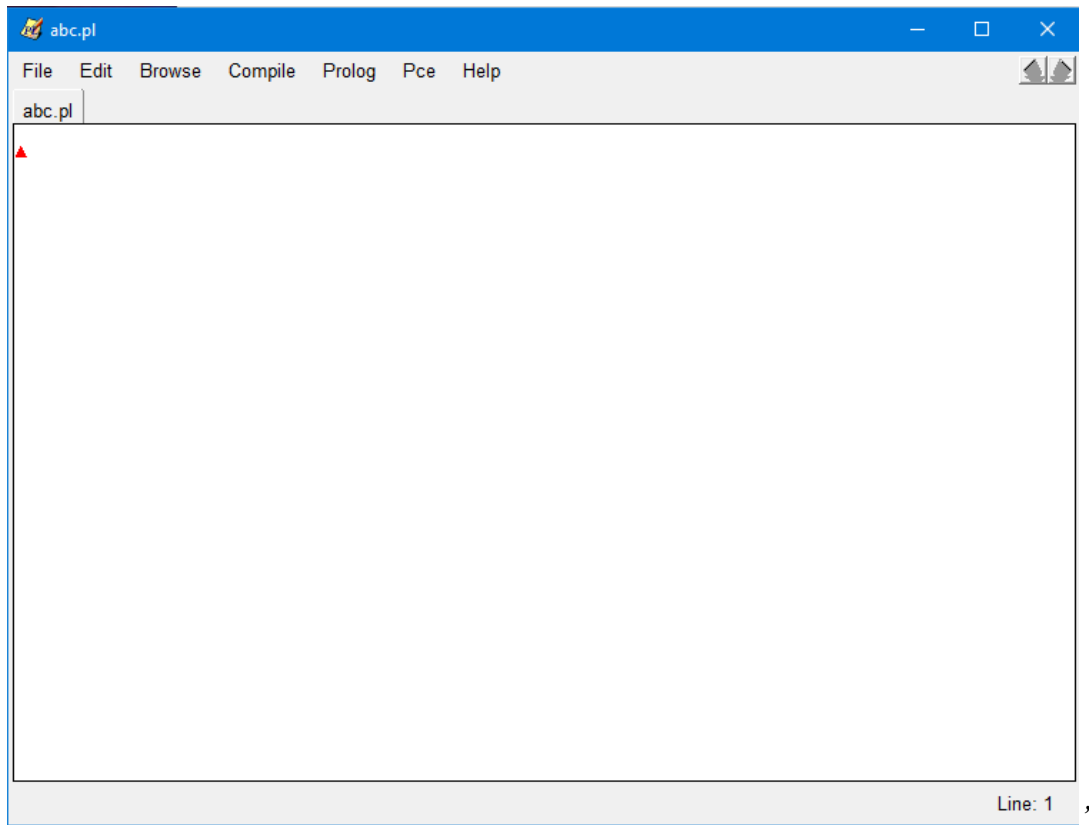
SWI-Prolog là dạng thực thi mã nguồn mở của ngôn ngữ lập trình Prolog. Được thiết kế và phát triển bởi Jan Wielemaker vào năm 1987.

Phần mềm có thể được sử dụng trên hệ điều hành Windows, Linux, MacOS.

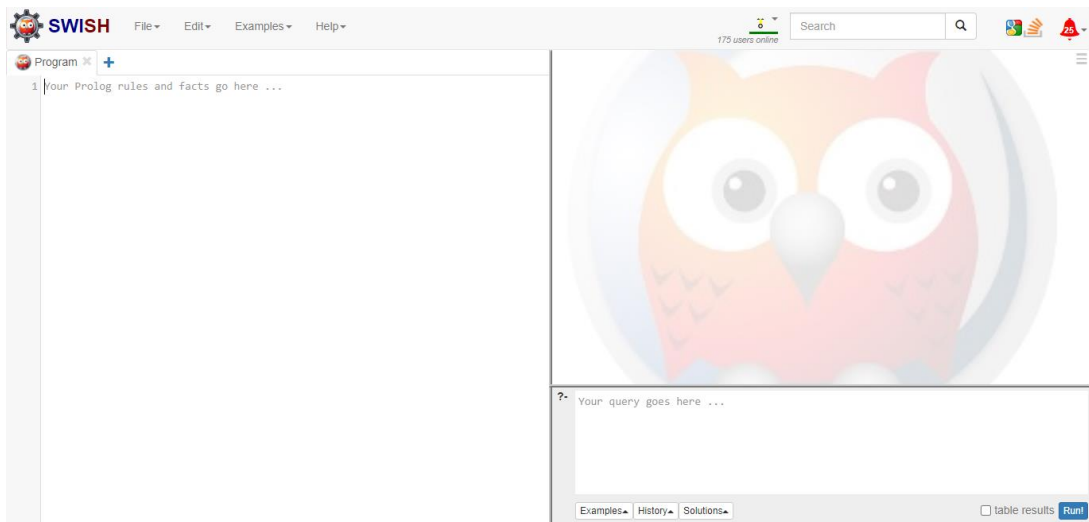
Ngoài ra còn có thể sử dụng ngay trên web: <https://swish.swi-prolog.org/>



Hình 4. Giao diện chính của phần mềm SWI-Prolog



Hình 5. Giao diện lập trình Prolog



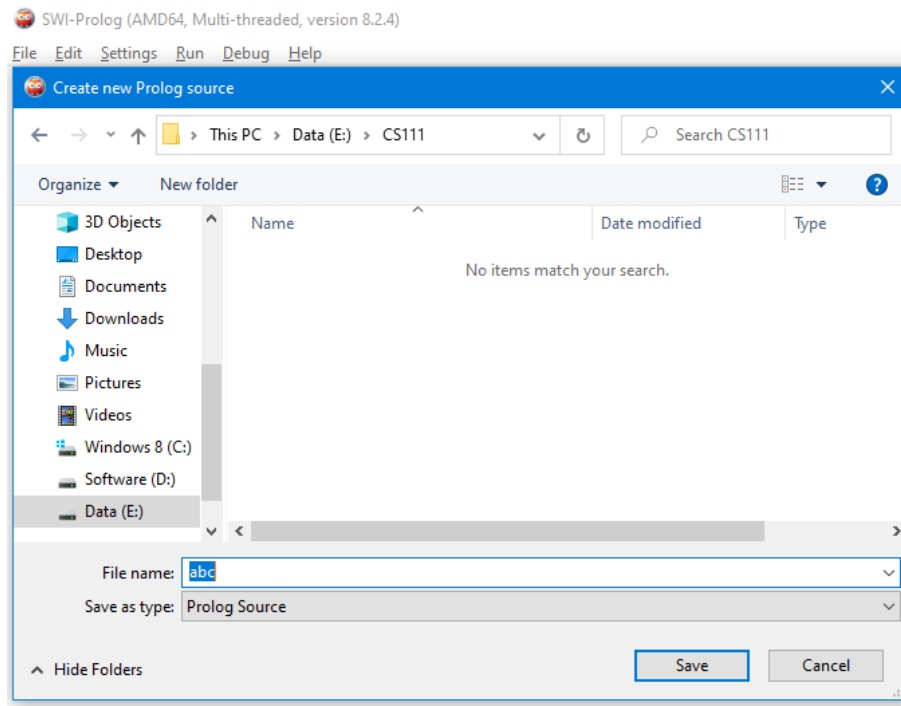
Hình 6. Giao diện trên trình duyệt web

4.2 Xây dựng một chương trình

Để xây dựng một chương trình Prolog và truy vấn trên bộ dữ liệu đó, thì trước hết chúng ta phải tạo một file chứa các đối tượng, vị từ tùy thuộc vào bài toán. Ta thực hiện các bước sau:

CS111.L21.KHCL - Nguyên lý và phương pháp lập trình

Bước 1: Ở giao diện chính của phần mềm, chọn File → New và chọn ổ đĩa nơi lưu trữ file prolog.



Hình 7. Chọn nơi lưu trữ file prolog

Bước 2: Khi đã có một file prolog thì chúng ta sẽ code ở trong file mới vừa được tạo ra như ảnh ví dụ dưới đây.

```

18520918.pl
File Edit Browse Compile Prolog Pce Help
abc.pl 18520918.pl
%%      Câu 1: Đếm số lượng phần tử trong mảng A
dodai([],0):-!.
dodai([_|L],N) :-
    length(L,N1),
    N is N1 + 1.
%%      Câu 2: Đếm số lượng phần tử có giá trị là số lẻ
demsole([],0):-!.
demsole([X|L],N) :- X mod 2 =\= 0,
    demsole(L,M),
    N is M + 1,!.
demsole([_|L],N) :- demsole(L,N).
%%      Câu 3: Đếm số lượng phần tử có giá trị là số chẵn
demsochan([],0):-!.
demsochan([X|L],N) :- X mod 2 == 0,
    demsochan(L,M),
    N is M + 1,!.
demsochan([_|L],N) :- demsochan(L,N).
%%      Câu 4: Tìm phần tử có giá trị là số chẵn
timsochan([],0).
timsochan([X|L],N) :- X mod 2 == 0,
    write(X),write(' '),
    timsochan(L,M),
    N is M + 1,!.
comment(structured)
Line: 13

```

Hình 8. Cách viết các vị từ trên prolog

Đoạn code Prolog xử lý danh sách:

```

%%      Câu 1: Đếm số lượng phần tử trong mảng A
dodai([],0):-!.
dodai([_|L],N) :-length(L,N1),N is N1 + 1.
%%      Câu 2: Đếm số lượng phần tử có giá trị là số lẻ
demsole([],0):-!.
demsole([X|L],N) :- X mod 2 =\= 0,demsole(L,M),N is M + 1,!.
demsole([_|L],N) :-demsole(L,N).
%%      Câu 3:Đếm số lượng phần tử có giá trị là số chẵn
demsochan([],0):-!.
demsochan([X|L],N) :- X mod 2 == 0,demsochan(L,M), N is M + 1,!.
demsochan([_|L],N) :- demsochan(L,N).
%%      Câu 4:Tìm phần tử có giá trị là số chẵn
timsochan([],0).
timsochan([X|L],N) :- X mod 2 == 0,write(X),write(' '),timsochan(L,M),N is M + 1,!.
timsochan([X|L],N) :- X mod 2 =\= 0,timsochan(L,N).

```

CS111.L21.KHCL - Nguyên lý và phương pháp lập trình

%% Câu 5: Tìm phần tử có giá trị là số lẻ

timsole([],0).

timsole([X|L],N) :- X mod 2 \= 0, write(X), write(' '), timsole(L,M), N is M + 1, !.

timsole([X|L],N) :- X mod 2 == 0, timsole(L,N).

%% Câu 6: Tìm phần tử có giá trị lớn nhất

max([N],[N]):-!.

max([X|L],M):- max(L,M), M >= X, !.

max([X|L],X):- max(L,M), X > M.

%% Câu 7: Tìm phần tử có giá trị nhỏ nhất

min([N],[N]):-!.

min([X|L],M):- min(L,M), M <= X, !.

min([X|L],X):- min(L,M), X < M.

%% Câu 8: Tính tổng các phần tử trong danh sách

sum([],0).

sum([X|L],N):- sum(L,M), N is M + X.

%% Câu 9: Sắp xếp các phần tử trong danh sách theo thứ tự giảm dần

insert1(N,[],[N]):-!.

insert1(N,[X|L],[N|[X|L]]):- N >= X, !.

insert1(N,[X|L],[X|L1]):-insert1(N,L,L1).

giamdan([N],[N]):-!.

giamdan([N|L],L2):-giamdan(L,L1),insert1(N,L1,L2).

%% Câu 10: Sắp xếp các phần tử trong danh sách theo thứ tự tăng dần

insert2(N,[],[N]):-!.

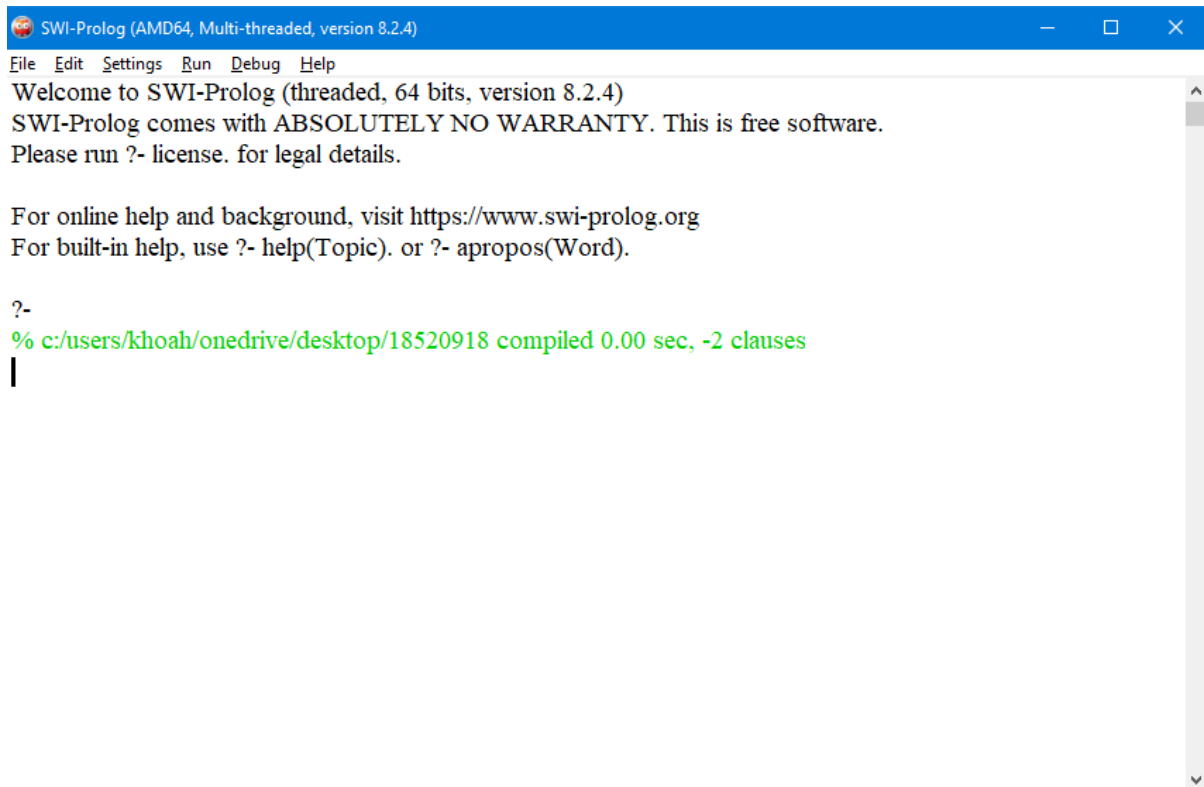
insert2(N,[X|L],[N|[X|L]]):- N <= X, !.

insert2(N,[X|L],[X|L1]):-insert2(N,L,L1).

tangdan([N],[N]):-!.

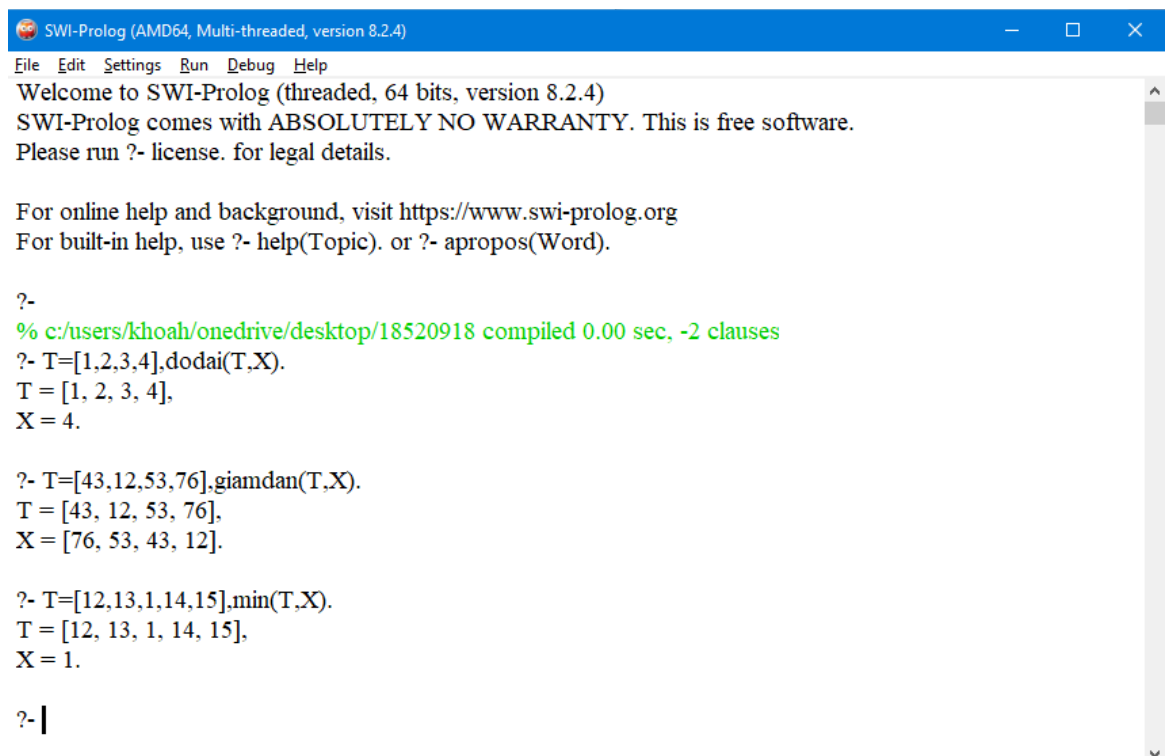
tangdan([N|L],L2):-tangdan(L,L1),insert2(N,L1,L2).

Bước 3: Tiến hành thông dịch chương trình bằng 2 cách. Cách 1: Chọn Compile → Compile buffer hoặc cách thứ 2 ở giao diện chính, chọn Consult và chọn file prolog cần được thông dịch. Khi thông dịch thành công sẽ xuất hiện dòng chữ màu xanh.



Hình 9. Thông dịch prolog thành công

Bước 4: Thực hiện truy vấn thông qua các vị từ và biến được định nghĩa ở bước trước.



Hình 10. Truy vấn trên Prolog

CS111.L21.KHCL - Nguyên lý và phương pháp lập trình

Đoạn code Prolog thực hiện truy vấn: (thực hiện sau dấu “?-“ và kết quả trả về tương ứng.

?- dodai([1,2,3,4],X).

X = 4.

?- demsole([1,2,3,4],X).

X = 2.

?- demsochan([1,2,3,4,5,6,7],X).

X = 3.

?- timsochan([1,2,3,4,5,6,7],X).

2 4 6

X = 3.

timsole([1,2,3,4,5,6,7],X).

1 3 5 7

X = 4.

?- max([432,545,23,1,2],X).

X = 545.

?- min([432,545,23,1,2],X).

X = 1.

?- sum([314,545,65,6],X).

X = 930.

?- giamdan([43,13,65,21,11],X).

X = [65, 43, 21, 13, 11].

?- tangdan([43,13,65,21,11],X).

X = [11, 13, 21, 43, 65].

Chương 5: KẾT LUẬN

Thông qua quá trình tìm hiểu về ngôn ngữ Prolog cũng như cách hoạt động của ngôn ngữ thì nhóm em đã rút ra một số kết luận về ưu, nhược điểm của ngôn ngữ như sau:

5.1 Ưu điểm

- Dễ dàng lập trình, không có nhiều cấu trúc chương trình phức tạp.
- Tìm kiếm kết quả dựa trên cơ chế mặc định là quay lui và khớp mẫu.
- Cấu trúc danh sách được xây dựng sẵn bao gồm các phương thức đi cùng.
- Có thể kết nối với Java, C#,... để xây dựng một hệ chuyên gia có giao diện hoàn chỉnh.

5.2 Nhược điểm

- Khó khăn trong việc mô tả dữ liệu đầu vào và đầu ra. Đặc biệt là với chương trình xử lý ngôn ngữ tự nhiên.
- Không hỗ trợ xử lý tập tin. Đây là nhược điểm lớn nhất khi thế mạnh của Prolog là xử lý logic liên quan đến từ ngữ nhưng quá trình truy vấn phải thực hiện thủ công.
- Không hỗ trợ sắp xếp vị từ.

TÀI LIỆU THAM KHẢO

1. PGS.TS. Phan Huy Khánh, “*Lập trình logic trong Prolog*” 2004. [Trực tuyến]. Địa chỉ: <https://masterwed.files.wordpress.com/2010/07/ltprolog123.pdf> [Truy cập lần cuối 29/6/2021].
2. "Logic Programming – Làm quen với một vài ví dụ," *Trí tuệ nhân tạo*, 2020.[Trực tuyến].Địa chỉ: <https://trituenhantao.io/lap-trinh/logic-programming-lam-quen-voi-mot-vai-vi-du/> [Truy cập lần cuối: 04/07/2021].