Bài thực hành
Chuyên đề ngôn ngữ lập trình KHMT
Bộ môn KHMT K.CNTT&TT T. Đại học Cần Thơ
Giảng viên: Nguyễn Bá Diệp

Phần 1 Prolog cơ bản

1) Hợp nhất với "="

Hãy thử nhập vào Prolog các lệnh sau đây, ghi nhận kết quả, so sánh kết quả và giải thích kết quả:

a) xe = honda.
b) xe = xehoi
c) honda = xemay.
d) toyota = honda.

2) Hợp nhất atom và biến với vị từ =

Hãy thử nhập vào Prolog các lệnh sau đây, ghi nhận kết quả, so sánh kết quả và giải thích kết quả:

a) Xe = honda.
b) honda = Xe.
c) Xe = honda, write(xe).
d) Write(Xe).
e) Toyota=Xe,Toyota=xe4banh.
f) write(Xe),Toyota=xe4banh,Toyota=Xe,write(Xe).
g) Xe = honda,toyota = Xe.
h) Xe = honda. Xe = toyota.
i) Xe = honda.
   Xe = toyota.

3) Hợp nhất biến (tt)

Cho biết các câu lệnh sau câu lệnh nào thực hiện được và giá trị của biến khi kết thúc câu truy vấn.

a) Computer = sun.
b) Macintosh = great.
c) BBC = Useful.
d) Computer = sun, unix = Sun.
e) BUGATTI = champion.
f) Monkey = tamarin, Monkey = marmoset.
g) Cathedral = Ely, Ely = Worcester.

4) Kiểm tra giá trị của biến

Sử dụng var(Term)

var(Variable).
Car = bugatti, var(Car).
var(Car), Car = bugatti.
nonvar(unix).

Sử dụng nonvar(Term)

nonvar(Variable).
Car = bugatti, nonvar(Car).
nonvar(Car), Car = bugatti.
nonvar(unix).

5) Kiểm tra dữ liệu số các kiểu dữ liệu số sau đây

number(Term)	integer(Term)	float(Term)
number(Term). *number(1991).* *number(17.5).* *number(haimuoi).*	*integer(Term).* *integer(1991).* *integer(17.5).* *integer(haimuoi).*	*float(Term).* *float(1991).* *float(17.5).* *float(chin_phay_ba).*

6) Kiểm tra dữ liệu

Sử dụng atom(Term)

atom(sqUIrrEl).
atom(SQuiRReL).
atom((debit)).
atom('Robert Fitz Ralph (1190-1193)').
atom(===>).
atom([]).
atom([squirrel]).

Sử dụng atomic(Term)

atomic(Term).
atomic(breakfast).
atom(breakfast), atomic(breakfast).
number(1991), atomic(1991).
integer(1991), atomic(1991).
float(17.5), atomic(17.5).

7) Biểu diễn các câu sau đây thành dữ liệu sự kiện trong Prolog (có thể phải sử dụng đối số)

1. Fred ăn cam.
2. Fred ăn thịt khủng long.
3. Tony ăn táo.
4. John ăn táo.
5. John ăn nho.

Sau đó đặt các câu hỏi sau cho Prolog, quan sát kết quả và giải thích:

a. Fred có ăn cam ko?
b. John có ăn táo ko?
c. Mike có ăn táo ko?
d. Fred có ăn táo ko?

8) Viết vị từ tuổi để biểu diễn các tri thức sau trong Prolog

1. John 32 tuổi
2. Agnes 41 tuổi
3. George 72 tuổi
4. Ian 2 tuổi
5. Thomas 25 tuổi

Đặt các câu hỏi sau cho Prolog

a. Thomas bao nhiêu tuổi?
b. Có phải Agnes 41 tuổi ko?
c. Có phải Ian "hai" tuổi ko?

Phần 2: Cơ chế suy luận của Prolog

9) **Biểu diễn các tri thức sau thành sự kiện trong Prolog:**
   1. Toàn thích Mai,
   2. Dũng thích Mai,
   3. Mai thích Hiệp.
   4. Hiệp thích Dũng.
   5. Dũng thích Hùng.

Truy vấn Prolog các câu hỏi sau và vẽ cây tìm kiếm (AND/OR)
   a. Ai thích Mai ?
   b. Mai thích ai?
   c. Dũng thích ai?

10) **Cho cơ sở dữ liệu tri thức như sau:**

```
tape(1,van_morrison,astral_weeks,madam_george).
tape(2,beatles,sgt_pepper,a_day_in_the_life).
tape(3,beatles,abbey_road,something).
tape(4,rolling_stones,sticky_fingers,brown_sugar).
tape(5,eagles,hotel_california,new_kid_in_town).
```

a - Hãy truy vấn Prolog nội dung của băng số 5
b - Tìm thông tin băng có bài hát sgt_pepper

11) **Định nghĩa luật**

```
cool(X) :-
        red(X),
        car(X).
cool(X) :-
        blue(X),
        bike(X).
car(vw_beatle).
car(ford_escort).
bike(harley_davidson).
red(vw_beatle).
red(ford_escort).
blue(harley_davidson).
```

Vẽ cây tìm kiếm với câu hỏi “xe nào cool (ngầu)”

12) **Biểu diễn các phát biểu sau bằng Prolog. Chú ý các vị từ giống nhau phải được viết liên tiếp nhau:**
   a. Gà là thức ăn
   b. Táo là thức ăn
   c. Thứ mà có ai đó ăn vào mà vẫn còn sống cũng là thức ăn.
   d. Bill ăn đậu phọng.
   e. Bill còn sống.
   f. John ăn tất cả những gì là thức ăn.
   g. Sue ăn những thứ mà Bill ăn

Nạp chương trình và truy vấn:
   a. Cái gì là thức ăn ?
   b. John có ăn đậu phọng không ?
   c. Sue ăn gì ?
   d. Ai ăn gì ?

13) Biểu diễn các phát biểu sau bằng Prolog.

a. Marcus là người
b. Marcus là người Pompeian
c. Người Pomeian là người La mã
d. Ceasar là một lãnh chúa
e. Một người La mã nếu không trung thành với vị lãnh chúa sẽ ám sát ông ta.
f. Marcus không trung thành với Ceasar.

Nạp chương trình và truy vấn:

a. Marcus có ám sát Ceasar không ?
b. Ai là người La mã ?
c. Ai là lãnh chúa ?
d. Ai ám sát ai ?

Phần 3 Xây dựng chương trình Prolog

14)

a) Viết vị từ bigger/2 so sánh lớn hơn trả về số lớn nhất trong 2 số
b) Cải vị từ bigger/2 thành vị từ bigger/3 truyền ngược kết quả qua đối số thứ 3

15)

a - Viết chương trình giải phương trình bậc nhất trong Prolog
b - Viết chương trình kiểm tra n có phải là số nguyên tố?

Phần 4 Lập trình đệ quy

16) Định nghĩa vị từ *luythua* cho phép tính a lũy thừa n. Ví dụ: **luythua(2, 3, X)** sẽ cho kết quả X = 8.
17) Viết chương trình giải bài toán tháp Hà Nội.
18) Định nghĩa vị từ tính số fibonacy thứ N.
19) Định nghĩa vị từ tính tổ hợp chập k của n phần tử.
20) Viết chương trình tính ước chung lớn nhất của 2 số theo thuật toán Euclide
21) Viết chương trình tính bội chung nhỏ nhất của 2 số
22) Định nghĩa vị từ tính tổng n số lẻ liên tiếp
23) Định nghĩa vị từ tính tổng n số chẵn liên tiếp (Có thể viết vị từ tính tổng cho cả câu 22 và 23)
24) Định nghĩa vị từ tính tích n số chẵn liên tiếp
25) Định nghĩa vị từ tính tích n số lẻ liên tiếp (Có thể viết vị từ tính tổng cho cả câu 24 và 25)

Phần 5 Danh sách

26) Viết vị từ **demchan** đếm số phần tử là số chẵn trên một danh sách.
27) Viết vị từ **inptn** nhận đối số là một danh sách cùng số nguyên dương n và in ra phần tử thứ n trong danh sách.
28) Viết vị từ **ptgiua** in ra phần tử chính giữa danh sách số nguyên dương, trả về 0 nếu số phần tử của danh sách là số chẵn.
29) Viết vị từ **saptang** và **sapgiam** nhận đối số là một danh sách và sắp danh sách theo thứ tự tăng hoặc giảm
30) Viết vị từ **xenke**(L1,L2,L3) trong đó danh sách L3 được tạo nên bằng cách lấy xen kẽ các phần tử của L1 và L2 (vd: [L1a,L2a,L1b,L2b,L1c,L2c…]
31) Viết vị từ **xoaptdau** nhận đối số là một danh sách cùng số nguyên dương X và xóa phần tử X đầu tiên trong danh sách.

32) Viết vị từ **xoapt** nhận đối số là một danh sách cùng số nguyên dương X và xóa tất cả những phần tử bằng X trong danh sách.

33) Viết vị từ **thaythe**(X,Y,L,R) trong đó R là danh sách L mà tất cả các phần tử X đã được thay thế bằng Y

Phần 6 Xuất nhập

Phần 7 Tìm tất cả lời giải

(thực hành với các vị từ findall, setoff, bagof)

34) Tìm tất cả lời giải với 2 câu hỏi sau cho bài 9:

a) Ai thích Mai (tìm tất cả người thích Mai)

b) Dũng thích ai (tìm tất cả người Dũng thích)

35) Làm lại bài tập 10.

Tìm tất cả cuộn bang

36) Làm lại bài tập 11.

Tìm tất cả các xe nhìn ngầu (cool)

37) Làm lại bài tập 12

a) Tìm tất cả thức ăn.

b) Tìm tất cả thức ăn John ăn

Tìm kiếm trong không gian trạng thái

Sử dụng khung chương trình tìm kiếm theo chiều sâu, chiều rộng và cải tiến chương trình để in ra thông tin trong lúc tìm kiếm

```
/*định nghĩa các phép toán, bước chuyển trạng thái thông qua vị từ successor. Có thể cần một số ràng buộc về trạng thái mới được sinh ra hay bước chuyển có hợp lệ không*/

successor(State,NewState):- …

…

/*Vị từ succ gọi vị từ successor sinh ra trạng thái mới S từ trạng thái X và thêm S vào đường đi hiện hành*/

get_succ([State|Path], [NewState, State |Path]) :- successor(State,NewState), \+member(NewState, Path).

/*Giải thuật tìm kiếm theo chiều sâu*/

dfs(Goal, [[Goal|Path]|_], [Goal|Path]).

dfs(Goal, [[State|Path]|Rest], Solution) :- Goal \= State,

                        findall(NewState, get_succ([State|Path], NewState), Successor),

                        append(Successor, Rest, NewAgenda),

                        dfs(Goal, NewAgenda, Solution).

%% goi dfs(Goal,[[InitialState]], Solution).
```

```
/*định nghĩa các phép toán, bước chuyển trạng thái thông qua vị từ successor. Có thể cần một số ràng buộc về trạng thái mới được sinh ra hay bước chuyển có hợp lệ không*/

successor(State,NewState):- …

…

/*Vị từ succ gọi vị từ successor sinh ra trạng thái mới S từ trạng thái X và thêm S vào đường đi hiện hành*/

get_succ([State|Path], [NewState, State |Path]) :- successor(State,NewState), \+member(NewState, Path).

/*Giải thuật tìm kiếm theo chiều sâu*/

bfs(Goal, [[Goal|Path]|_], [Goal|Path]).

bfs(Goal, [[State|Path]|Rest], Solution) :- Goal \= State,

                        findall(NewState, get_succ([State|Path], NewState), Successor),

                        append(Rest, Successor, NewAgenda),

                        bfs(Goal, NewAgenda, Solution).

%% goi bfs(Goal,[[InitialState]], Solution).
```

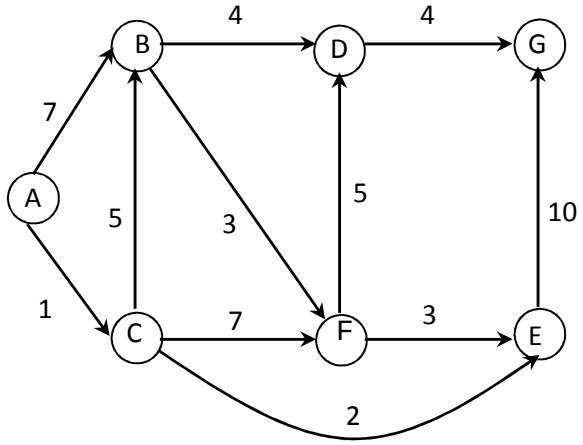
*Chú ý: khi sử dụng chương trình khung lời giải Solution sẽ trả về danh sách kết quả ngược. Để nhận danh sách đúng thứ tự cần dung vị từ Reverse (Solution, FinalResult)

Ngoài ra có thể định nghĩa vị từ solve để chương trình có cấu trúc đơn giản hơn:

solve (InitialState, Goal, FinalResult):- bfs(Goal, [[InitialState]], Solution), reverse(Solution, FinalResult).

38) Biểu diễn đồ thị có hướng như hình bên dưới trong không gian trạng thái. Gỉa sử các cung có trọng số bằng 1. Hãy tìm đường đi từ thành phố A đến G.
   1. Sử dụng giải thuật tìm kiếm theo chiều rộng
   2. Sử dụng giải thuật tìm kiếm theo chiều sâu

39) Các cung có trọng số như hình. Hãy tìm đường đi từ thành phố A đến G. Lưu lại và in ra tổng độ dài của đường đi
   1. Sử dụng giải thuật tìm kiếm theo chiều sâu
   2. Sử dụng giải thuật tìm kiếm theo chiều rộng



40) Áp dụng khung chương trình tìm kiếm theo chiều rộng hoặc sâu, giải bài toán 3 tu sĩ và 3 con quỷ.

41) Áp dụng khung chương trình tìm kiếm theo chiều rộng hoặc sâu, giải bài toán dê, sói và bắp cải.