Chương 8: Ngôn ngữ lập trình logic

Giảng viên: Ph.D Nguyễn Văn Hòa

Khoa KT-CN-MT – ĐH An Giang

Nội dung

- Giới thiệu lập trình logic
- Mệnh đề
- Ngôn ngữ Turbo ProLog

Giới thiệu lập trình logic

- Các họ ngôn ngữ lập trình bậc cao
  - Lập trình mệnh lệnh (imparative)
    - Thủ tục (procedural)
    - Hướng đối tượng (object)
  - Lập trình khai báo (declarative)
    - Hàm (functional)
    - Logic

Giới thiệu lập trình logic

- Phương thức lập trình khai báo khác với phương thức LT mệnh lệnh ở những điểm nào?
- LT logic là LT khai báo (declarative)
  - Dùng ngôn ngữ mô tả để đặc tả các vấn đề
  - Nhấn mạnh kết quả mong đợi hơn là cách thức nhận được kết quả
- Ứng dụng nhiều trong xử lý ngôn ngữ tự nhiên và Trí tuệ nhân tạo

Giới thiệu lập trình logic

- Một chương trình logic (Prolog) là tập hợp các mệnh đề
- Mỗi một mệnh đề được xây từ nhiều vị từ
- Vị từ là phát biểu về một đối tượng có thể là đúng hoặc sai

→ Chương trình Prolog = các đối tượng dữ liệu và quan hệ giữa các đối tượng dữ liệu

Giới thiệu lập trình logic

- Hạng (term) được xem là đối tượng dữ liệu
- Hạng gồm:
  - Hạng sơ cấp (elementary term) như hằng, biến
  - Hạng phức hợp (compound term) như một hàm tử (functor) có chứa các đối, có dạng:
    - Tên_hàm_tử(đối1, đối2, …)
    - VD student(an)

Giới thiệu lập trình logic

- Tam đoạn luận
  - Socrates là người — nguoi(socrates).
  - Mọi người đều phải chết — chet(X):- nguoi(X).

  ⇒ Socrates phải chết

  - Jerry là một kẻ cướp — robber(jerry).
  - Tom là con của John — childof(tom,john).
  - John thì giàu có — rich(john).
  - X là kẻ giàu có nếu như X có cha giàu có hoặc X là một kẻ cướp — rich(X):- childof(X,Y), rich(Y)

    Rich(X):- robber(X)

Mệnh đề

- Một mệnh đề có thể có một trong hai hình thức sau:
  - *Sự kiện (fact)*: khẳng định 1 thực thể có 1 hoặc vài tính chất; woman(thuy), man(an)
  - *Luật*: định nghĩa quan hệ đựa vào các quan hệ; wife(A,B):- husband(B,A)
- Chương trình prolog là tập hợp các sự kiện và luật xử lí và mô tả quan hệ giữa các đối tượng.
- Qui ước sự kiện:
  - P(A): A có tính chất P; student(an)
  - P(A,B): A là P đối với B; husband(an,thuy)
  - P(A1,A2,…, An): P là tên của tính chất; A1…An là các đối; nguyentu(atom, symbol)

Mệnh đề

- Luật:
  - Từ *nếu* được viết «:-» trong Prolog
  - Luật gồm có 2 phần
    - Phần bên trái chỉ kết luận, được gọi là đầu (head) của luật
    - Phần bên phải chỉ điều kiện, được gọi là thân của luật. Nếu có nhiều điều kiện thì chúng được cách nhau bởi dấu phẩy.
  - Sự khác nhau giữa sự kiện và luật
    - Sự kiện là khẳng định luôn luôn đúng
    - Luật do các điều kiện trong phần thân quyết định nên có thể đúng hoặc sai

```
robber(jerry).
childof(tom,john).
rich(john).
rich(X):- childof(X,Y), rich(Y)
Rich(X):- robber(X)
```

Ngôn ngữ Turbo Prolog

- Prolog: Programming in logic
- Ra đời vào năm 1973 do C.Camerauer (Đại học Marseilles, Pháp) và nhóm đồng sự phát triển
- Prolog là một ngôn ngữ cấp cao
- Có đặc điểm gần với ngôn ngữ tự nhiên
- Turbo Prolog được phát triển bởi Borland

Ngôn ngữ Turbo Prolog (tt)

- Với một số sự kiện và quy luật suy diễn đã mô tả, Prolog sẽ suy luận cho ta các kết quả
- Ví dụ

```
nguoi(socrates).
nguoi(xeda).
vua(xeda).
xanhphuc(X):- vua(X),nguoi(X).
```

```
hanhphuc(xeda)?
hanhphuc(socrates)?
hanhphuc(Y)?
```

Chương trình Turbo Prolog mẫu

```
domains
  nguoi = string
predicates
  cha(nguoi,nguoi)
  me(nguoi,nguoi)
  ong_noi(nguoi,nguoi)
  ong_ngoai(nguoi,nguoi)
clauses
  /*cac qui tac */
  ong_noi(X,Y):- cha(X,Z),cha(Z,Y).
  ong_ngoai(X,Y):- cha(X,Z),me(Z,Y).
  /* cac su kien */
  cha(nam,minh).
  cha(minh,lam).
  cha(long,giang).
  cha(long,thu).
  me(thu,phi).
```

Các yếu tố cơ bản của Turbo Prolog

- Trong một chương trình Prolog, ta cần khai báo các yếu tố sau đây: đối tượng, quan hệ giữa các đối tượng, sự kiện và các luật
- Đối tượng
  - Gồm có 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 một chuỗi ký tự được bắt đầu bằng một ký tự hoa
  - Biến không có tên và người ta dùng ký hiệu _

Các yếu tố cơ bản (tt)

- Sử dụng biến trong mệnh đề
  - Các biến là cục bộ trong mỗi mệnh đề. Nghĩa là nếu biến X xuất hiện trong 2 mệnh đề khác nhau thì sẽ tương ứng với 2 biến phân biệt
  - Biến xuất hiện trong 1 mệnh đề là biến tự do
    - Ví dụ

      have_a_child(X):- parent(X,Y).

      được đọc là: …
  - Biến chỉ xuất hiện 1 lần trong mệnh đề thì không cần đặt tên (biến vô danh)
    - have_a_child(X):- parent(X,_).

Các yếu tố cơ bản (tt)

- Quan hệ giữa các đối tượng
  - Quan hệ giữa các đối tượng được dùng dưới hình thức vị từ
  - Ví dụ: Thich(X,Y) là vị từ diễn tả câu “X thích Y” trong ngôn ngữ tự nhiên
  - Blue(car) là vị từ diễn tả câu “Car is blue”
  - Như vậy các vị từ sẽ bao gồm tên của vị từ và các đối số. Các đối số được đặt trong ngoặc và phân cách nhau bởi dấu phẩy

Cấu trúc của một CT Turbo Prolog

- Một chương trình Turbo Prolog thường gồm có 3 hoặc 4 đoạn cơ bản: clauses, predicates, domains và goal
- Phần goal có thể bỏ đi, nếu ta không thiết kế goal trong chương trình, thì khi thực hiện, hệ thống sẽ yêu cầu ta nhập goal vào

Phần domains

- Là phần định nghĩa kiểu mới dựa vào các kiểu đã biết
- Các kiểu được định nghĩa sẽ được sử dụng cho các đối số của vị từ
- Cú pháp định nghĩa kiểu
  - <DS kiểu mới> = <kiểu đã biết> hoặc
  - <DS kiểu mới> = <DS kiểu đã biết>

  Trong đó các kiểu mới phân cách bởi dấu «,» các kiểu đã biết phân cách bởi dấu «;»

Phần domains (tt)

- VD

  Domains

  ten, tac_gia, nha_xb, dia_chi = string

  nam, thang, so_luong = integer

  dien_tich = real

  nam_xb = nxb(thang, nam)

  do_vat = sach(tac_gia, ten, nha_xb, nam_xb); xe(ten, so_luong); nha(dia_chi, dien_tich)

Phần Predicates : vị từ

- Là phần bắt buộc phải có
- Phần predicates cần phải khai báo đầy đủ các vị từ sử dụng trong phần Clauses
- Cú pháp

  <Tên vị từ> (<danh sách các kiểu>)

  Các kiểu được phân cách nhau bởi «,»
- VD

  Predicates

  so_huu (ten, do_vat)

  so_nguyen_to(integer)

Phần Clauses : luật

- Là phần bắt buộc phải có; dùng để mô tả các sự kiện và các luật
- Sử dụng các vị từ đã khai báo trong phần predicates
- Cú pháp

  <Tên vị từ>(<danh sách các tham số>) <kí hiệu>

  <Tên vị từ 1>(<danh sách các tham số 1>) <kí hiệu>

  ……

  <Tên vị từ N>(<danh sách các tham số N>) <kí hiệu>

  Các ký hiệu bao gồm :- (điều kiện nếu);

  , (điều kiện và)

  ; (điều kiện hoặc)

  . (kết thúc vị từ)

Phần Clauses (tt)

- VD

  Clauses

  so_nguyen_to(2):-!.

  so_nguyen_to(N):-N>0,

  so_nguyen_to(M),

  M<N,

  N MOD M <>0.

  so_huu(“Nguyen Van A”, sach(“Do Xuan Loi”, “Cau truc DL”, “Khoa hoc Ky thuat”, nxb(8,1985))).

Phần goal

- Bao gồm các mục tiêu mà ta yêu cầu Prolog xác định và tìm kết quả
- Không bắt buộc phải có
- Nếu được viết sẵn trong CT thì đó gọi là goal nội; Nếu không, khi chạy CT Prolog sẽ yêu cầu ta nhập goal vào, goal ngoại
- VD
  - Constants
  - Pi = 3.141592653

```
predicates
  ktnt(integer,integer)
  tieptuc(integer,real,real,integer)
clauses
  ktnt(1,_):-write("Day la so nguyen to").
  ktnt(2,_):-write("Day la so nguyen to").
  ktnt(N,M):-N1=N-1,
    N2=M/N1,N3=round(M/N1),tieptuc(N1,N2,N3,M).
  tieptuc(_,N2,N3,_):-N2=N3, write("Day khong phai la so
    nguyen to").
  tieptuc(N1,N2,N3,M):-N2<>N3,ktnt(N1,M).
goal
  clearwindow,write("Nhap N:"),readint(N),ktnt(N,N).
```

Các nguyên tắc của NN Prolog

- Có 2 nguyên tắc: đồng nhất và 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

Các nguyên tắc của NN Prolog (tt)

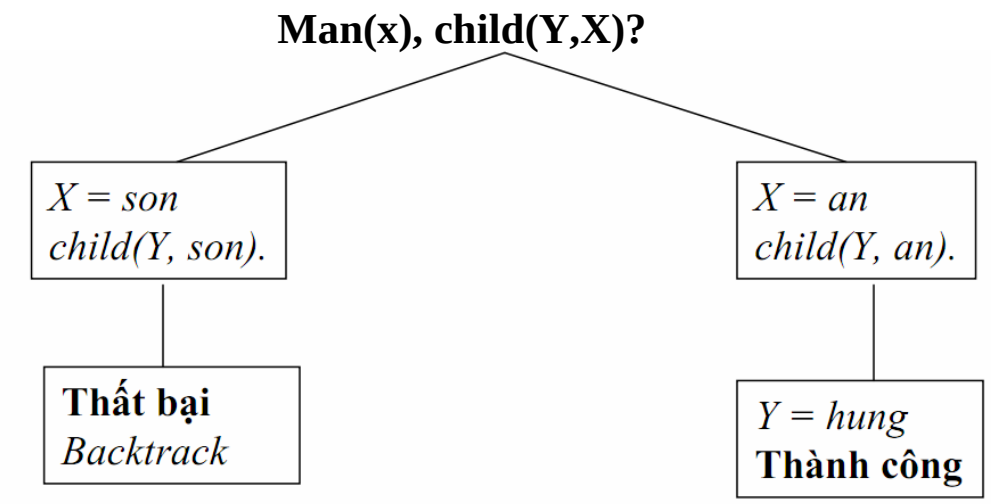
- Nguyên tắc quay lui (backtract, backtracting)
  - Cần chứng minh Goal : :- g_1, g_2, …, g_{j-1}, g_j, …, g_n
  - Kiểm chứng từ trái sang phải, đến g_i là sai, hệ thống cần phải qui lui lại g_{i-1}
  - Ví dụ

    man(son)

    man(an)

    child(hung,an)

    goal: man(X), child(Y,X)?



Cây hợp giải

- Xét các mệnh đề sau đây

sister(X,Y) :- child(X,P),child(Y,P),
woman(X), X<>Y.

woman(hien).

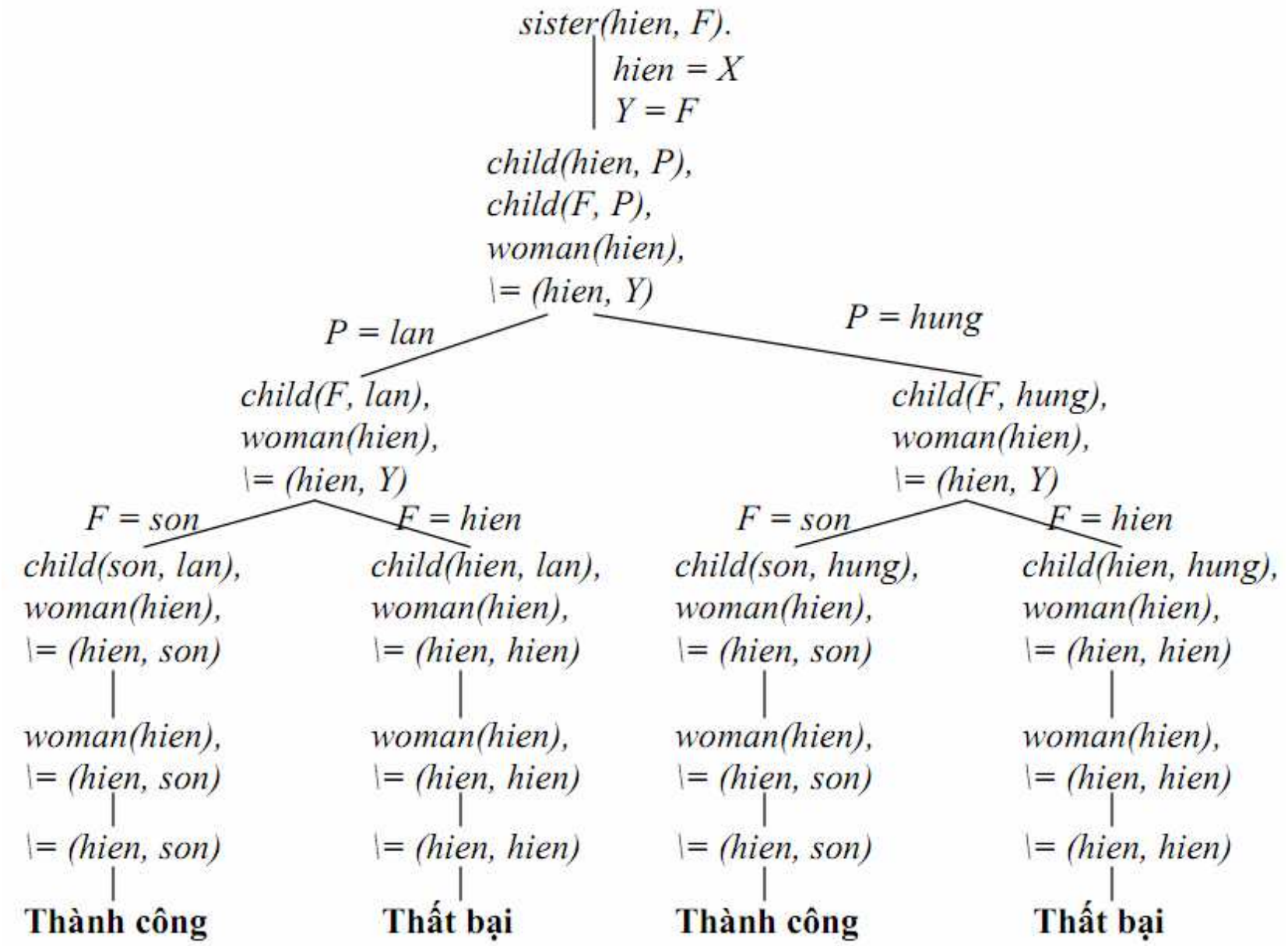
child(son,lan).

child(hien,lan).

child(son,hung).

child(hien,hung).

goal: sister(hien, F)?


sister(hien, F).
hien = X
Y = F
child(hien, P),
child(F, P),
woman(hien),
\= (hien, Y)
P = lan
P = hung
child(F, lan),
woman(hien),
\= (hien, Y)
child(F, hung),
woman(hien),
\= (hien, Y)
F = son
F = hien
F = son
F = hien
child(son, lan),
woman(hien),
\= (hien, son)
child(hien, lan),
woman(hien),
\= (hien, hien)
child(son, hung),
woman(hien),
\= (hien, son)
child(hien, hung),
woman(hien),
\= (hien, hien)
woman(hien),
\= (hien, son)
woman(hien),
\= (hien, hien)
woman(hien),
\= (hien, son)
woman(hien),
\= (hien, hien)
\= (hien, son)
\= (hien, hien)
\= (hien, son)
\= (hien, hien)
Thành công
Thất bại
Thành công
Thất bại

Cây hợp giải

- Trong ví dụ trên, chúng ta tìm thấy 2 câu trả lời giống hệt nhau được thực hiện bởi 2 con đường khác nhau (cha và mẹ).
- Để tránh điều đó, chúng ta viết lại

sister(X,Y) :- mother(M,X), father(F,X), mother(M,Y),
father(F,Y), woman(X), X<>Y.

woman(hien).

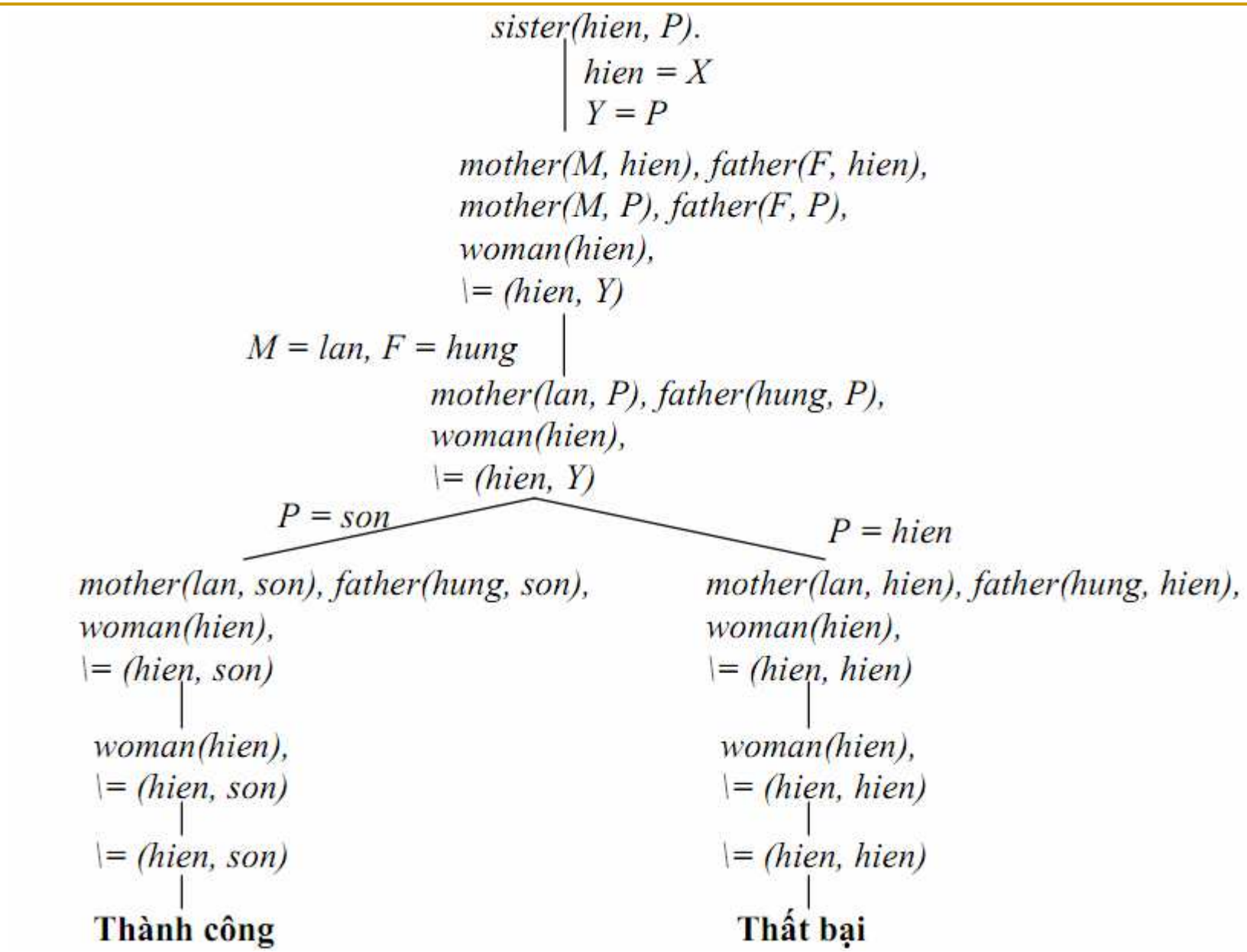
child(son,lan).

child(hien,lan).

child(son,hung).

child(hien,hung).

goal: sister(hien, F)


sister(hien, P).
hien = X
Y = P
mother(M, hien), father(F, hien),
mother(M, P), father(F, P),
woman(hien),
\= (hien, Y)
M = lan, F = hung
mother(lan, P), father(hung, P),
woman(hien),
\= (hien, Y)
P = son
P = hien
mother(lan, son), father(hung, son),
woman(hien),
\= (hien, son)
woman(hien),
\= (hien, son)
\= (hien, son)
Thành công
mother(lan, hien), father(hung, hien),
woman(hien),
\= (hien, hien)
woman(hien),
\= (hien, hien)
\= (hien, hien)
Thất bại

Bộ ký tự và từ khóa

- Prolog dùng bộ ký tự sau:
  - Các chữ cái và chữ số (A – Z, a – z, 0 – 9);
  - Các toán tử (+, -, *, /, <, =, >)
  - Các ký hiệu đặc biệt
- Một vài từ khóa
  - Trace: Khi có từ khoá này ở đầu chương trình, thì chương trình được thực hiện từng bước để theo dõi
  - Fail: Khi ta dùng goal nội, để nhận về tất cả các kết quả khi chạy goal nội, ta dùng toán tử Fail
  - ! hay còn gọi là nhát cắt, nhận chỉ một kết quả từ goal ngoại, ta dùng ký hiệu !

Kiểu dữ liệu chuẩn

- Kiểu do prolog định nghĩa sẵn: char, integer, real string và symbol
- char: ký tự, hằng phải nằm trong dấu nháy: ‘a’, ‘#’
- integer: -32768 đến 32767
- real: kiểu số thực
- string: chuỗi ký tự, hằng chuỗi ký tự nằm trong dấu nháy kép; ”prolog”

Kiểu dữ liệu chuẩn: phép toán số học

Phép toán	Ý nghĩa	Kiểu đối số	Kiểu kết quả
+	Cộng 2 số	integer, real	giống kiểu ĐS
-	Trừ 2 số	integer, real	giống kiểu ĐS
*	Nhân 2 số	integer, real	giống kiểu ĐS
/	Chia 2 số	integer, real	giống kiểu ĐS
Mod	Chia lấy phần dư	integer	integer
Div	Chia lấy phân nguyên	integer	integer

Kiểu dữ liệu chuẩn: PT quan hệ

Phép toán	Ý nghĩa	Kiểu đối số	kết quả
<	Nhỏ hơn	Char,integer,real, string	Yes or No
<=	Nhỏ hơn hay bằng	Char,integer,real, string	Yes or No
=	Bằng	Char,integer,real, string	Yes or No
>	Lớn hơn	Char,integer,real, string	Yes or No
>=	Lớn hơn hay bằng	Char,integer,real, string	Yes or No
<> hay ><	Khác nhau	Char,integer,real, string	Yes or No

Kiểu dữ liệu chuẩn: các vị tự như hàm toán học

Phép toán	Ý nghĩa	Kiểu đối số	Kiểu kết quả
Sin(X)	Tính sin của X	Real	Real
Tan(X)	Tính tang của X	Real	Real
Arctan(X)	Tính arctang của X	Real	Real
Exp(X)	Tính eX	Real	Real
Ln(X)	Tính logarit cơ số e của X	Real	Real
Log(X)	Tính Logarit cơ số 10 của X	Real	Real

Kiểu do người dùng định nghĩa

- Kiểu mẩu tin
  - Cú pháp <tên kiểu mẩu tin> = tên mẩu tin (danh sách các kiểu phần tử)

```
Domains
ten, tac_gia, nha_xb, dia_chi = string
nam, thang, so_luong = integer
dien_tich = real
nam_xb = nxb(thang, nam)
```

Kiểu do người dùng định nghĩa (tt)

- Kiểu danh sách
  - Cú pháp <tên kiểu danh sách> = <tên kiểu phần tử>*

Domains

intlist = integer*

  - Một danh sách là một dãy các phần tử phân cách nhau bởi dấu phẩy và đặt trong cặp dấu []
  - Ví dụ:

[]% Danh sách rỗng

[1,2,3] % Danh sách gồm ba số nguyên 1, 2 và 3.

[X|Y] : X là phần tử đầu và Y là danh sách đuôi

[_|Y] : phần tử đầu tiên của DS là biến tự do

Kỹ thuật đệ quy

- Sử dụng đệ quy khi một luật được định nghĩa nhờ vào chính luật đó
- Ví dụ 1
  - Chúng ta có quan hệ child(X,Y), X là con cua Y
  - Chúng ta định nghĩa quan hệ con cháu hay hậu duệ (descendant) như sau:
    - X là hậu duệ của Y nếu X là con của Y
    - X là hậu duệ của Y nếu là con của Z và Z là hậu duệ của Y.
  - Mô tả Prolog
    - descendant(X,Y):- child(X,Y).
    - descendant(X,Y):- child(X,Z), descendant(Z,Y).

Kỹ thuật đệ quy

- Ví dụ 2
  - Chúng ta có quan hệ giai thừa *facto(N,Y)*, giai thừa của N bằng Y
  - Giai thừa của 0 bằng 1: facto(0,1)
  - Giai thừa của N được tính từ giai thừa của N-1
  - Mô tả Prolog

```
Facto(0,1):- !.
Facto(N, Y) :- N>0,M = N–1, facto(M, Z), Y=N*Z.
```

- Trong kỹ thuật đệ quy, trường hợp dừng được thể hiện bằng một sự kiện

Các hàm xuất nhập chuẩn

- Xuất ra màn hình
  - write(Arg1, Arg2, … ,Argn) in ra màn hình giá trị của các đối số.
  - writef(đinh_dang, Arg1, Arg2, … ,Argn) in ra màn hình giá trị của các đối số theo định_dạng
  - Các định_dạng
    - “%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

Các hàm xuất nhập chuẩn (tt)

- Nhập vào 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

Bài tập 1

- Cho quan hệ cha mẹ (parent) được định nghĩa như sau

  parent(an, hung).

  parent(thuy,hung).

  parent(linh,an).

  parent(le,linh).

  parent(anh,thuy).

  parent(ngoc,le).

  - Hãy định nghĩa quan hệ tổ tiên (ancestor) bằng cách sử dụng quan hệ parent.
    - X là tổ tiên của Y nếu X là cha hay mẹ của Y
    - X là tổ tiên của Y nếu X là cha hay mẹ của Z và Z là cha hay mẹ của Y

Bài tập 2

Dê là động vật ăn cỏ

Chó sói là động vật hung dữ

Động vật hung dữ là động vật ăn thịt

Động vật ăn thịt thì ăn thịt

Động vật ăn cỏ thì ăn cỏ

Động vật ăn thịt thì ăn các động vật ăn cỏ

Động vật ăn thịt và động vật ăn cỏ đều uống nước

Một động vật tiêu thụ cái mà nó uống hoặc cái mà nó ăn

- Câu hỏi có động vật hung dữ không và nó tiêu thụ cái gì?
- Xác định các luật và sự kiện