



Báo cáo đồ án 2:

Biểu diễn tri thức bằng logic

Môn: Cơ sở trí tuệ nhân tạo

Sinh viên thực hiện:

Bùi Thị Cẩm Nhung - 1712645

Nguyễn Ngọc Băng Tâm - 1712747

Mục lục

1. Tổng quan	2
1.1 Mô tả đồ án	2
1.2 Đánh giá mức độ hoàn thành	2
1.3 Tổ chức chương trình	3
2. Biểu diễn tri thức	4
2.1 Hợp giải trên logic mệnh đề	4
2.1.1 Kịch bản kiểm thử	4
Test case 1	5
Test case 2	5
Test case 3	6
Test case 4	7
Test case 5	7
2.1.2 Đánh giá	8
Ưu điểm	8
Nhược điểm	8
Cải tiến	8
2.2 Biểu diễn tri thức bằng logic bậc nhất với Prolog	9
2.2.1 Sơ lược về Prolog và SWI-Prolog	9
2.2.2 Xây dựng cơ sở tri thức phá hệ Hoàng gia Anh	10
Định nghĩa và suy diễn vị từ	10
Bộ câu hỏi truy vấn	11
2.2.3 Xây dựng cơ sở tri thức phá hệ Harry Potter	12
Mô tả sơ đồ phá hệ	12
Định nghĩa và suy diễn vị từ	13
Bộ câu hỏi truy vấn	13
2.3 Suy diễn trên logic bậc nhất	15
2.3.1 Cải tiến	15
2.3.1 Mô tả chương trình	16
3. Tài liệu tham khảo	16

1. Tổng quan

1.1 Mô tả đề án

Biểu diễn tri thức (Knowledge representation) là một lĩnh vực của Trí tuệ nhân tạo nghiên cứu về các cách biểu diễn thông tin trong thế giới thực trên máy tính.

Biểu diễn tri thức thường đi đôi với lập luận, suy diễn tự động: cho phép sử dụng suy luận logic để suy diễn trên cơ sở tri thức đã có hoặc phát sinh ra các tri thức mới.

Trong đề án này, sinh viên thực hiện tìm hiểu về các cách biểu diễn tri thức khác nhau, cụ thể bằng logic mệnh đề và logic bậc nhất. Đồng thời cài đặt một số các phương pháp suy diễn trên logic đã cho.

1.2 Đánh giá mức độ hoàn thành

- Đánh giá tổng thể: 100%
- Chi tiết từng yêu cầu:

Yêu cầu	Hoàn thành	Người thực hiện
Hợp giải trên logic mệnh đề	100%	Nhung
Biểu diễn tri thức bằng logic bậc nhất với Prolog	100%	Tâm
Suy diễn tiến trên logic bậc nhất	100%	Nhung
Kịch bản kiểm thử	100%	Tâm

1.3 Tổ chức chương trình

Thư mục source code:

- Thư mục `propositional_logic`:
 - Code của chương trình hợp giải trên logic mệnh đề.
 - Thư mục test: kịch bản kiểm thử
 - `<01, 02, 03>.txt`: file đầu vào
 - `<01, 02, 03>_out.txt`: file đầu ra
 - Thư mục ref: các tài liệu tham khảo
- Thư mục `first_order_logic`:
 - Code của chương trình suy diễn tiên trên logic bậc nhất.
 - Thư mục test: kịch bản kiểm thử
 - `knowledge.pl`: cơ sở tri thức ban đầu (đã được chuẩn hóa theo quy tắc được đề cập trong mục 2.3.1).
 - `query.pl`: câu truy vấn
 - `answers.txt`: câu trả lời cho các truy vấn
 - Thư mục ref: các tài liệu tham khảo

Thư mục script:

- Thư mục `royal_family_tree`:
 - `royal_family_tree.pl`: cơ sở tri thức về gia phả hoàng gia Anh
 - `Test_royal_family_tree.txt`: câu truy vấn
 - `Output_royal_family_tree.txt`: câu trả lời cho các truy vấn
- Thư mục `harry_potter_family_tree`:
 - `harry_potter_family_tree.pl`: cơ sở tri thức về gia phả trong Harry Potter
 - `Test_harry_poter_tree.txt`: câu truy vấn
 - `Output_harry_potter_family_tree.txt`: câu trả lời cho các truy vấn

2. Biểu diễn tri thức

2.1 Hợp giải trên logic mệnh đề

Logic mệnh đề (Propositional logic) được xem là một trong những cách biểu diễn tri thức đơn giản nhất. Trong đó, đơn vị cơ bản nhất để biểu diễn tri thức là mệnh đề. Mỗi mệnh đề chỉ có thể nhận một trong hai chân trị: đúng/sai.

Cú pháp:

- Ký tự in hoa hoặc từ in hoa để biểu diễn cho một mệnh đề. Ví dụ: P, Q, $W_{1,3}$.
- Ngoặc và phép toán logic được sử dụng để nối các mệnh đề với nhau. Các phép toán logic được sử dụng bao gồm: \neg (phủ định), \wedge (hội), \vee (tuyển), \Rightarrow (kéo theo) và \Leftrightarrow (tương đương).

Hợp giải (Resolution) trên logic mệnh đề là phương pháp suy diễn ứng dụng chứng minh phản chứng: để chứng minh $KB \models \alpha$, ta chứng minh không tồn tại $(KB \wedge \neg\alpha)$.

```

function PL-RESOLUTION( $KB, \alpha$ ) returns true or false
  inputs:  $KB$ , the knowledge base, a sentence in propositional logic
            $\alpha$ , the query, a sentence in propositional logic

   $clauses \leftarrow$  the set of clauses in the CNF representation of  $KB \wedge \neg\alpha$ 
   $new \leftarrow \{ \}$ 
  loop do
    for each pair of clauses  $C_i, C_j$  in  $clauses$  do
       $resolvents \leftarrow$  PL-RESOLVE( $C_i, C_j$ )
      if  $resolvents$  contains the empty clause then return true
       $new \leftarrow new \cup resolvents$ 
    if  $new \subseteq clauses$  then return false
     $clauses \leftarrow clauses \cup new$ 

```

Figure 7.12 A simple resolution algorithm for propositional logic. The function PL-RESOLVE returns the set of all possible clauses obtained by resolving its two inputs.

2.1.1 Kịch bản kiểm thử

Nhóm xây dựng 5 kịch bản, bao gồm 3 test case với kết quả "YES" và 2 test case với kết quả "NO". Đầu vào và đầu ra của từng test case được trình bày cụ thể dưới đây.

Test case 1

Input	Output	Ghi chú
D	4	(A) hợp giải với (-A OR C)
4	C	(-D) hợp giải với (-B OR -C OR D)
A	-B OR -C	(B) hợp giải với (-B OR -C OR D)
B	-C OR D	(-A OR C) hợp giải với (-B OR -C OR D)
-A OR C	-A OR -B OR D	
-B OR -C OR D	6	
	-B	(C) hợp giải với (-B OR -C)
	-C	(B) hợp giải với (-B OR -C)
	D	(C) hợp giải với (-C OR D)
	-A OR -B	(-A OR C) hợp giải với (-B OR -C)
	-A OR D	(B) hợp giải với (-A OR -B OR D)
	-B OR D	(A) hợp giải với (-A OR -B OR D)
	2	
	{}	(B) hợp giải với (-B)
	-A	(C) hợp giải với (-A OR C)
	YES	KB entails α tồn tại mệnh đề rỗng trong KB

Test case 2

Input	Output	Ghi chú
C	7	
4	-B	(-B OR C) hợp giải với (-C)
A OR B OR -D	B OR -D	(A OR B OR -D) hợp giải với (-A)
A OR B OR C OR D	A OR B OR C	(A OR B OR -D) hợp giải với (A OR B OR C OR D)
-B OR C	A OR B OR D	(A OR B OR C OR D) hợp giải với (-C)
-A	A OR C OR D	(A OR B OR C OR D) hợp giải với (-B OR C)
	A OR C OR -D	(A OR B OR -D) hợp giải với (-B OR C)
	B OR C OR D	(A OR B OR C OR D) hợp giải với (-A)
	9	
	-D	(-B) hợp giải với (B OR -D)
	A OR B	(A OR B OR C) hợp giải với (-C)
	A OR C	(A OR B OR C) hợp giải với (-B OR C)
	A OR D	(A OR C OR D) hợp giải với (-C)
	A OR -D	(A OR B OR -D) hợp giải với (-B)
	B OR C	(A OR B OR C) hợp giải với (-A)

	B OR D C OR D C OR -D 4 A B C D 1 {} YES	(A OR B OR D) hợp giải với (-A) (A OR C OR D) hợp giải với (-A) (-B OR C) hợp giải với (B OR -D) (A OR B) hợp giải với (-A) (A OR C) hợp giải với (-A) (A OR C) hợp giải với (-C) (A OR D) hợp giải với (-A) (A) hợp giải với (-A) KB entails α tồn tại mệnh đề rỗng trong KB
--	--	--

Test case 3

Input	Output	Ghi chú
C 5 A -A OR B -B OR C OR D -D OR E -E	5 B -D -B OR D -A OR C OR D -B OR C OR E 9 -B D -A OR C -A OR D -B OR C -B OR E C OR D C OR E -A OR C OR E 5 {} -A C E -A OR E YES	(A) hợp giải với (-A OR B) (-D OR E) hợp giải với (-E) (-B OR C OR D) hợp giải với (-C) (-A OR B) hợp giải với (-B OR C OR D) (-B OR C OR D) hợp giải với (-D OR E) (-D) hợp giải với (-B OR D) (B) hợp giải với (-B OR D) (-D) hợp giải với (-A OR C OR D) (-B OR D) hợp giải với (-A OR B) (-D) hợp giải với (-B OR C OR D) (-B OR D) hợp giải với (-D OR E) (B) hợp giải với (-B OR C OR D) (B) hợp giải với (-B OR C OR E) (-A OR C OR D) hợp giải với (-D OR E) (B) hợp giải với (-B) (-A OR D) hợp giải với (-D) (-B OR C) hợp giải với (B) (-B OR E) hợp giải với (B) (-A OR C OR E) hợp giải với (-C)

Test case 4

Input	Output	Ghi chú
-C 5 A -A OR B -B OR C OR D -D OR E -E	4 B -D -A OR C OR D -B OR C OR E 5 -A OR C -B OR C C OR D C OR E -A OR C OR E 0 NO	KB KHÔNG entail α vì không phát sinh được mệnh đề mới và không tìm thấy mệnh đề rỗng.

Test case 5

Input	Output	Ghi chú
-C 4 A OR B OR -D A OR B OR C OR D -B OR C -A	5 B OR -D A OR B OR C A OR C OR D A OR C OR -D B OR C OR D 4 A OR C B OR C C OR D C OR -D 0 NO	KB KHÔNG entail α vì không phát sinh được mệnh đề mới và không tìm thấy mệnh đề rỗng.

2.1.2 Đánh giá

Ưu điểm

- Tính hoàn tất: luôn trả về kết quả.
 - Nếu mâu thuẫn xảy ra trong quá trình suy diễn, ta có thể phủ nhận mệnh đề cần phủ nhận.
 - Nếu không thể áp dụng hợp giải để phát sinh thêm bất cứ mệnh đề mới nào, ta có thể kết luận rằng không thể chứng minh mệnh đề cần phủ nhận là đúng hay sai từ cơ sở tri thức hiện có.
- Tính đúng đắn: kết quả trả về luôn là hệ quả được suy ra từ cơ sở tri thức ban đầu.

Nhược điểm

- Phát sinh tất cả các cặp mệnh đề có thể có trong cơ sở tri thức dẫn đến số lượng cặp mệnh đề cần hợp giải lớn.
- Hệ thống phát sinh các cặp mệnh đề không hợp lý khiến mệnh đề sau khi hợp giải có nhiều literal hơn mệnh đề ban đầu.
- Thiếu cơ chế định hướng: các mệnh đề được phát sinh có thể không liên quan đến mệnh đề cần phủ định.

Cải tiến

- Ưu tiên hợp giải các mệnh đề có ít literal hơn giúp tăng khả năng thu được mệnh đề sau hợp giải ngắn hơn.
- Ưu tiên hợp giải các cặp mệnh đề có liên quan đến mệnh đề cần phủ định hoặc các mệnh đề được suy diễn từ mệnh đề cần phủ định.

2.2 Biểu diễn tri thức bằng logic bậc nhất với Prolog

2.2.1 Sơ lược về Prolog và SWI-Prolog

Prolog là ngôn ngữ lập trình logic và có vai trò quan trọng trong Trí tuệ nhân tạo. Trong đồ án này, nhóm sử dụng công cụ SWI-Prolog để xây dựng cơ sở tri thức phả hệ Hoàng gia Anh và phả hệ một nhánh nhân vật trong Harry Potter.

Một số lưu ý về cú pháp Prolog:

Ký hiệu	Ký hiệu tương ứng trong Prolog
and	,
or	;
if	:-
not	not

- Tên biến trong Prolog bắt đầu bằng chữ in hoa. Tên vị từ và tên hàm bắt đầu bằng chữ thường.
- Tất cả các câu trong Prolog phải kết thúc bằng dấu chấm (.).
- Khi định nghĩa một fact trong Prolog, nếu không khai báo biến cụ thể thì mặc định chương trình hiểu biến đó mang giá trị “với mọi”.

Ví dụ:

likes(X, susie). / Mọi người đều thích susie */*

- Luật (Rule) trong Prolog có dạng:

left_hand_side :- right_hand_side .

Được hiểu là: $\text{left_hand_side} \leftarrow \text{right_hand_side}$. Trong đó biểu thức left_hand_side bắt buộc chỉ chứa 1 literal dương.

- Truy vấn (Query) trong Prolog có dạng:

?- query

Nếu chương trình tìm được đáp án cho truy vấn hoặc chứng minh được truy vấn của người dùng là đúng, SWI-Prolog sẽ hiển thị **true** hoặc các câu trả lời. Ngược lại, SWI-Prolog hiển thị **false**.

2.2.2 Xây dựng cơ sở tri thức phả hệ Hoàng gia Anh

a. Định nghĩa và suy diễn vị từ

Nhóm vị từ định nghĩa sẵn: gồm `parent(Person,Child)`, `male(Person)`, `female(Person)`, `divorced(Person, Person)` và `married(Person, Person)`.

Với nhóm vị từ này, lưu ý mối quan hệ tình trạng hôn nhân cần định nghĩa cả hai fact: `married(X, Y)` và `married(Y, X)`; `divorced(X, Y)` và `divorced(Y, X)`.

Nhóm vị từ được suy diễn:

husband(Person, Wife): Person là nam và đang kết hôn với đối tượng Wife.

wife(Person, Husband): Person là nữ và đang kết hôn với đối tượng Husband.

father(Parent, Child): Parent là nam và là phụ huynh của đối tượng Child.

mother(Parent, Child): Parent là nữ và là phụ huynh của đối tượng Child.

child(Child, Parent): Child có phụ huynh là đối tượng Parent.

son(Child, Parent): Child là nam và là con của đối tượng Parent.

daughter(Child, Parent): Child là nữ và là con của đối tượng Parent.

grandparent(GP, GC): GP là phụ huynh của phụ huynh của đối tượng GC.

grandmother(GM, GC): GP là nữ và ông/bà của đối tượng GC.

grandfather(GF, GC): GP là nam và ông/bà của đối tượng GC.

grandchild(GC, GP): GC là có ông/bà là đối tượng GP.

grandson(GS, GP): GS là nam và là cháu của đối tượng GP.

granddaughter(GD, GP): GD là nữ và là cháu của đối tượng GP.

sibling(Person1, Person2): Person1 khác Person2, cha của Person1 và Person2 giống nhau, mẹ của Person1 và Person2 cũng giống nhau.

brother(Person, Sibling): Person là nam và là anh/chị/em ruột với Sibling.

sister(Person, Sibling): Person là nữ và là anh/chị/em ruột với Sibling.

aunt(Person, NieceNephew): Person là nữ, là chị/em của ba/mẹ hoặc là vợ của chú của NieceNephew.

uncle(Person, NieceNephew): Person là nam, là anh/em của ba/mẹ hoặc là chồng của dì của NieceNephew.

niece(Person, AuntUncle): Person là nữ và có dì/chú là AuntUncle.

nephew(Person, AuntUncle): Person là nam và có dì/chú là AuntUncle.

b. Bộ câu hỏi truy vấn

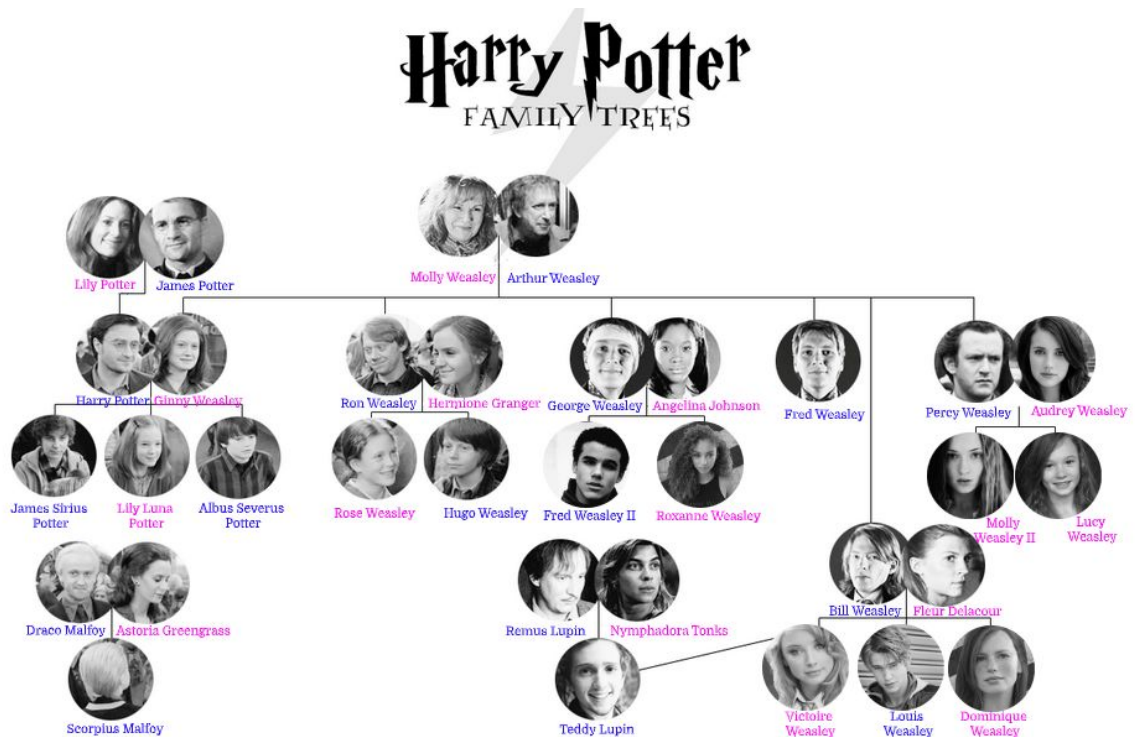
Thứ tự của bộ câu hỏi tương ứng với thứ tự câu hỏi trong file test và câu trả lời trong file output.

1. Ai là mẹ của Prince Andrew?
2. Ai là cháu gái (niece) của Princess Anne?
3. Timothy Laurence có phải chú (uncle) của James?
4. Ai là dì của Princess Charlotte?
5. Camilla Parker Bowles có phải bà (grandmother) của Archie Harrison?
6. Ai là bà của Archie Harrison?
7. Ai là cháu trai (nephew) của Meghan Markle?
8. Ai là cháu trai (grandson) của Prince Charles?
9. Ai là cháu (grandchild) của Princess Diana?
10. Ai là anh trai của Zara Tindall?
11. Ai là chú của Princess Eugenie?
12. Ai là cháu của Queen Elizabeth II?
13. Ai là cha của Autumn Phillips?
14. Mike Tindall có phải con của Mark Phillips không?

15. Ai là vợ của Prince Charles?
16. Mark Phillips có phải chồng của Princess Anne không?
17. Ai là chồng của Princess Anne?
18. Ai là anh chị em ruột của Prince Louis?
19. Ai là chị em gái của James?
20. Nữ hoàng Elizabeth có phải vợ của Mike Tindall không?
21. Ai là ông bà của Prince Louis?
22. Ai là con gái của Princess Anne?
23. Ai là con trai của Meghan Markle?
24. Ai là con trai của Sarah Ferguson?

2.2.3 Xây dựng cơ sở tri thức phả hệ Harry Potter

a. Mô tả sơ đồ phả hệ



b. Định nghĩa và suy diễn vị từ

Nhóm vị từ định nghĩa sẵn: gồm parent(Parent,Child) male(Person), female(Person), dating(Person, Person) và married(Person, Person).

Nhóm vị từ được suy diễn:

Tương tự như phả hệ Hoàng gia, nhưng có thêm các vị từ sau:

boyfriend(Person, Girlfriend): Person là nam và đang hẹn hò với đối tượng Girlfriend.

girlfriend(Person, Boyfriend): Person là nữ và đang hẹn hò với đối tượng Boyfriend.

c. Bộ câu hỏi truy vấn

Thứ tự của bộ câu hỏi tương ứng với thứ tự câu hỏi trong file test và câu trả lời trong file output.

1. Ai là con của Molly Weasley?
2. Hermione Granger có phải vợ của Harry Potter không?
3. Ai là bạn trai của Victoire Weasley?
4. Roxanne Weasley có phải bạn gái của Teddy Lupin không?
5. Ai là con trai của Audrey Weasley?
6. Ai là dì của Dominique Weasley?
7. Lily Luna Potter có những người chú nào?
8. Arthur Weasley có những người cháu (grandchild) nào?
9. James Sirius Potter là cháu trai (grandson) của ai?
10. Lucy Weasley có phải cháu gái (granddaughter) của James Potter không?
11. Ai là con gái của Fleur Delacour?
12. Ai là cha của Hugo Weasley?

13. Molly Weasley có phải là mẹ của Molly Weasley II không?
14. Ai là chồng của Nymphadora Tonks?
15. Ai là anh trai của Scorpius Malfoy?
16. Rose Weasley có phải chị/em gái của Hugo Weasley không?
17. Ai là anh/chị/em ruột của Albus Severus Potter?
18. Ai là ông của Louis Weasley?
19. Ai là bà của Fred Weasley II?
20. Kể tên ông/ bà của James Sirius Potter?
21. Ai là cháu gái (niece) của Angelina Johnson?
22. Ai là cháu trai của Fred Weasley?
23. Percy Weasley là chú của những ai?

2.3 Suy diễn trên logic bậc nhất

Xây dựng chương trình suy diễn logic bằng phương pháp **suy diễn tiến** (forward chaining). Thuật toán sử dụng tham khảo từ quyển *Artificial Intelligence: A Modern Approach*.

```

function FOL-FC-ASK( $KB, \alpha$ ) returns a substitution or false
  inputs:  $KB$ , the knowledge base, a set of first-order definite clauses
            $\alpha$ , the query, an atomic sentence
  local variables:  $new$ , the new sentences inferred on each iteration

  repeat until  $new$  is empty
     $new \leftarrow \{\}$ 
    for each  $rule$  in  $KB$  do
       $(p_1 \wedge \dots \wedge p_n \Rightarrow q) \leftarrow \text{STANDARDIZE-VARIABLES}(rule)$ 
      for each  $\theta$  such that  $\text{SUBST}(\theta, p_1 \wedge \dots \wedge p_n) = \text{SUBST}(\theta, p'_1 \wedge \dots \wedge p'_n)$ 
        for some  $p'_1, \dots, p'_n$  in  $KB$ 
           $q' \leftarrow \text{SUBST}(\theta, q)$ 
          if  $q'$  does not unify with some sentence already in  $KB$  or  $new$  then
            add  $q'$  to  $new$ 
             $\phi \leftarrow \text{UNIFY}(q', \alpha)$ 
            if  $\phi$  is not fail then return  $\phi$ 
    add  $new$  to  $KB$ 
  return false

```

Figure 9.3 A conceptually straightforward, but very inefficient, forward-chaining algorithm. On each iteration, it adds to KB all the atomic sentences that can be inferred in one step from the implication sentences and the atomic sentences already in KB . The function STANDARDIZE-VARIABLES replaces all variables in its arguments with new ones that have not been used before.

2.3.1 Cải tiến

Thuật toán Suy diễn tiến được đề cập đến ở trên tuy có thể làm việc nhưng chưa thật sự hiệu quả. Dưới đây là một số các cải tiến được áp dụng để tăng tốc độ xử lý của thuật toán:

1. Ở vòng lặp thứ t , ta chỉ bỏ qua những luật mà trong đó không có điều kiện nào có thể thỏa mãn dựa trên các tri thức mới phát sinh từ vòng lặp thứ $t - 1$.
2. Giả định rằng các luật có điều kiện chỉ được nối với nhau bằng phép toán hội, bỏ qua bước chuẩn hóa các biến trong luật (Standardize-Variable(rule)).
3. Trích ra tập hợp các câu có khả năng thỏa mãn điều kiện của luật đang xét trước khi phát sinh tổ hợp các cặp câu và tìm ra phép biến đổi Θ phù hợp.

2.3.1 Mô tả chương trình

- Đầu vào: tập tin chứa cơ sở tri thức và câu hỏi của người dùng.
- Đầu ra: câu trả lời của hệ thống (được suy diễn từ câu hỏi trên cơ sở tri thức đã có).
- Cú pháp biểu diễn vị từ tương tự như cú pháp của Prolog, tuy nhiên:
 - Các luật có điều kiện chỉ được nối với nhau bằng phép toán hội.
 - Quan hệ hai chiều cần được định nghĩa cụ thể, chẳng hạn:

`sibling(Person1, Person2) :- sibling(Person2, Person1).`

- Các câu truy vấn có cú pháp giống như vị từ thông thường, không thêm dấu nháy đơn. Chẳng hạn muốn thực hiện truy vấn "Ai là mẹ của Hoàng tử Charles?", ta sử dụng cú pháp như sau:

`mother(X, prince_charles).`

3. Tài liệu tham khảo

- [1] [Trincoll Computer Science - Prolog](#)
- [2] [Prolog operator](#)
- [3] Giáo trình Artificial Intelligence - A Modern Approach 3rd edition