

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA: CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN 2

Biểu diễn tri thức bằng Logic

LỚP : CƠ SỞ TRÍ TUỆ NHÂN TẠO - CQ2017/21

GVLT : Lê Hoài Bắc

GVHDTH: Nguyễn Ngọc Thảo - Lê Ngọc Thành - Nguyễn Ngọc Đức

SINH VIÊN THỰC HIỆN:

1712787 - Nguyễn Văn Thìn

1712842 - Huỳnh Lương Phương Trúc

1712920 - Nguyễn Minh Vũ

Tp. Hồ Chí Minh, ngày 2 tháng 12 năm 2019

MỤC LỤC

1	4
Tổng quan & Milestone	4
Thông tin nhóm.....	4
Milestone	4
Mức độ hoàn thành bài tập	5
2	6
Nội dung 1 – Hợp giải trên logic mệnh đề	6
2.1 Nội dung thuật toán hợp giải	6
2.2 Mô tả cấu trúc lưu trữ.....	7
2.3 Cách chạy file script.....	7
Command line arguments:	7
Ví dụ:	8
2.4 Testcases	9
Testcase 1	9
Testcase 2	9
Testcase 3	9
Testcase 4	10
Testcase 5	11
2.5 Đánh giá giải thuật hợp giải trên logic mệnh đề và đề xuất giải pháp.....	12
a) Đánh giá:	12
b) Giải pháp:	12
3	13
Nội dung 2 – Biểu diễn tri thức bằng logic bậc nhất với Prolog	13
3.1 - Bài tập cây phả hệ cho Hoàng gia Anh	13
Bộ câu hỏi truy vấn	13
3.2 - Bài tập tự chọn một chủ đề để xây dựng cơ sở tri thức	15

Tóm tắt cơ sở tri thức về chủ đề nhà hàng:.....	15
Các quan hệ cơ bản nhất trong chủ đề:.....	16
Sơ đồ các đối tượng và quan hệ cơ bản:	16
Các quan hệ nâng cao:	18
Bộ câu hỏi test cho cơ sở tri thức nhà hàng:	19
4	21
References	21

1

Tổng quan & Milestone

Thông tin nhóm

MSSV	Họ tên	Email
1712787	Nguyễn Văn Thìn	vanthin7111999@gmail.com
1712842	Huỳnh Lương Phương Trúc	huynhtruc0309@gmail.com
1712920	Nguyễn Minh Vũ	yumnguyenn@gmail.com

Milestone

Cột mốc	Công việc dự kiến	Ước lượng (hour)	Sản phẩm
21/11/2019 Tìm hiểu yêu cầu đề bài	Phân tích yêu cầu đề bài	1	Đọc, hiểu được yêu cầu nội dung
	Lên kế hoạch thực hiện	0.5	Kế hoạch thực hiện
	Phân chia công việc	0.5	Lịch phân chia công việc
24/11/2019 Nội dung 2	Tìm hiểu Prolog	2	Prolog
	Tìm hiểu suy luận bậc nhất trên Prolog	2	
	Tìm hiểu công cụ SWI-Prolog	1	SWI-Prolog & cách thức triển khai & ví dụ minh họa
	Thực hiện nội dung xây dựng cơ sở tri thức cho cây phả hệ	3	royal_family_tree.pl
	Đưa ra 20 bộ câu hỏi để hỏi hệ tri thức cho cây phả hệ	3	royal_family_tree_test.pl
	Xây dựng cơ sở tri thức theo chủ đề tự chọn	3	restaurant.pl

	Đưa ra 20 bộ câu hỏi để hỏi hệ tri thức theo chủ đề tự chọn	3	restaurant_test.pl
27/11/2019 Nội dung 1	Tìm hiểu hợp giải trên logic mệnh đề	2	Hợp giải PL-Resolution
	Lập trình phần Input và Output file	2	<code>def read_input(path):</code> <code>def write_output(file,</code> <code>buffer):</code>
	Lập trình phần giải thuật hợp giải	10	plresolve.py
	Chuẩn bị vài bộ kiểm thử	2	5 bộ testcases
	Kiểm thử, sửa lỗi	5	Hoàn thiện nội dung 1
24/10/2019 Hoàn tất báo cáo	Hoàn tất báo cáo	2	Báo cáo đồ án 2
25/10/2019 Nộp đồ án	Nộp đồ án về Moodle	0.5	Nộp trên Moodle

Mức độ hoàn thành bài tập

Nội dung	Mô tả	Độ hoàn thành
Hợp giải trên logic mệnh đề	Đọc dữ liệu đầu vào và lưu trong cấu trúc dữ liệu phù hợp (5%)	100%
	Cài đặt giải thuật hợp giải trên logic mệnh đề (10%)	
	Các bước suy diễn phát sinh đủ mệnh đề và kết luận đúng (30%)	
	Tuân thủ mô tả định dạng của đề bài (5%)	
	Báo cáo test case và đánh giá (10%)	
Biểu diễn tri thức bằng logic bậc nhất với Prolog	Xây dựng cây phả hệ cho gia đình Hoàng gia Anh và đưa ra bộ 20 câu hỏi (20%)	100%
	Xây dựng cơ sở tri thức theo chủ đề tự chọn và đưa ra bộ 20 câu hỏi (20%)	

2

Nội dung 1 – Hợp giải trên logic mệnh đề

2.1

Nội dung thuật toán hợp giải

Thủ tục suy diễn dựa trên hợp giải hoạt động dựa trên nguyên tắc chứng minh phản chứng.

Nghĩa là, để chứng minh cơ sở tri thức $KB \models \alpha$ (KB bao hàm α), ta chứng minh phủ định của nó ($KB \wedge \neg\alpha$) được biến đổi về dạng CNF. Sau đó, luật hợp giải được áp dụng lên các mệnh đề kết quả. Mỗi cặp mệnh đề bù nhau được hợp giải để tạo ra mệnh đề mới và được thêm vào cơ sở tri thức nếu nó chưa xuất hiện. Quá trình tiếp tục cho đến khi một trong 2 điều này xảy ra.

- Không có mệnh đề mới nào có thể được thêm vào, trong trường hợp đó KB không suy dẫn được α
- Hai mệnh đề hợp giải nhận được một mệnh đề rỗng, KB suy dẫn được α

```
function HợpGiải( $KB, \alpha$ ) return true hay false
input:  $KB$ , cơ sở tri thức (tập câu logic mệnh đề)
        $\alpha$ , câu truy vấn (câu logic mệnh đề)
MệnhĐề  $\leftarrow$  tập các câu theo biểu diễn CNF của  $KB \wedge \neg\alpha$ 
MđMới  $\leftarrow \{\}$ 
loop
  for each  $C_i, C_j$  in tập MệnhĐề
    KếtQuảHợpGiải  $\leftarrow$  HợpGiảiCâu( $C_i, C_j$ )
    nếu KếtQuảHợpGiải chứa mệnh đề rỗng thì trả về true
    MđMới  $\leftarrow$  MđMới  $\cup$  KếtQuảHợpGiải
  nếu MđMới  $\subseteq$  MệnhĐề thì trả về false
  MệnhĐề  $\leftarrow$  MệnhĐề  $\cup$  MđMới
```

Thuật toán hợp giải

2.2

Mô tả cấu trúc lưu trữ

- Các câu logic mệnh đề được lưu thành một list bao gồm các tuple, mỗi tuple là một câu logic mệnh đề (câu α cũng được đọc thành một list có 1 phần tử là 1 tuple). Trong đó:
 - + dấu phẩy “,” của mỗi tuple tương đương với phép OR
 - + dấu phẩy “,” của list tương đương với phép AND
- Ví dụ:

input.txt	Được lưu thành
-A	alpha = [(-A',)]
4	
-A OR B B OR -C A OR -B OR C -B	KB_sentences = [(-A', 'B'), ('B', '-C'), ('A', '-B', 'C'), ('-B',)]

- Để thuận lợi cho việc lưu trữ kết quả hợp giải, ta sử dụng class KB. Trong đó:
 - + **KB.sentences (kiểu list)**: được gán bằng KB_sentences được đọc ở trên, không thay đổi trong suốt quá trình giải quyết bài toán
 - + **KB.clauses (kiểu list)**: được khởi tạo bởi KB_sentences hợp với phủ định của câu α , sau mỗi vòng lặp sẽ được hợp với các kết quả mới mà chưa tồn tại trong nó
 - + **KB.results (kiểu string)**: dùng để lưu kết quả hợp giải theo đúng trình tự thực hiện
 - + **KB.PL_Resolution(alpha, flag)**: hàm hợp giải theo mã giả đã nêu ở trên. Các tham số: alpha – câu α cần chứng minh; flag – nếu là True, KB.PL_Resolution() sẽ dừng ngay khi phát hiện mệnh đề rỗng (“{}”), nếu là False, KB.PL_Resolution() vẫn tiếp tục hợp giải cho đến khi hết cặp câu mới dừng

2.3

Cách chạy file script

Command line arguments:

```
D:\Study\3rdYear\1stTerm\AI Basis\Lab02>python plresolve.py -h
usage: plresolve.py [-h] [-es {True,False,T,F}] input output

Propositional Logic Resolution

positional arguments:
  input          The input file
  output         The output file

optional arguments:
  -h, --help            show this help message and exit
  -es {True,False,T,F}, --earlystack {True,False,T,F}
                        Stop at the first occurrence of {} if True/Show all the
                        remaining resolutions if False
```

-es/--earlystack là tham số tùy chọn:

+ True/T: dừng ngay khi phát hiện mệnh đề rỗng “{}”

+ False/F: khi phát hiện mệnh đề rỗng thì chưa dừng ngay mà tiếp tục hợp giải cho đến khi hết cặp câu

Ví dụ:

```
D:\Study\3rdYear\1stTerm\AI Basis\Lab02>python plresolve.py input.txt output.txt -es True
3
-A
B
-C
4
-B OR C
A OR C
A OR -B
{}
YES
```

```
D:\Study\3rdYear\1stTerm\AI Basis\Lab02>python plresolve.py input.txt output.txt -es False
3
-A
B
-C
5
-B OR C
A OR C
A OR -B
{}
{}
YES
```


2.4

Testcases

Các test case dưới đây được chạy trong trường hợp earllystop == False

Testcase 1

Input1.txt	Output1.txt
S	2
2	P OR S
P OR Q	-Q
-Q OR S	1
	P
	0
	NO

Testcase 2

Input2.txt	Output2.txt
S	4
4	R
P	-R OR S
Q	-P OR -Q OR S
-P OR R	-Q OR -R
-Q OR -R OR S	6
	-Q OR S
	-P OR S
	-R
	-P OR -Q
	S
	-Q
	4
	-P
	{}
	{}
	{}
	YES

Testcase 3

Input3.txt	Output3.txt
R	4
3	Q OR R
P OR Q	P OR R
-P OR R	-P
-Q OR R	-Q
	3
	Q
	P
	R
	3
	{}
	{}
	{}
	YES

Testcase 4

Input4.txt	Output4.txt
-A OR -D	8
4	-A OR B OR E
-A OR B OR D	-A OR B OR F
-D OR E	-A OR D OR -E
-D OR F	B OR D
-E OR -A OR -B	-A OR -B OR -D
	E
	F
	-B OR -E
	10
	B OR E
	-B OR -D
	-A OR -E OR F
	B OR F
	-A OR -B
	D OR -E
	-A OR -D OR E
	-A OR -D OR F
	-A OR D

	-B
	3
	-A OR E
	-E OR F
	-A OR F
	0
	NO

Testcase 5

Input5.txt	Output5.txt
-A OR -B OR -D	9
4	-A OR B OR E
-A OR B OR D	-A OR B OR F
-D OR E	-A OR D OR -E
-D OR F	B OR D
-E OR -A OR -B	-A OR -B OR -D
	E
	F
	-B OR -E
	-A OR -E
	14
	B OR E
	-B OR -D
	-A OR -D
	-A OR -E OR F
	B OR F
	-A OR -B
	D OR -E
	-E
	-A OR -D OR E
	-A OR B
	-A OR -D OR F
	-A OR D
	-B
	-A
	7
	-D
	-A OR E

	-E OR F -A OR F {} {} {} YES
--	---

2.5

Đánh giá giải thuật hợp giải trên logic mệnh đề và đề xuất giải pháp

a) Đánh giá:

Ưu điểm	Khuyết điểm
Tính tổng quát: thuật toán này có thể suy dẫn mọi câu đúng dẫn từ cơ sở tri thức cho trước	Không giống với cách tư duy của con người
Tính phổ dụng: mọi bài toán logic khi đưa về dạng hội chuẩn CNF đều có thể được hợp giải	Khi chuyển các câu về dạng mệnh đề, ta đã đánh mất giá trị về kinh nghiệm chứa trong câu ban đầu
	Về kĩ thuật, thuật giải này có độ phức tạp thời gian là $\Theta(2^n)$, với n là độ dài của mảng chứa các mệnh đề gốc và mệnh đề phái sinh ($\text{len}(\text{KB.clauses})$)

b) Giải pháp:

- Sử dụng logic vị từ
- Về kĩ thuật, có thể lưu trữ kết quả suy diễn giữa các cặp mệnh đề vào ma trận $n \times n$ để tham chiếu nhanh mà không cần phải hợp giải lại trong vòng lặp mới. Điều này làm tăng độ phức tạp về không gian, tuy nhiên giảm đáng kể độ phức tạp về thời gian.

3

Nội dung 2 – Biểu diễn tri thức bằng logic bậc nhất với Prolog

3.1 - Bài tập cây phả hệ cho Hoàng gia Anh

Code Prolog được đính kèm vào file nộp.

Bộ câu hỏi truy vấn

1. Liệt kê những người có giới tính là nam?

```
?-  
|   male(X).  
X = phillip ;  
X = charles ;  
X = harry_duke ;  
X = william ;  
X = george ;  
X = louis ;  
X = harrison ;  
X = timothy_laurence ;  
X = mark_phillips ;  
X = peter_phillips ;  
X = mike ;  
X = andrew ;  
X = edward ;  
X = james.
```

2. Harry có phải là con của Charles không?

```
?- child(harry_duke, charles).  
true.
```

3. Cha của George là ai?

```
?- father(X, george).  
X = william ;  
false.
```

4. Diana và Charles có phải là vợ chồng không?

```
?- married(diana, charles).  
false.
```

5. Chồng của Anne là ai?

```
?- husband(X, anne).  
X = timothy_laurence.
```

6. Liệt kê các con trai của Elizabeth?

```
?- findall(X, son(X, elizabeth), Son).
Son = [charles, andrew, edward].
```

7. Con của William là ai?

```
?- findall(X, child(X, william), Children).
Children = [george, charlotte, louis].
```

8. Các con gái của Sarah?

```
?- findall(X, daughter(X, sarah), Daughter).
Daughter = [beatrice, eugene].
```

9. Ai là ông của James?

```
?- grandfather(X, james).
X = phillip .
```

10. Anh em ruột của Andrew?

```
?- findall(X, sibling(andrew, X), Siblings).
Siblings = [charles, anne, edward].
```

11. Meghan là dì của ai?

```
?- findall(X, aunt(meghan, X), NieceNephew).
NieceNephew = [george, charlotte, louis].
```

12. Anh trai của Zara Phillips là ai?

```
?- findall(X, brother(X, zara), Brother).
Brother = [peter_phillips].
```

13. Tìm tất cả cháu (quan hệ ông/bà cháu) của nữ hoàng?

```
?- grandchild(X, 'elizabeth').
X = william ;
X = harry_duke ;
X = peter_phillips ;
X = zara ;
X = beatrice ;
X = eugene ;
X = james ;
X = louise_windsor.
```

14. Anne và Harry Duke có phải quan hệ dì cháu không?

```
?- aunt(anne, harry_duke).
true .
```

15. Ai vừa là cậu của William và Peter Phillip?

```
?- uncle(X, william), uncle(X, peter_phillips).
X = andrew ;
X = edward ;
false.
```

16. Cháu gái của Phillip ma còn độc thân?

```
?- findall(X, (granddaughter(X, phillip), not(married(X, Y))), FA_granddaughter).
FA_granddaughter = [beatrice, eugene, louise_windsor].
```

17. Harrison và George có cùng ông hay không?

```
?- grandfather(X, harrison), grandfather(X, george).
X = charles ;
false.
```

18. Cháu trai của Harry_Duke là ai?

```
?- nephew(X, harry_duke).
X = george ;
X = louis ;
false.
```

19. Ai là chị/em gái của George?

```
?- findall(X, sister(X, 'george'), Sister).
Sister = [charlotte].
```

20. James có phải là con của Anne?

```
?- child(james, Anne).
Anne = sophie ;
Anne = edward.
```

3.2 - Bài tập tự chọn một chủ đề để xây dựng cơ sở tri thức

- Tóm tắt cơ sở tri thức về chủ đề nhà hàng:

Các đối tượng thuộc các tập cơ bản nhất là:

Person: người. Mỗi người có thể là một đầu bếp (*Chef*), người ăn chay (*Vegetarian*) hoặc

người bình thường

Ingredient: nguyên liệu. Nguyên liệu có thể có thuộc kiểu nguyên liệu ngọt (*Sweet*), mặn

(*Salty*), chua (*Sour*), cay (*Spicy*), là thịt (*Meat*), là hải sản (*Seafood*), rau (*Vegetables*), trái

cây (*Fruit*) hoặc nguyên liệu bình thường.

Place: nguồn gốc nguyên liệu.

Drink: đồ uống. Đồ uống có thể thuộc loại bình thường hoặc có cồn (*Alcoholic*).

Dish: món ăn.


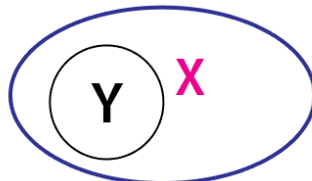
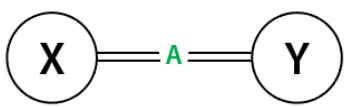
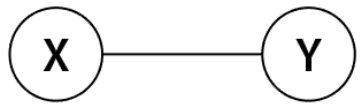
Event: sự kiện/ loại buổi tiệc.

Các quan hệ cơ bản nhất trong chủ đề:

source(I, S): nguyên liệu I có nguồn gốc từ nơi S.
 dish_in_event(D, E): món D được phục vụ trong buổi tiệc E.
 ingredient_in_dish(I, D): nguyên liệu I được dùng trong món D.
 drink_in_event(D, E): đồ uống D được phục vụ trong buổi tiệc E.
 allergy(P, I): người P bị dị ứng với nguyên liệu I.
 chef_in_event(C, E): đầu bếp C phụ trách buổi tiệc E.
 invited(E, P): người P được mời tới buổi tiệc E.
 cant_drink(P, D): người P không uống được loại đồ uống D.

Sơ đồ các đối tượng và quan hệ cơ bản:

Chú thích:

STT	Ký hiệu	Ý nghĩa
1		Đối tượng X.
2		Đối tượng Y trong tập hợp X.
3		Các phần tử trong tập X và Y nếu có quan hệ với nhau thì đó là quan hệ loại A.
4		Đối tượng X có quan hệ với đối tượng Y. Loại quan hệ được xác định bằng quan hệ của 2 tập hợp như chú thích [3]

Xem sơ đồ trong link sau:

<https://drive.google.com/file/d/1HPjsScni66--KDc8r9tCp-1Oa47rvmPl/view?usp=sharing>

Lưu ý:

- Mở bằng draw.io Diagrams
- Ký hiệu thứ 4 không được vẽ lên sơ đồ vì quá nhiều làm cho hình bị rối

Các quan hệ nâng cao:

- `ingredient_in_event(I, E)`: nguyên liệu I được dùng trong sự kiện E, khi có món D làm từ nguyên liệu I được phục vụ trong sự kiện E.
- `cant_eat(P, D)`: người P không ăn được món D, khi có nguyên liệu I trong món D khiến người P bị dị ứng; hoặc người P ăn chay và món D có nguyên liệu làm từ thịt hoặc cá.
- `cant_join(P, E)`: người P không thể tham gia sự kiện E, khi có món D trong sự kiện E mà người P không ăn được.
- `not_fun_event(E)`: bữa tiệc E không vui, nếu có một người P được mời nhưng không thể tham gia hoặc không uống được đồ uống có cồn D (nếu có) trong bữa tiệc.
- `same_ingredient_events(I, E1, E2)`: nguyên liệu I có được dùng trong cả 2 sự kiện E1 và E2 không.
- `same_ingredients_event(I1, I2, E)`: nguyên liệu I1 và I2 có cùng được dùng trong sự kiện E hay không.
- `same_ingredient_dishes(I, D1, D2)`: nguyên liệu I có được dùng chung trong 2 món D1 và D2 không.
- `same_ingredients_dish(I1, I2, D)`: nguyên liệu I1 và I2 có được dùng chung trong món D hay không.
- `same_dish_events(D, E1, E2)`: món D có được phục vụ trong cả 2 sự kiện E1 và E2 hay không.
- `same_dishes_event(D1, D2, E)`: món D1 và D2 có cùng được phục vụ trong sự kiện E hay không.
- `high_grade_ingredient(I)`: nguyên liệu I là cao cấp, nếu nó có nguồn gốc từ Nhật Bản.
- `high_grade_dish(D)`: món D là cao cấp, nếu nó có sử dụng 1 nguyên liệu cao cấp.
- `high_grade_event(E)`: sự kiện E là cao cấp, nếu nó phục vụ 1 món cao cấp.
- `same_chef_events(C, E1, E2)`: đầu bếp C có phụ trách cả 2 sự kiện E1 và E2 không.
- `same_chefs_event(C1, C2, E)`: 2 đầu bếp C1 và C2 có cùng phụ trách sự kiện E không.
- `same_source_ingredients(S, I1, I2)`: 2 nguyên liệu I1 và I2 có chung nguồn gốc là S không.
- `source_in_dish(S, D)`: món D có dùng nguyên liệu nào có nguồn gốc từ S không.
- `same_source_dishes(S, D1, D2)`: món D1 và D2 có dùng nguyên liệu cùng đến từ S không.
- `same_sources_dish(S1, S2, D)`: món D có dùng nguyên liệu đến từ cả S1 và S2 không.

- source_in_event(S, E): sự kiện E có dùng nguyên liệu đến từ S không.
- same_source_events(S, E1, E2): 2 sự kiện E1 và E2 có cùng dùng nguyên liệu đến từ S1 và S2 không.
- same_sources_event(S1, S2, E): sự kiện E có dùng nguyên liệu đến từ S1 và S2 không.

Bộ câu hỏi test cho cơ sở tri thức nhà hàng:

1. vu_1 có phải là đầu bếp?

```
?-
|   chef(vu_1).
false.
```

2. Đầu bếp có những ai?

```
?- findall(X, chef(X), Chef).
Chef = [thin_1, thin_2, truc_1].
```

3. Những nguyên liệu nào đến từ Trung Quốc?

```
?- findall(X, source(X, china), FromChina).
FromChina = [egg, potato, onion, chili, garlic, ginger, peanut].
```

4. truc_1 có bị dị ứng với “Hành” hay không?

```
?- allergy(truc_1, garlic).
false.
```

5. Ai là người ăn chay?

```
?- findall(X, vegetarian(X), Vegetarian).
Vegetarian = [truc_1, truc_2].
```

6. Làm món trứng chiên thì cần những nguyên liệu gì?

```
?- findall(X, ingredient_in_dish(X, fried_egg), I).
I = [egg, salt, pepper, butter, fish_sauce, tomato].
```

7. Cá chiên có cần để thịt hay không?

```
?- ingredient_in_dish(meat(X), fried_fish).
false.
```

8. Có được uống rượu trong sinh nhật không?

```
?- drink_in_event(wine, birthday).
false.
```

9. thin_1 là đầu bếp trong những sự kiện nào?

```
?- findall(E, chef_in_event(thin_1, E), Event).
Event = [family_meeting, wedding].
```

10. truc_1 không thể ăn được những món nào?

```
?- findall(X, cant_eat(truc_1, X), FoodTruc).
FoodTruc = [poached_pork_and_egg, beefsteak, salad, salad, poached_fish, fried_fish, salad].
```

11. Tiệc đám cưới có vui không?

```
?- not_fun_event(wedding).
true.
```

12. truc_2 có đến được buổi tiệc sinh nhật không?

```
?- cant_join(truc_2, birthday).
true.
```

13. Người ăn chay thì không ăn được những thứ gì?

```
?- findall(X, (vegetarian(P), cant_eat(P, X)), V).
V = [poached_pork_and_egg, beefsteak, salad, salad, poached_fish, fried_fish, salad, poached_pork_and_egg, beefsteak|...].
```

14. Liệt kê những món cao cấp (high_grade_dish)?

```
?- findall(D, high_grade_dish(D), Dish).
Dish = [beefsteak, salad].
```

15. Liệt kê những nguyên liệu cao cấp, có nguồn gốc từ Nhật Bản?

```
?- findall(D, high_grade_ingredient(D), FoodJapan).
FoodJapan = [beef].
```

16. Đám cưới thì cần những nguyên liệu từ nguồn nào?

```
?- ingredient_in_event(I, wedding).
I = fish ;
I = olive_oil ;
I = garlic ;
I = fish_sauce ;
I = vinegar ;
I = chili ;
I = pepper ;
I = beef ;
I = shrimp ;
I = chicken ;
I = tomato ;
I = peanut ;
I = vinegar ;
I = beef ;
I = olive_oil ;
I = garlic ;
I = potato.
```

17. thin_2 có uống được rượu không?

```
?- cant_drink(thin_2, wine).
false.
```

18. `thin_1` và `thin_2` có làm đầu bếp trong cùng tiệc đám cưới không?

```
?- same_chefs_event(thin_1, thin_2, wedding).  
false.
```

19. Những món ăn nào được sử dụng trong tiệc gia đình và đến từ Trung Quốc?

```
?- findall(D, (dish_in_event(D, family_meeting), source_in_dish(china, D)), Dish).  
Dish = [fried_egg, fried_fish, fried_fish, poached_fish, poached_pork_and_egg, poached_pork_and_egg, poached_pork_and_egg, salad].
```

20. Đầu bếp `thin_1` có phụ trách cả 2 sự kiện `wedding` và `birthday` không?

```
?- same_chef_events(thin_1, wedding, birthday).  
false.
```

4

References

- Sách Cơ sở trí tuệ nhân tạo – Lê Hoài Bắc – Tô Hoài Việt
- Source code Artificial Intelligence: A Modern Approach <https://github.com/aimacode/aima-python>