



**ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN**

BIỂU DIỄN TRI THỨC BẰNG LOGIC

CƠ SỞ TRÍ TUỆ NHÂN TẠO

• Nhóm thực hiện:

1712702 – Nguyễn Hà Quang

1712227 – Lâm Thành Lộc

1712441 – Trần Đình Tôn Hiếu

MỤC LỤC

PHẦN 0: THÔNG TIN NHÓM	3
PHẦN 1: CÂU 1.....	4
1. Môi trường lập trình.....	4
2. Cấu trúc dữ liệu.....	4
a. Class Word	4
b. Class Sentence	5
3. Chi tiết cách cài đặt.....	6
a. Đọc dữ liệu đầu vào và lưu trong cấu trúc dữ liệu	6
b. Hàm PL_Resolution thực thi thuật toán hợp giải và ghi dữ liệu vào file output	7
4. Chạy test	8
PHẦN 2: CÂU 2.....	10
1. Cấu trúc đồ án	10
2. Chi tiết cài đặt chương trình Prolog đối với Cây phả hệ Hoàng gia Anh	10
3. Chạy thử một số testcase đối với Cây phả hệ Hoàng gia Anh	11
4. Cơ sở trí thức tự chọn: Tổ chức lớp học trong trường đại học	12
5. Chạy thử một số testcase đối với Tổ chức lớp học.....	13
PHẦN 3: CÂU 3.....	15
1. Cây phả hệ Hoàng gia Anh	15
2. Cơ sở trí thức tự chọn: Tổ chức lớp học trong trường đại học	17
3. Đánh giá.....	18
THAM KHẢO	19

PHẦN 0: THÔNG TIN NHÓM

1. Phân công công việc

Thành viên	MSSV	Nhiệm vụ
Trần Đình Tôn Hiếu	1712441	Câu 2
Nguyễn Hà Quang	1712702	Câu 1, viết báo cáo
Lâm Thành Lộc	1712227	Câu 3

2. Mức độ hoàn thành mỗi yêu cầu

Câu	Tỉ trọng	Công việc	Hoàn thành
1	60/130	Hoàn thành đầy đủ các yêu cầu	100%
2	40/130	Hoàn thành đầy đủ các yêu cầu	100%
3	30/130	Hoàn thành đầy đủ các yêu cầu	100%

PHẦN 1: CÂU 1

1. Môi trường lập trình

Ngôn ngữ lập trình sử dụng: Python 3

Hệ điều hành: Linux (chạy được trên Windows, MAC OS)

2. Cấu trúc dữ liệu

a. Class Word:

Lưu các từ (Literal) của câu ở dạng chuẩn CNF. Dữ liệu của mỗi đối tượng Word gồm có:

- + Character (string): lưu kí tự
- + Positive (boolean): lưu dấu của từ

Ví dụ:

- + Với từ A, đối tượng Word sẽ có dữ liệu là ('A', True)
- + Với từ -B, đối tượng Word sẽ có dữ liệu là ('B', False)

Các phương thức quan trọng của class Word:

- + __init__(self, string): khởi tạo đối tượng từ chuỗi
- + to_string(self): chuyển đối tượng thành chuỗi
- + get_negat(self): lấy phủ định của từ

```
class Word:
    def __init__(self, string):
        if len(string) == 1:
            self.character = string
            self.positive = True
        else:
            self.character = string[1]
            self.positive = False

    def to_string(self):
        if self.positive:
            return self.character
        else:
            return "-" + self.character

    def get_negat(self):
        if self.positive:
            nega = Word("-" + self.character)
        else:
            nega = Word(self.character)
        return nega
```

b. Class Sentence:

- Lưu câu mệnh đề dạng chuẩn CNF, mỗi câu là tập hợp của các từ (phép nối OR không cần quan tâm), vì thế dữ liệu của mỗi đối tượng Sentence là một mảng content chứa các Word
- Để thêm từ mới vào câu, sử dụng hàm **add_word(self, word)**, hàm sẽ kiểm tra từ mới đã có trong câu hay chưa. Nếu đã có thì không thêm vào, nếu chưa thì thêm vào và sắp xếp lại câu để đảm bảo tính thứ tự.
- Một số phương thức quan trọng khác của class Sentence:
 - + to_string(self): chuyển đối tượng thành chuỗi
 - + get_negate_to_sentence(self): phủ định một câu và trả về mảng các câu
 - + resolution(self, other): hợp giải hai câu, trả về mảng các câu kết quả

```
class Sentence:
    def __init__(self):
        self.content = []

    def add_word(self, word):
        if not self.is_exist(word):
            self.content.append(word)
            self.content.sort()

    def resolution(self, other):
        sentences = []
        for i in self.content:
            nega = i.get_negate()
            if other.is_exist(nega):
                sen = Sentence()
                for j in self.content:
                    if j == i:
                        continue
                    sen.add_word(j)
                for j in other.content:
                    if j == nega:
                        continue
                    sen.add_word(j)

                if sen.is_make_sense():
                    sentences.append(sen)
        return sentences

    def to_string(self):
        if len(self.content) == 0:
            return "{}"
```

3. Chi tiết cách cài đặt

a. Đọc dữ liệu đầu vào và lưu trong cấu trúc dữ liệu

```
KB = [] # array of Sentence
a = Sentence()
nega_of_a = [] # array of Sentence

def read_input():
    input_file = open("input5.txt", "r")
    line1 = input_file.readline().rstrip("\n").split("OR")
    for i in line1:
        a.add_word(Word(i.strip()))

    num = int(input_file.readline())
    for i in range(num):
        line = input_file.readline().rstrip("\n").split("OR")
        new_sen = Sentence()
        for j in line:
            new_sen.add_word(Word(j.strip()))
        KB.append(new_sen)
```

b. Hàm PL_Resolution thực thi thuật toán hợp giải và ghi dữ liệu vào file output

```
def PL_resolution():
    sentences = KB + nega_of_a
    new_sentences = []
    f = open("output.txt", "w")

    while True:
        printed_sentences = []
        for i in range(len(sentences)):
            for j in range(i + 1, len(sentences)):
                result_resolution = sentences[i].resolution(sentences[j])

                for k in result_resolution:
                    if not is_contain(k, sentences) and not is_contain(k, new_sentences):
                        printed_sentences.append(k)
                        new_sentences.append(k)

        # if result_resolution contains any empty sentence, print final result and return
        for k in result_resolution:
            if k.is_empty():
                f.write(str(len(printed_sentences)) + "\n")
                for p in printed_sentences:
                    f.write(p.to_string() + "\n")
                f.write("YES\n")
                f.close()
                return True
```

```
        f.write(str(len(printed_sentences)) + "\n")
        for k in printed_sentences:
            f.write(k.to_string() + "\n")

        # if new_sentences is child of sentences, return False
        if is_child(new_sentences, sentences):
            f.write("NO\n")
            f.close()
            return False

        for i in new_sentences:
            if not is_contain(i, sentences):
                sentences.append(i)
```

4. Chạy test

a. Test 1

input1.txt - Notepad	output.txt - Notepad
File Edit Format View Help	File Edit Format View Help
-A	3
4	-A
-A OR B	B
B OR -C	-C
A OR -B OR C	4
-B	-B OR C
	A OR C
	A OR -B
	{}
	YES

b. Test 2

input2.txt - Notepad	output.txt - Notepad
File Edit Format View Help	File Edit Format View Help
A	2
4	-C
-A OR B	-B OR C
-C OR B	2
A OR C OR -B	-A OR C
-B	A OR -B
	1
	A OR -C
	0
	NO

c. Test 3

input3.txt - Notepad	output.txt - Notepad
File Edit Format View Help	File Edit Format View Help
-B OR A	3
3	-B
-A OR B OR C	A
-B OR A	-C
-C OR A	4
	-A OR C
	B OR C
	-A OR B
	{}
	YES

d. Test 4

input4.txt - Notepad	output.txt - Notepad
File Edit Format View Help	File Edit Format View Help
B OR -A	2
3	B OR C
-A OR B OR C	-A OR C
-B OR A	4
-C OR A	A OR C
	-B OR C
	A OR B
	C
	0
	NO

e. Test 5

input5.txt - Notepad	output.txt - Notepad
File Edit Format View Help	File Edit Format View Help
-A OR B OR -A	3
4	-A
-A OR B	B
B OR -C	-C
A OR -B OR C	4
-B	-B OR C
	A OR C
	A OR -B
	{}
	YES

PHẦN 2: CÂU 2

1. Cấu trúc đề án:

Bài nộp gồm có 4 file như sau:

- royal_family_tree.pl: Chứa cơ sở tri thức và cài đặt các vị từ đối với Cây phả hệ Hoàng gia Anh.
- royal_family_tree_test.txt: Chứa bộ dữ liệu test tạo sẵn của nhóm đối với cơ sở tri thức cây phả hệ Hoàng gia Anh.
- university.pl: Chứa cơ sở tri thức và cài đặt các vị từ đối với cơ sở tri thức tự chọn là tổ chức các lớp trong trường đại học.
- university_test.txt: Chứa bộ dữ liệu test tạo sẵn của nhóm đối với cơ sở tri thức tổ chức các lớp trong trường đại học.

2. Chi tiết cài đặt chương trình Prolog đối với Cây phả hệ Hoàng gia Anh:

* Chương trình Prolog sẽ bao gồm có 2 phần: phần khai báo cơ sở tri thức và phần định nghĩa các vị từ dựa vào cơ sở tri thức sẵn có.

* Cơ sở tri thức được định nghĩa dựa vào cây phả hệ được cho ở mô tả. Cách diễn giải mỗi quan hệ được nhóm sử dụng như sau:

- parent(Parent, Child): Cha mẹ với con cái. Con ở đây là những người có mũi tên rẽ nhánh xuống dưới trong cây phả hệ, không quan tâm là có phải con ruột hay không. Con dâu/con rể sẽ không được tính vào mối quan hệ này.
- male, female(Person): Đàn ông với phụ nữ.
- married(Person, Person): Hai người đang có quan hệ hôn nhân hiện tại, không kể người đã ly dị.
- divorced(Person, Person): Hai người đã ly dị.

* Các vị từ tự định nghĩa được cài đặt theo logic như sau:

- husband(X,Y): X và Y đang có quan hệ hôn nhân và Y phải là nữ.
- wife(X,Y): X và Y đang có quan hệ hôn nhân và Y phải là nam.
- father(X,Y): X phải là bố mẹ của Y và X là nam.
- mother(X,Y): X phải là bố mẹ của Y và X là nữ.
- child(X,Y): Y phải là con của X.
- son(X,Y): X là con của Y và X là nam.
- daughter(X,Y): X là con của Y và X là nữ.
- grandparent(X,Y): X là bố của bố Y.
- grandfather(X,Y): X là ông bà của Y và X là nam.
- grandmother(X,Y): X là ông bà của Y và X là nữ.
- sibling(X,Y): X và Y là anh chị em khi bố mẹ của X cũng là bố mẹ của Y.

- brother(X,Y): X là anh (em trai) của Y nếu như X và Y là anh chị em (sibling) và X là nam.
- sister(X,Y): X là chị (em gái) của Y nếu như X và Y là anh chị em (sibling) và X là nữ.
- aunt(X,Y): Chia làm 2 trường hợp.
 - + AuntDirrect(X,Y): Bố mẹ của Y là chị (em gái) của X.
 - + AuntMarry(X,Y): Y có một chú bác ruột (UncleDirrect) là chồng của X.
 => aunt sẽ được gán bằng cả AuntDirrect và AuntMarry.
- uncle(X,Y): Chia làm 2 trường hợp tương tự như aunt.
- niece(X,Y): X là cháu gái của Y nếu như Y là cô hoặc chú của X (aunt or uncle) và X là nữ.
- nephew(X,Y): X là cháu trai của Y nếu như Y là cô hoặc chú của X (aunt or uncle) và X là nam.

* Đối với các vị từ đơn giản chỉ cần sử dụng chính tham số X và Y thì ta có thể cài đặt đơn giản là liên kết các vị từ bằng quan hệ “and”. Tuy nhiên đối với các vị từ mà để định nghĩa, ta cần sử dụng thêm biến trung gian như quan hệ grandfather, ta sẽ gặp trường hợp kết quả bị trùng lặp rất nhiều. Nếu không trùng lặp, chúng ta lại sẽ gặp phải trường hợp có thêm một kết quả **false** ở sau cùng. Vì thế, đối với những trường hợp như vậy, ta cần cài đặt theo phương pháp như sau:

```
grandchild(X,Y) :- setof((X,Y), Z^(child(X, Z), child(Z, Y), \+X=Y), Gchilds),
    member((X,Y), Gchilds),
    \+ (Y@<X, member((Y,X), Gchilds)).
```

Cài đặt 2 vị từ đối với 2 trường hợp có và không có sử dụng thêm biến trung gian

* Ý nghĩa của cách cài đặt quan hệ grandchild(X,Y) như sau: Đầu tiên, chúng ta tạo ra một tập hợp các kết quả thỏa mãn yêu cầu, sau đó đưa lần lượt chúng vào trong tập Gchilds, tuy nhiên vẫn còn có trùng lặp. Ở dòng tiếp theo, chúng ta sẽ loại bỏ trùng lặp. Cuối cùng, chúng ta tạo ra kết quả từ các cặp còn lại mà không có cặp nào có đối số không theo thứ tự chuẩn.

3. Chạy thử một số testcase đối với Cây phả hệ Hoàng gia Anh:

Câu hỏi 1: Ai là mẹ của Prince Andrew?

```
?- mother(X, 'Prince Andrew').
X = 'Queen Elizabeth II' .
```

Câu hỏi 2: Nữ hoàng Elizabeth có phải vợ của Mia Grace Tindall không?

```
?- wife('Queen Elizabeth II', 'Mia Grace Tindall').
false.
```

Câu hỏi 3: Những ai là cháu của nữ hoàng Elizabeth?

```
?- grandchild(X, 'Queen Elizabeth II').  
X = 'James' ;  
X = 'Lady Louise Windsor' ;  
X = 'Peter Phillips' ;  
X = 'Prince Harry' ;  
X = 'Prince William' ;  
X = 'Princess Beatrice' ;  
X = 'Princess Eugenie' ;  
X = 'Zara Tindall'.
```

Câu hỏi 4: Những ai là chú (bác/cậu/dượng) của Lady Louise Windsor?

```
?- uncle(X, 'Lady Louise Windsor').  
X = 'Prince Andrew' ;  
X = 'Prince Charles' ;  
X = 'Timothy Laurence'.
```

Câu hỏi 5: Những ai là cháu trai (grandson) của Prince Charles?

```
?- grandson(X, 'Prince Charles').  
X = 'Archie Harrison' ;  
X = 'Prince Geogre' ;  
X = 'Prince Louis'.
```

4. Cơ sở tri thức tự chọn: Tổ chức lớp học trong trường đại học

Vì chi tiết cài đặt cũng tương tự hoàn toàn như câu trên nên ở đây, em chỉ trình bày một số quan hệ có trong cơ sở tri thức tự chọn này và đưa ra một số testcase ví dụ.

* Cơ sở tri thức bao gồm 3 quan hệ là:

- schoolyear(Năm học, Lớp).
- faculty(Khoa, Lớp).
- student(Mã số sinh viên, Lớp).

Cơ sở tri thức trong file mã nguồn bao gồm 105 vị từ được khai báo.

* Các vị từ định nghĩa thêm dựa vào các vị từ có sẵn bao gồm:

- schoolYearOfStudent(X,Y): Niên khoá X của sinh viên Y.
- facultyOfStudent(X,Y): Ngành X của sinh viên Y.
- class(X,Y): Lớp X của sinh viên Y.
- friend(X,Y): Hai sinh viên X và Y là bạn học cùng khoá.
- diffirentYearFriend(X,Y): Hai sinh viên X và Y là khác niên khoá.
- classmate(X,Y): Hai sinh viên X và Y là bạn cùng lớp.
- diffirentClassFriend(X,Y): Hai sinh viên X và Y là bạn cùng khoá nhưng khác lớp.
- sameFacultyFriend(X,Y): Hai sinh viên X và Y là bạn cùng khoa.
- sameFacultyClass(X,Y): Hai lớp thuộc cùng một khoa.

5. Chạy thử một số testcase đối với Tổ chức lớp học:

Câu hỏi 1: Lớp 17CNTN thuộc niên khoá 2018 đúng không?

```
?- schoolyear('2018','17CNTN').  
false.
```

Câu hỏi 2: Sinh viên 1712001 học ngành gì?

```
?- facultyOfStudent(X, '1712001').  
X = 'Infomation Technology'.
```

Câu hỏi 3: Những ai là bạn cùng khoá của 1712001?

```
?- friend(X, '1712001').  
X = '1712002' ;  
X = '1712003' ;  
X = '1712004' ;  
X = '1712005' ;  
X = '1712006' ;  
X = '1712007' ;  
X = '1712008' ;  
X = '1712009' ;  
X = '1714001' ;  
X = '1714002' ;  
X = '1714003' ;  
X = '1714004' ;  
X = '1714005' ;  
X = '1714006' ;  
X = '1718001' ;  
X = '1718002' ;  
X = '1718003' ;  
X = '1718004' ;  
X = '1718005' ;  
X = '1718006' .
```

Câu hỏi 4: Hai sinh viên 1718005 và 1818001 là bạn cùng khoa đúng không?

```
?- sameFacultyFriend('1718005', '1818001').  
true.
```

PHẦN 3: CÂU 3

Cách sử dụng:

`python main.py -input <file name>`

1. Cây phả hệ Hoàng gia Anh

Dữ liệu vào được đính kèm vào file nộp (input.pl)

Kiểm chứng kết quả:

1. Ai là mẹ của Prince Andrew?

```
(base) C:\Users\HP\AI_PROJECT2\q3>py main.py
?- mother(X, 'Prince Andrew')
{'X': "'QueenElizabethII'"}
True
```

2. Nữ hoàng Elizabeth có phải vợ của Mia Grace Tindall không?

```
?- wife('Queen Elizabeth II', 'Mia Grace Tindall')
False
~
```

3. Những ai là cháu của nữ hoàng Elizabeth?

```
?- grandchild(X, 'Queen Elizabeth II')
{'X': "'James'"}
{'X': "'PrincessBeatrice'"}
{'X': "'PeterPhillips'"}
{'X': "'ZaraTindall'"}
{'X': "'PrinceWilliam'"}
{'X': "'LadyLouiseWindsor'"}
{'X': "'PrincessEugenie'"}
{'X': "'PrinceHarry'"}
True
```

4. Những ai là chú (bác/cậu/dượng) của Lady Louise Windsor?

```
(base) C:\Users\HP\AI_PROJECT2\q3>py main.py
?- uncle(X, 'Lady Louise Windsor')
{'X': "'PrinceEdward'"}
{'X': "'PrinceCharles'"}
{'X': "'PrinceAndrew'"}
True
?-
```

5. Những ai là cháu trai (grandson) của Prince Charles?

```
?- grandson(X, 'Prince Charles')
{'X': "'ArchieHarrison'"}
{'X': "'PrinceLouis'"}
{'X': "'PrinceGeogre'"}
True
?-
```

6. Những ai là nữ?

```
?- female(X)
{'X': "'PrincessCharlotte'"}
{'X': "'CamillaParkerBowles'"}
{'X': "'QueenElizabethII'"}
{'X': "'PrincessBeatrice'"}
{'X': "'MeghanMarkle'"}
{'X': "'Diana'"}
{'X': "'LadyLouiseWindsor'"}
{'X': "'SophieRhys-Jones'"}
{'X': "'AutumnPhillips'"}
{'X': "'PrincessAnne'"}
{'X': "'ZaraTindall'"}
{'X': "'PrincessEugenie'"}
{'X': "'KateMiddleton'"}
{'X': "'SarahFerguson'"}
True
```


2. Cơ sở trí thức tự chọn: Tổ chức lớp học trong trường đại học

1. Lớp 17CNTN thuộc niên khoá 2018 đúng không?

```
?- schoolyear('2018', '17CNTN').  
False  
?
```

2. Sinh viên 1712001 học ngành gì?

```
?- facultyOfStudent(X, '1712001').  
{'X': "'InfomationTechnology'"}  
True  
?
```

3. Những ai là bạn cùng khoá của 1712001?

```
(base) C:\Users\HP\AI_PROJECT2\q3>py main.py --input input2.pl  
?- friend(X, '1712001').  
{'X': "'1718004'"}  
{'X': "'1718003'"}  
{'X': "'1712002'"}  
{'X': "'1712001'"}  
{'X': "'1718002'"}  
{'X': "'1714006'"}  
{'X': "'1714004'"}  
{'X': "'1718005'"}  
{'X': "'1714003'"}  
{'X': "'1714005'"}  
{'X': "'1712004'"}  
{'X': "'1712006'"}  
{'X': "'1712008'"}  
{'X': "'1718006'"}  
{'X': "'1714002'"}  
{'X': "'1712003'"}  
{'X': "'1712005'"}  
{'X': "'1712007'"}  
{'X': "'1712009'"}  
{'X': "'1714001'"}  
?
```

4. Các lớp thuộc khoa Biotechnology

```
(base) C:\Users\HP\AI_PROJECT2\q3>py main.py --input input2.pl
?- faculty('Biotechnology', X)
{'X': "'18CSH2'"}
{'X': "'17CSH1'"}
{'X': "'16CSH1'"}
{'X': "'16CSH2'"}
{'X': "'17CSH2'"}
{'X': "'18CSH1'"}
True
```

3. Đánh giá

- a. Không dùng được phép vị từ **OR**
- b. Chạy được thuật toán suy diễn lùi và ra kết quả đúng với **Rule**

THAM KHẢO

<https://www.csee.umbc.edu/~finin/prolog/sibling/siblings.html?fbclid=IwAR27JKQ08WmyDeq7N7-pSdxkdQmJlx9dEWTl4Z0U1xZc9lrxp02isXqZwgE>