

# Lab 1: Lập trình Prolog

ThS. Lê Thị Thùy Trang

2024-12-02

## 1. Sơ lược về Prolog

Prolog là một ngôn ngữ lập trình logic (Programming Logic) được áp dụng nhiều trong trí tuệ nhân tạo (Artificial Intelligence) và ngôn ngữ học máy tính (Computational Linguistics).

Prolog có nguồn gốc từ logic bậc nhất (first-order logic), một dạng logic hình thức, và không giống như các ngôn ngữ lập trình khác, Prolog có tính mô tả: một chương trình Prolog được trình bày theo các quan hệ, được biểu diễn như là các sự kiện (fact) và luật (rule). Một chương trình Prolog được sử dụng bằng cách đặt ra một truy vấn trên các quan hệ này. Mục tiêu của Prolog là giúp người dùng mô tả bài toán dựa trên ngôn ngữ logic, theo đó máy tính sẽ tự động suy diễn dựa trên các cơ chế suy diễn có sẵn để tìm ra lời giải cho người dùng.

Hệ thống Prolog đầu tiên được phát triển vào năm 1972 bởi Alain Colmerauer tại Marseille, Pháp. Prolog là một trong số các ngôn ngữ lập trình logic đầu tiên vẫn còn phổ biến tới ngày hôm nay, với một vài dạng thực thi tự do và thương mại. Nó đã được sử dụng trong việc chứng minh định lý (theorem proving), các hệ chuyên gia (expert systems), đặc biệt là trong xử lý ngôn ngữ tự nhiên (Natural Language Processing) vì đây là mục tiêu thiết kế ban đầu của nó.

Trong Prolog có ba cấu trúc cơ bản là sự kiện (hay dữ kiện) (fact), luật (hay quy tắc) (rule) và truy vấn (query). Một tập hợp các sự kiện và các luật được gọi là một cơ sở tri thức (knowledge base) hay cơ sở dữ liệu (database) và lập trình trong Prolog hoàn toàn chỉ là việc viết các cơ sở tri thức. Nghĩa là, một chương trình trong Prolog đơn giản là các cơ sở tri thức.

Để sử dụng một chương trình Prolog, chúng ta chỉ cần đặt ra các truy vấn (query) về các thông tin được lưu trong cơ sở tri thức.

## 2. Toán tử Prolog

Các biểu thức Prolog được tạo thành từ một số ký hiệu của hàm chân lý

Tiếng Việt	Logic mệnh đề	Prolog
NẾU	$P \rightarrow Q$	: -
VÀ	$P \wedge Q$	,
HOẶC	$P \vee Q$	;
PHỦ ĐỊNH	$\neg P$	not hoặc \+
HỢP NHẤT	$X = Y$	=
KHÁC	$X \neq Y$	\=
LỚN HƠN	$X > Y$	>
NHỎ HƠN	$X < Y$	<
LỚN HƠN HOẶC BẰNG	$X \geq Y$	>=
NHỎ HƠN HOẶC BẰNG	$X \leq Y$	=<

### 3. Cú pháp

Một số cú pháp đơn giản trong Prolog

#### 3.1 Facts (Sự kiện)

Sử dụng để định nghĩa các thông tin đã biết

```
has_money(john).  
has_valid_passport(john).  
has_money(mary).
```

Các sự kiện này có nghĩa là:

- John có tiền.
- John có hộ chiếu hợp lệ.
- Mary có tiền.

#### 3.2 Rules (Quy tắc)

Sử dụng để suy luận từ các sự kiện

```
can_travel(Person) :- has_money(Person), has_valid_passport(Person).
```

Quy tắc trên định nghĩa: Một người có thể đi du lịch miễn là họ có tiền và hộ chiếu hợp lệ.

#### 3.3 Queries (Truy vấn)

Sử dụng để hỏi thông tin từ cơ sở dữ liệu.

Để kiểm tra xem “Mary có thể đi du lịch không?”, ta sử dụng truy vấn

```
?- can_travel(mary).
```

Kết quả

```
false.
```

Để trả lời cho câu hỏi “Ai có thể đi du lịch?”, ta sử dụng truy vấn

```
?- can_travel(Who).
```

Kết quả

```
Who = john.
```

Ở ví dụ truy vấn “Ai có thể đi du lịch?”, ta sử dụng *Who*, ở đây *Who* được xem như một biến. Trong lập trình Prolog, các biến bắt đầu bằng chữ cái viết hoa hoặc ký tự `_`.

### 4. Thực hành

#### Lab 1.1

Cho cơ sở tri thức chứa thông tin về mối quan hệ gia đình giữa các cá nhân (cha mẹ, con cái), giới tính của họ, cũng như các quy tắc để xác định ai là mẹ, ai là cha và mối quan hệ anh chị em giữa các thành viên trong gia đình.

Sử dụng trình soạn thảo văn bản để tạo file *family.pl* (Có thể sử dụng sublime text, hoặc cài đặt plug-in prolog trên VScode để viết)

```
father(charles, francois).
father(charles, julie).
father(charles, alain).
father(louis, marie).
father(louis, rose).
father(francois, jean).
father(francois, pierre).
father(emile, paul).
father(jacques, philippe).
father(philippe, henri).
father(remi, chantal).

mother(chantal, francois).
mother(chantal, julie).
mother(chantal, alain).
mother(rose, philippe).
mother(france, henri).
mother(julie, paul).
mother(anne, marie).
mother(anne, rose).
mother(marie, jean).
mother(marie, pierre).
mother(pauline, charles).
```

Sau khi tạo xong file *family.pl*, mở cmd, di chuyển đến thư mục chứa file *family.pl*, sau đó sử dụng lệnh sau để khởi chạy prolog

```
swipl
```

Sau khi khởi động Prolog, tải tệp chứa các mệnh đề logic bằng lệnh sau

```
['family.pl'].
```

**Q1:** Phát biểu cơ sở tri thức trên bằng lời?

**Q2:** Thực hiện các truy vấn

Sau khi tải file chứa các mệnh đề logic, thực hiện truy vấn.

Ví dụ, để truy vấn ai là mẹ của Marie, ta sử dụng cú pháp như sau.

```
?- mother(X, marie).
```

Kết quả trả về:

```
X = anne.
```

Hãy thực hiện các truy vấn sau:

- Julie có phải là mẹ của Paul hoặc Pierre không?
- Charles là cha của những ai?
- Cha và mẹ của Henri tên là gì?

**Q3:** Sửa đổi chương trình Prolog

Tiếp theo, chúng ta tìm cách sửa đổi chương trình PROLOG có trong tệp *family.pl*, bằng cách xác định các quy tắc thiết lập mối quan hệ mới giữa các sự kiện. Các quy tắc mà bạn được yêu cầu xác định sau đây phải được thêm vào

tệp *family.pl*

- Định nghĩa quy tắc  $child(X, Y)$  biểu thị  $X$  là con của  $Y$
- Định nghĩa quy tắc  $male$  và  $female$ , sau đó định nghĩa quy tắc  $son$  và  $daughter$
- Định nghĩa quy tắc  $grandfather$  và  $grandmother$
- Định nghĩa quy tắc  $ancestor$  sao cho  $ancestor(X, Y)$  là đúng nếu  $X$  là tổ tiên của  $Y$
- Định nghĩa quy tắc  $sibling$  sao cho  $sibling(X, Y)$  là đúng nếu  $X$  và  $Y$  có cùng cha và cùng mẹ

## Lab 1.2

Các bảng bên dưới thể hiện số chuyến bay khác nhau giữa các sân bay khởi hành và sân bay đến khác nhau cũng như thành phố nơi mỗi sân bay tọa lạc:

Flight Number	Departure Airport	Arrival Airport
it386	blagnac	cdg
af55	cdg	heathrow
af56	blagnac	gatwick
it387	blagnac	orly
tat32	blagnac	marignane
it856	marignane	cdg
it857	marignane	orly
af65	marignane	heathrow
af66	marignane	gatwick

Bảng 1: Danh sách các chuyến bay.

City	Airport
paris	orly
paris	cdg
london	heathrow
london	gatwick
toulouse	blagnac
marseille	marignane

Bảng 2: Danh sách các thành phố và sân bay.

**Q1:** Hãy biểu diễn các thông tin trên dưới dạng các sự kiện, lưu vào file *flight.pl*

**Q2:** Từ thông tin trên, xác định quy tắc  $vol(N, AD, AA)$ , quy tắc này đúng nếu chuyến bay số  $N$  kết nối sân bay khởi hành  $AD$  và sân bay đến  $AA$

**Q3:** Xác định quy tắc  $airport(C, A)$  đúng nếu  $A$  là sân bay ở thành phố  $C$

**Q4:** Kiểm tra các sự kiện đã khai báo bằng các câu hỏi sau:

- Số lượng chuyến bay thẳng từ Blagnac đến Orly?
- Tất cả các chuyến bay khởi hành từ Blagnac?
- Các sân bay ở London?

**Q5:** Xác định quy tắc  $connection(N, CD, CA)$ , quy tắc này đúng nếu chuyến bay số  $N$  kết nối thành phố khởi hành  $CD$  với thành phố đến  $CA$ . Liệt kê tất cả các chuyến bay đến Paris?