

Chương 8. LẬP TRÌNH PROLOG (PROgramming in LOGic)



TRÍ TUỆ NHÂN TẠO *Artificial Intelligence*

Đoàn Vũ Thịnh
Khoa Công nghệ Thông tin
Đại học Nha Trang
Email: thinhdv@ntu.edu.vn

Nha Trang, 06-2023

Chương 8. LẬP TRÌNH PROLOG (PROgramming in LOGic)

■ Khái niệm

- NNLT Prolog do Alain Colmerauer tạo ra (1972). Prolog được xây dựng trên cơ sở lý thuyết của logic vị từ, nhằm biểu diễn các tri thức cùng các mối quan hệ cơ bản giữa chúng. Từ đó, khi đã được trang bị sẵn một cơ chế suy luận thông qua một mô-đen suy diễn, Prolog sẽ sản sinh ra và kiểm tra các mối quan hệ hệ quả khác.
- Prolog thuộc nhóm NNLT mô tả: Để giải quyết một bài toán, đặc biệt là các bài toán xử lý ký hiệu trừu tượng và suy luận lôgic, ta chỉ cần chọn cách biểu diễn phù hợp và mô tả các đối tượng cùng các tính chất và mối quan hệ cơ bản đã biết giữa chúng.

Chương 8. LẬP TRÌNH PROLOG (PROgramming in LOGic)

■ Khái niệm

- Thông qua môơ suy diễn của Prolog, ta sẽ nhận được các câu trả lời liên quan đến các mối quan hệ nhân quả và các kết luận liên quan đến bài toán cần giải quyết.
- Khi tri thức đã biết về bài toán không tăng thêm, việc trả lời các kết luận khác liên quan đến bài toán là thông qua các câu hỏi thêm mà không cần phải lập trình lại như các NNLT kiểu thủ tục.
 - `la_nguoi(socrates).` `%socrates la nguoi`
 - `phai_chet(X):-la_nguoi(X).` `%la nguoi thi ai cung phai chet`
 - `%Cau truy van:`
 - `?- la_nguoi(mickey).` `→ FALSE`
 - `la_nguoi(socrates).` `→ TRUE`

Chương 8. LẬP TRÌNH PROLOG (PROgramming in LOGic)

■ Khái niệm

- Các khái niệm chính trong Prolog là: vị từ (predicate), đối tượng, biến, sự kiện (fact), qui tắc (rule), câu hỏi (đích, mục tiêu: Goal), danh sách.
- Vài kỹ thuật thường được dùng trong Prolog là: quay lui (mặc định là tìm kiếm theo chiều sâu trên cây suy diễn), đệ qui, lát cắt.

Chương 8. LẬP TRÌNH PROLOG (PROgramming in LOGic)

■ Các bước thực thi

- Bước 1. Khám phá tri thức bài toán và hình thức hóa thành logic vị từ
- Bước 2. Tổ chức, tinh giản và chuyển tri thức về dạng **Horn**
- Bước 3. Viết tri thức Horn thành chương trình logic đúng cú pháp.
- Một số quy tắc biến đổi dạng chuẩn Horn:
 - $\neg a \cup b \equiv a \rightarrow b$
 - $p \cup q \rightarrow r \equiv p \rightarrow r, q \rightarrow r$
 - $(p \cap q) \rightarrow r$
 - $p \rightarrow (\neg q \cup r) \equiv \neg p \cup (\neg q \cup r) \equiv (\neg p \cup \neg q) \cup r \equiv \neg(p \cap q) \cup r \equiv p \cap q \rightarrow r$
 - $p \rightarrow (q \cap r) \equiv \neg p \cup (q \cap r) \equiv \neg p \cup q \cap \neg p \cup r \equiv p \rightarrow p, p \rightarrow r$
 - $(p \rightarrow q) \rightarrow r \equiv \neg(\neg p \cup q) \cup r \equiv (\neg\neg p \cap \neg q) \cup r \equiv (p \cap \neg q) \rightarrow r$
 - $p \rightarrow (q \rightarrow r) \equiv \neg p \cup (\neg q \cup r) \equiv (\neg p \cup \neg q) \cup r \equiv \neg(p \cap q) \cup r \equiv p \cap q \rightarrow r$

Chương 8. LẬP TRÌNH PROLOG (PROgramming in LOGic)

■ Các bước thực thi

- Truy vấn trong Prolog là 1 câu logic có dạng truy vấn hạn định, là câu có dạng $\exists(A_1 \wedge A_2 \wedge \dots \wedge A_n)$ với A_i là các biểu thức đơn.
- Việc suy diễn tri thức là việc kiểm tra, xem xét một tri thức nào đó có phải là hệ quả logic của một cơ sở tri thức hay không?
- Câu trả lời của Prolog = false, tức tri thức cần kiểm tra không là hệ quả của chương trình.
- Xét ví dụ:
 - `la_nguoi(socrates).` %socrates la nguoi
 - `phai_chet(X):-la_nguoi(X).` %la nguoi thi ai cung phai chet

Chương 8. LẬP TRÌNH PROLOG (PROgramming in LOGic)



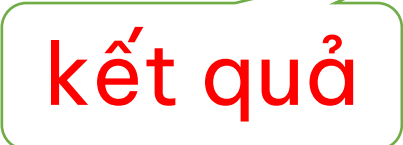

■ Các bước thực thi

- Với câu truy vấn “la_nguoi(socrate).” muốn hỏi “socrate” có phải là người không thì câu trả lời là true vì “la_nguoi(socrate).” là hệ quả logic được suy diễn trực tiếp từ chương trình trên.
- Với câu truy vấn “phai_chet(socrate).” muốn hỏi “socrate” có phải chết không sẽ có câu trả lời là true vì “phai_chet(socrate).” được suy diễn từ câu (2) và (1) của chương trình trên.
- Truy vấn “phai_chet(mickey).” sẽ có câu trả lời là false vì đây không phải là hệ quả của chương trình trên.

Chương 8. LẬP TRÌNH PROLOG (PROgramming in LOGic)

- Các bước thực thi

- Xem xét chương trình sau:

-  `la_nguoi(socrates).`  `%socrates la nguoi`
- `phai_chet(X):-la_nguoi(X).` `%la nguoi thi ai cung phai chet`
 

- Với dữ liệu là biến, PROLOG tự thêm $\forall X$: không cần viết

Chương 8. LẬP TRÌNH PROLOG (PROgramming in LOGic)

■ Ví dụ

- Dùng Prolog biểu diễn tri thức sau:
 - 1. Lan là nữ
 - 2. Mẹ của Lan là nữ
 - 3. Mai yêu Nam
 - 4. Nam yêu hết thấy mọi người
- Truy vấn các tri thức sau:
 - Lan có phải là nữ không?
 - Mai có phải là nữ không?
 - Nam có phải là nam không?
 - Có ai đó là nữ không?
- `la_nu(lan).`
- `la_nu(me_cua(lan)).`
- `yeu(mai,nam).`
- `yeu(nam,_).`
- `la_nu(X) :- X = me_cua(Y).`
- `?-la_nu(lan). = TRUE`
- `?-la_nu(mai). = FALSE`
- `?-la_nam(nam). = LỖI`
- `?-la_nu(X).`
 - `X = lan ; X = me_cua(lan) ;`
 - `X = me_cua(_);`

Chương 8. LẬP TRÌNH PROLOG (PROgramming in LOGic)

■ Ví dụ

■ Truy vấn các tri thức sau:

- Nam có yêu Mai không?
- có ai yêu mẹ của mình không?

- có ai đó yêu nhau không?

$yeu(X, Y) \cap yeu(Y, X)$

- ?- `yeu(nam, mai).` = TRUE
- ?- `yeu(nam, ba_ngoi(mai)).` = TRUE
- ?- `yeu(Y, me_cua(X)).`
 - `Y = nam ;`
- ?- `yeu(X, Y), yeu(Y, X).`
 - `X = mai,`
 - `Y = nam ;`
 - `X = nam,`
 - `Y = mai ;`
 - `X = Y, Y = nam ;`

HẾT CHƯƠNG 8