

# BÀI GIẢNG MÔN HỌC :

## Trí Tuệ Nhân Tạo

### Nội dung :

- Tổng quan về trí tuệ nhân tạo
- Các phương pháp giải quyết vấn đề cơ bản
- Hệ chuyên gia
- Các phương pháp biểu diễn tri thức
- Vài ứng dụng trí tuệ nhân tạo lý giải với logic
- Xử lý tri thức không chắc chắn
- Việc học máy

# Chương 1 :

## Tổng Quan Về Trí Tuệ Nhân Tạo

**Chương này gồm có :**

- Trí tuệ nhân tạo là gì ?
- Lịch sử phát triển trí tuệ nhân tạo
- Các thành phần cơ bản của trí tuệ nhân tạo
- Giới Thiệu Về Ngôn Ngữ Lập Trình Prolog

# Trí tuệ nhân tạo là gì ?

- Trí tuệ nhân tạo là lĩnh vực khoa học chuyên nghiên cứu các phương pháp chế tạo trí tuệ máy sao cho giống như trí tuệ con người.
- Vài định nghĩa của trí tuệ nhân tạo điển hình là
  - Hệ thống mà biết suy nghĩ như con người
  - Hệ thống mà biết hành động như con người
- Để hệ thống mà biết suy nghĩ và hành động giống như con người thì hệ thống đó phải có tri thức, phải có khả năng lý giải, phải có khả năng học, phải có thị giác và thính giác đồng thời phải biết di chuyển giống như con người.

- Trí tuệ con người là gì ? Là khả năng giải quyết vấn đề của con người.
- Thường, khả năng giải quyết vấn đề của con người thông qua bốn thao tác cơ bản là

➤ **Xác định tập hợp đích**

- ❖ Xét quá trình suy nghĩ giải quyết vấn đề của con người, quá trình suy nghĩ này phải có điểm bắt đầu và điểm kết thúc.
- ❖ Điểm bắt đầu được gọi là trạng thái ban đầu và điểm kết thúc được gọi là trạng thái đích của bài toán.
- ❖ Giữa hai trạng thái của quá trình suy nghĩ được phân rã ra nhiều mảnh nhỏ, mỗi mảnh nhỏ được gọi là đích từng phần.
- ❖ Tập các đích từng phần này định hướng cách giải quyết vấn đề của con người.

➤ **Thu thập các sự kiện và luật suy diễn**

- ❖ Con người thường xuyên thu thập các sự kiện và công thức hóa các sự kiện thành luật để làm cơ sở tri thức giải quyết vấn đề.
- ❖ Luật là mối quan hệ giữa vài sự kiện đã biết và một vài sự kiện chưa biết.
- ❖ Tri thức là sự hiểu biết về một lĩnh vực nào đó.

## ➤ Cơ chế tập trung

- ❖ Khi giải quyết vấn đề, có nhiều đường suy nghĩ khác nhau xuất hiện cùng lúc.
- ❖ Để tập trung bám theo đường suy nghĩ giải quyết vấn đề và đồng thời loại bỏ các đường suy nghĩ khác nhờ một cơ chế có sẵn trong bộ óc con người đó là cơ chế tập trung.

## ➤ Bộ máy suy diễn

- ❖ Quá trình giải quyết vấn đề là một quá trình lập luận từ sự kiện đã biết để suy diễn ra sự kiện chưa biết.
- ❖ Quá trình này được lặp lại cho đến khi dẫn đến đích.
- ❖ Để làm được điều này, nhờ một cơ chế có sẵn trong bộ óc con người đó là bộ máy suy diễn.

- **Trí tuệ máy là gì ?** Trí tuệ máy là khả năng giải quyết vấn đề của máy sao cho giống khả năng giải quyết vấn đề của con người đó là

- Hành động giống như con người.
  - Suy nghĩ giống như con người.
  - Học thích nghi với tình huống giống như con người.
  - Xử lý thông tin giống như con người.
  - Hành động và suy nghĩ trên cơ sở logic và chính xác giống như con người.
- Nền tảng của trí tuệ nhân tạo cần đến các nguồn tri thức của nhiều lĩnh vực khoa học khác nhau như
    - ❖ Khoa học máy tính và kỹ thuật máy tính ( chế tạo phần mềm và phần cứng).
    - ❖ Triết học ( Thiết kế luật suy diễn lý giải )
    - ❖ Toán học ( Kiến thức suy luận logic, thiết kế thuật toán và tối ưu hóa )
    - ❖ Tri học và tâm lý học (Mô hình hóa suy nghĩ con người ở mức cao )
    - ❖ Khoa học thần kinh ( Mô hình hóa hoạt động bộ óc con người ở mức thấp )
    - ❖ Ngôn ngữ học ( tạo ngôn ngữ nói tự nhiên cho máy )

# Lịch sử phát triển trí tuệ nhân tạo

- Ý tưởng chế tạo trí tuệ máy đã có từ lâu nhưng mãi đến năm 1950, nhà toán học người Anh công bố công trình khoa học của ông ta đó là “Máy tính và Thông minh”.
- Đây được xem như là mốc lịch sử bắt đầu phát triển khoa học trí tuệ nhân tạo. Những năm ngay sau đó có các công trình được công bố như
  - ❖ Năm 1956, chương trình giải bài toán tổng quát.
  - ❖ Năm 1958, chương trình chứng minh định lý hình học.
- Đỉnh cao của việc phát triển ở lĩnh vực này phải nói đến những năm 1960. Dù rằng còn bị hạn chế nhiều về trang thiết bị nhưng những năm này đã có nhiều công trình được công bố như
  - ❖ Năm 1960, ngôn ngữ Lisp.
  - ❖ Năm 1961, chương trình giải các bài toán đại số sơ cấp.
  - ❖ Năm 1963, chương trình trò chơi cờ vua.
  - ❖ Năm 1964, chương trình tính tích phân.
  - ❖ Năm 1966, chương trình phân tích và học nói.
  - ❖ Năm 1968, chương trình điều khiển Robot theo phương án mắt và tay.

- Năm 1972, ngôn ngữ Prolog.
- Từ những năm 1969 đến năm 1999, có nhiều chương trình được xây dựng trên các hệ cơ sở tri thức.
- Thật vậy, lĩnh vực trí tuệ đã đi vào đời sống dân dụng từ những năm 1980 đến nay.



# Các thành phần cơ bản của trí tuệ nhân tạo

- Có hai thành phần cơ bản của trí tuệ nhân tạo đó là biểu diễn tri thức và tìm kiếm tri thức trong miền biểu diễn.
- Tri thức của bài toán có thể được phân ra làm ba loại tri thức cơ bản đó là tri thức mô tả, tri thức thủ tục và tri thức điều khiển.
  - ❖ **Tri thức mô tả** : mô tả những gì được biết về bài toán. Loại tri thức này bao gồm các sự kiện, các quan hệ và các tính chất của bài toán.
  - ❖ **Tri thức thủ tục** : mô tả tổng quát cách giải quyết bài toán. Loại tri thức này bao gồm luật suy diễn hợp lệ, chiến lược tìm kiếm và giải thuật tìm kiếm.
  - ❖ **Tri thức điều khiển** : là luật chủ chốt để điều khiển quá trình tìm kiếm giải quyết bài toán.

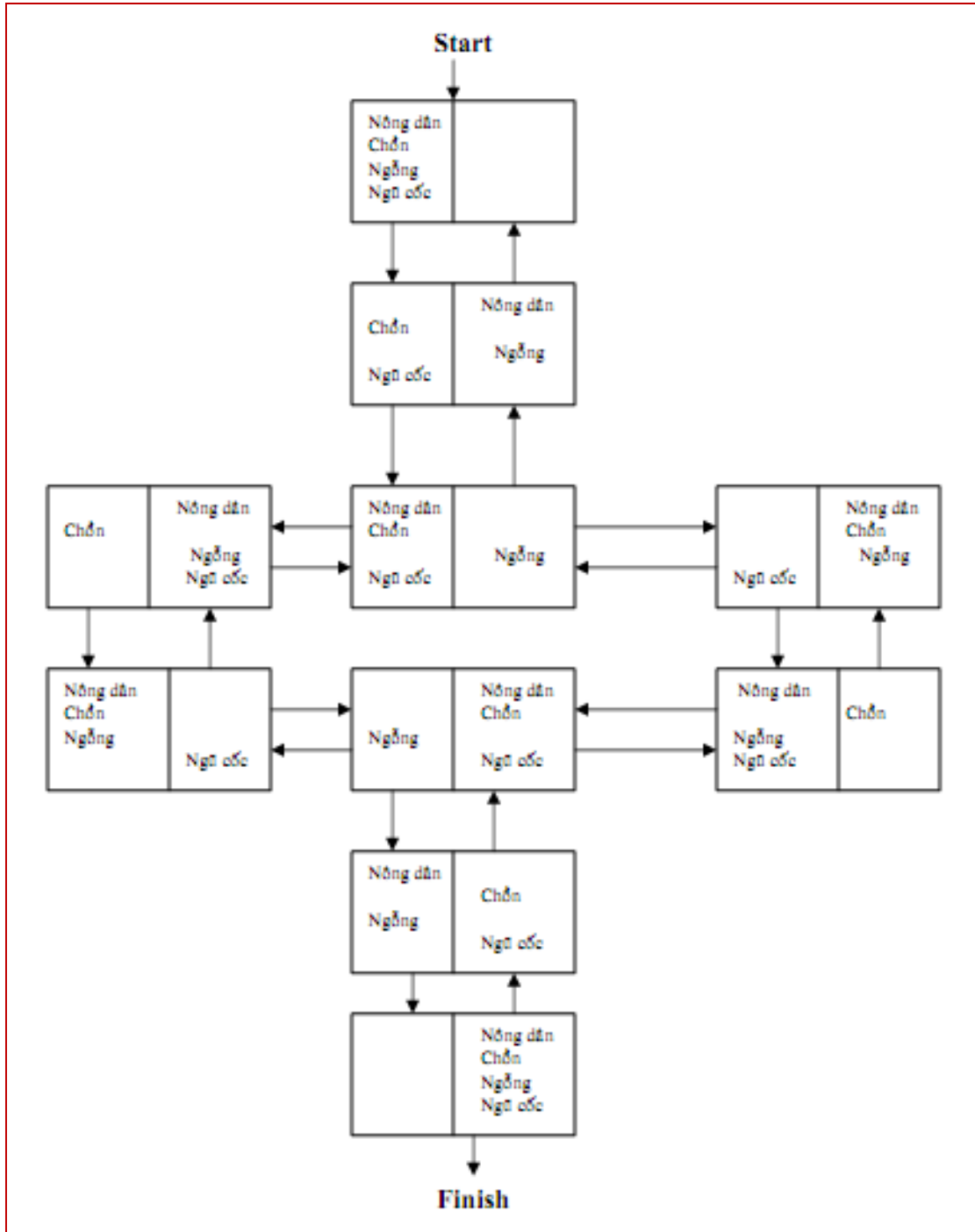
- Như vậy, **biểu diễn tri thức là gì** ? Đó là phương pháp thể hiện tri thức trong máy sao cho đủ và hiệu lực để bài toán được giải tốt nhất.

**Ví dụ** : Xét bài toán người nông dân, chồn, ngỗng và ngũ cốc.

- Bài toán đặt ra là người nông dân muốn mang theo với mình một con chồn, một con ngỗng và một số ngũ cốc qua bên kia sông bằng một chiếc thuyền.
- Tuy nhiên, thuyền của ông ta quá bé chỉ có thể mang theo một thứ duy nhất với ông ta trên mỗi chuyến thuyền sang sông.
- Nếu ông ta để lại chồn và ngỗng bên này sông thì chồn sẽ ăn ngỗng và nếu ông ta để lại ngỗng và ngũ cốc thì ngỗng sẽ ăn hết số ngũ cốc.
- Hãy sắp xếp các chuyến thuyền qua lại sông sao cho người nông dân mang mọi thứ sang bên kia sông an toàn?

- Với bài toán này, cách biểu diễn tri thức tốt nhất có thể vạch ra các ràng buộc vốn sẵn có trong bài toán đó là
  - ❖ xây dựng một biểu đồ với các nút có đánh nhãn biểu diễn người nông dân mang theo thứ mà ông ta cần phải mang theo trên mỗi chuyến thuyền
  - ❖ và các cạnh liên kết giữa các nút biểu diễn bằng các đường mũi tên chỉ các chuyến thuyền qua lại sông.
- Cách biểu diễn này hàm chứa các thành phần như ngữ từ học, cấu trúc, thủ tục và ngữ nghĩa.
  - ❖ Ngữ từ học (Lexical) : là các từ vựng hợp lệ được sử dụng như các ký hiệu trong biểu diễn.
  - ❖ Cấu trúc (Structure) : là các đường mũi tên liên kết giữa các nút chỉ định các chuyến thuyền qua lại sông.
  - ❖ Thủ tục (Procedure) : là mô tả cách giải bài toán từ nút này đến nút kia nhờ thông các đường chỉ định mũi tên.
  - ❖ Ngữ nghĩa (Semantic) : là ý nghĩa của các nút và các cạnh liên kết thông qua cách giải bài toán.

Biểu đồ biểu diễn tri thức của bài toán người nông dân được mô tả như hình



- Với người nông dân và ba thứ ông ta muốn mang theo ở bên này sông hoặc bên kia sông, ta có  $2^{1+3} = 16$  lần sắp xếp khác nhau, trong đó có 10 lần sắp xếp qua lại sông an toàn và 6 lần sắp xếp khác qua lại sông không an toàn đó là
  - ❖ Chồn, ngỗng và ngũ cốc bên này sông hoặc bên kia sông,
  - ❖ Ngỗng và ngũ cốc bên này sông hoặc bên kia sông
  - ❖ Chồn và ngỗng bên này sông hoặc bên kia sông.
- Khi tri thức của bài toán đã được biểu diễn, phương pháp giải bài toán trong lĩnh vực trí tuệ nhân tạo đó là kỹ thuật tìm kiếm trong miền biểu diễn tri thức của bài toán đó.
- Để giải bài toán người nông dân, kỹ thuật tìm kiếm là bằng cách bám dò tìm theo các đường mũi tên bắt đầu từ nút khởi tạo liên kết qua các nút để đi đến đích trong biểu diễn của bài toán.

# Giới Thiệu Về Ngôn Ngữ Lập Trình Prolog

- Ngôn ngữ lập trình Prolog là công cụ lập trình suy luận logic dành riêng cho lĩnh vực trí tuệ nhân tạo với các đặc điểm như sau :
  - ❖ Nó là ngôn ngữ cho phép xử lý song song và đệ quy.
  - ❖ Nó là ngôn ngữ cho phép liên kết với nhiều loại ngôn ngữ lập trình khác như C, Assembler và Pascal.
- Giống như các ngôn ngữ lập trình khác, cấu trúc chương trình của ngôn ngữ Prolog với các từ khóa được mô tả như sau :
  - **domains**  
<Vùng khai báo biến >
  - **predicates**  
< Vùng khai báo hàm vị từ >
  - **clauses**  
< Vùng khai báo các mệnh đề sự kiện và mệnh đề luật>
  - **goal**  
< Vùng khai báo mệnh đề đích của chương trình>

- **domains** : là từ khóa cho phép vùng khai báo biến trong một chương trình Prolog với các kiểu dữ liệu không chuẩn và dữ liệu chuẩn. Dữ liệu chuẩn của prolog là **short, word, integer, real, string, symbol**.

- Cú pháp khai báo biến với các kiểu dữ liệu là

**<Variable\_name> = <Data\_type>**

- Ví dụ : Khai báo biến d với kiểu dữ liệu integer và List là biến danh sách chứa các số integer với các câu lệnh là

**domains**

**d = integer**

**List = d\***

- **predicates** : là từ khóa cho phép vùng khai báo các hàm vị từ.

- Ví dụ : Khai báo hàm vị từ tính father với hai biến đối số của nó là X và Y kiểu symbol với các câu lệnh là

domains

$x = \text{symbol}$

predicates

$\text{father}(X, Y)$

- **Lưu ý** : Các biến trong các hàm vị từ tính là chữ cái in hoa hoặc chuỗi bắt đầu với chữ cái in hoa.
- **clauses** : là từ khóa cho phép vùng khai báo các mệnh đề sự kiện và mệnh đề luật suy diễn.
  - **Mệnh đề sự kiện** : là các hàm vị từ tính được thiết lập để khai báo các sự kiện. Cú pháp tổng quát của hàm vị từ là  $\text{predicate\_name}(<\text{Const1}>, \dots, <\text{ConstN}>)$ .
  - **Lưu ý** : Các đối số trong các vị từ tính sự kiện là các chuỗi hằng số với các chữ cái in thường.
  - **Ví dụ** : Sự kiện cho rằng John là father của marry, sự kiện này được khai báo với mệnh đề sự kiện father là  $\text{father}(\text{john}, \text{marry})$ .



- **Mệnh đề luật suy diễn** : Để khai báo luật suy diễn **If <Conditions> Then <Clusion>** trong chương trình Prolog, mệnh đề luật suy diễn được khai báo với cú pháp tổng quát là

**predicate\_name(<Clusion>):- predicate\_name(<Conditions>).**

Trong đó, vế trái của mệnh đề là tương ứng với vế kết luận và vế phải của mệnh đề là tương ứng với vế điều kiện của luật if-Then.

- **Ví dụ** : Chương trình prolog với mệnh đề sự kiện và luật là

```
domains
    x = symbol
predicates
    nondeterm    father(X, Y)
    nondeterm    grandfather(X,Z)
clauses
    father(john, marry).
    father(bod, john).
    grandfather(X,Z) :- father(X,Y), father(Y,Z).
goal
    grandfather(X,Z), write(X, "Grandfather of ", Z),nl.
```

Chạy chương trình cho kết quả là **Bod is grandfather of marry**

## ■ Toán tử trong prolog :

### ■ Toán tử logic :

- ❖ Toán tử and : ,
- ❖ Toán tử or : ;
- ❖ Toán tử not : not

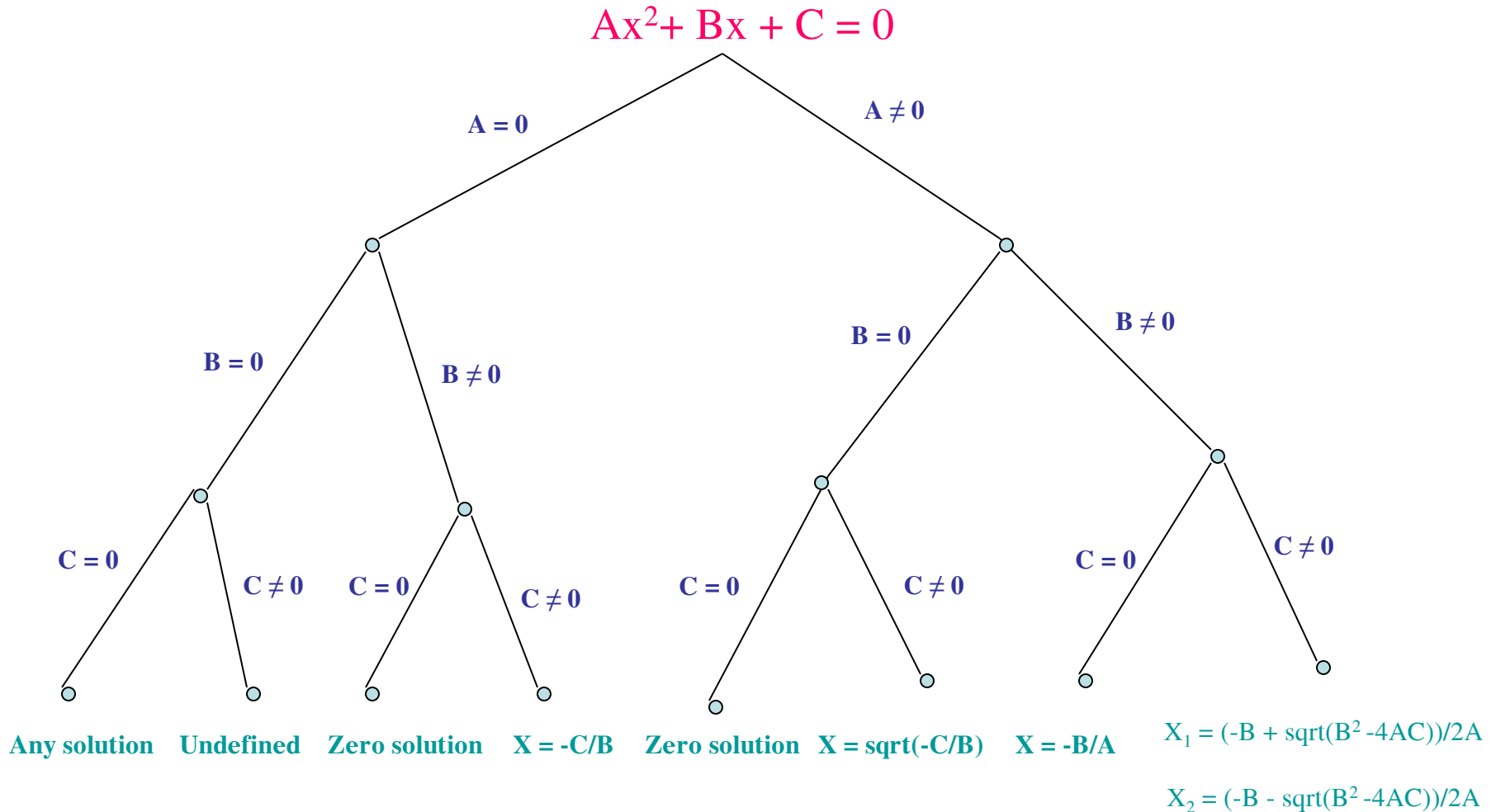
### ■ Toán tử so sánh :

- ❖ Toán tử nhỏ hơn : <
- ❖ Toán tử nhỏ hoặc bằng : <=
- ❖ Toán tử bằng nhau : =
- ❖ Toán tử lớn hơn : >
- ❖ Toán tử lớn hoặc bằng : >=
- ❖ Toán tử không bằng : <>

### ■ Toán tử số học :

- ❖ Cộng : +
- ❖ Trừ : -
- ❖ Nhân : \*
- ❖ Chia : /

**Ví dụ** : Giải phương trình bậc 2 :  $Ax^2 + Bx + C = 0$  với thuật toán được mô tả như hình



- Luật If- Then giải phương trình bậc 2 với thuật toán hình cây được thiết lập là
  - ❖ If  $A = 0$  and  $B = 0$  and  $C = 0$  Then Display Any solution.
  - ❖ If  $A = 0$  and  $B = 0$  and  $C \neq 0$  Then Display Undefined solution.
  - ❖ If  $A = 0$  and  $B \neq 0$  and  $C = 0$  Then Display Zero solution
  - ❖ If  $A = 0$  and  $B \neq 0$  and  $C \neq 0$  Then Display  $x = -C/B$ .
  - ❖ If  $A \neq 0$  and  $B = 0$  and  $C = 0$  Then Display Zero solution.
  - ❖ If  $A \neq 0$  and  $B = 0$  and  $C \neq 0$  Then Display  $x = \pm\sqrt{-C/A}$ .
  - ❖ If  $A \neq 0$  and  $B \neq 0$  and  $C = 0$  Then Display  $x = 0$  and  $x = -B/A$ .
  - ❖ If  $A \neq 0$  and  $B \neq 0$  and  $C \neq 0$  Then Display  $x = (-B \pm\sqrt{B^2 - 4AC})/2A$ .
- Chương trình Prolog được thiết lập trên cơ sở luật If-Then giải bài toán phương trình bậc 2 là
  - domains
    - r = real
  - predicates
    - solve(R, R, R)

## clauses

```
solve(A, B, C) :- A = 0, B = 0, C = 0, write("Any solution "),nl,!.  
solve(A, B, C) :- A = 0, B = 0, C <> 0, write("Undefined solution "),nl,!.  
solve(A, B, C) :- A = 0, B <> 0, C = 0, write("Zero solution "),nl,!.  
solve(A, B, C) :- A = 0, B <.0, C <> 0, X = -C/B, write("X = ",X),nl,!.  
solve(A, B, C):- A <> 0, B = 0, C = 0, write("Zero solution "),nl,!.  
solve(A, B, C) :- A <> 0, B = 0, C <> 0, D = -C/A, D > 0,  
    X1 = sqrt(D), X2 = -sqrt(D),  
    write("X1 = ", X1," X2 = ",X2),nl,!.  
solve(A, B, C) :- A <> 0, B <> 0, C = 0, X1 = 0, X2 = -B/A,  
    write("X1 = ",X1," X2 = ",X2),nl,!.  
solve(A, B, C) :- A <> 0, B <> 0, C <> 0, D = B*B - 4*A*C, D > 0,  
    X1 = (-B + sqrt(D))/(2*A), X2 = (-B - sqrt(D))/(2*A),  
    write("X1 = ", X1," X2 = ",X2),nl,!.  

```

## goal

```
write("Enter coefficient A : ), readreal(A),  
write("Enter coefficient A : ), readreal(A),  
write("Enter coefficient A : ), readreal(A),  
solve(A, B, C).
```

## ■ Xử lý danh sách :

- Xác định thành viên của danh sách với các mệnh đề là

`member(X, [X | _ ]) :- !.`

`member(X, [_ | L ]) :- member(X, L).`

Nếu X là thành viên của danh sách L thì mệnh đề trả về giá trị logic true; mặt khác trả về giá trị logic false.

- Nối hai danh sách với các mệnh đề là

`append([ ], L, L).`

`append([X | L1], L2, [X | L3]) :- append(L1, L2, L3).`

Nối hai danh sách L1 và L2 cho kết quả L3.

- Hiển thị nội dung của danh sách với các mệnh đề là

`writelist([ ]).`

`Writelist([H | T]) :- write(H), nl, writelist(T).`

- Hiển thị đảo nội dung của danh sách với các mệnh đề là

`reverse_writelist([ ]).`

`reverse_writelist([H | T]) :- reverse_writelist(T), write(H),nl.`

- Loại bỏ phần tử đầu tiên của danh sách T với mệnh đề là

`dequeue(E, [E | T], T).`

- Cộng một phần tử vào cuối danh sách với các mệnh đề là  
 $\text{add\_to\_queue}(E, [], [E]).$   
 $\text{add\_to\_queue}(E, [H \mid T], [H \mid T_{\text{new}}]) \text{ :- } \text{add\_to\_queue}(E, T, T_{\text{new}}).$
- Cộng thêm một phần tử vào đầu danh sách stack với mệnh đề là  
 $\text{stack}(\text{Top}, \text{Stack}, [\text{Top} \mid \text{Stack}]).$
- Cộng thêm một danh sách vào cuối danh sách stack với mệnh đề là  
 $\text{add\_list\_to\_stack}(L, \text{Stack}, \text{Result}) \text{ :- } \text{append}(L, \text{Stack}, \text{Result}).$
- Xác định thành viên là cXa1cdanh sách stack với mệnh đề là  
 $\text{member\_stack}(E, \text{Stack}) \text{ :- } \text{member}(E, \text{Stack}).$
- Hiển thị đảo nội dung của stack với các mệnh đề là  
 $\text{reverse\_print\_stack}(S) \text{ :- } \text{empty\_stack}(S).$   
 $\text{reverse\_print\_stack}(S) \text{ :- } \text{stack}(E, \text{Res}, S), \text{reverse\_print\_stack}(\text{Res}),$   
 $\text{wrie}(E), \text{nl}.$

- Xác định thành viên thuộc là của tập hợp S với mệnh đề là  
`member_set(X, S) :- member(X,S).`
- Cộng thành viên không thuộc về danh sách vào đầu danh sách với các mệnh đề là  
`add_if_not_in_set(X, L, L) :- member(X, L), !.`  
`add_if_not_in_set(X, L, [X | L]).`
- Xác định tập hợp là tập con của một tập hợp khác với các mệnh đề là  
`subset([ ], _ ).`  
`Subset({H | T}, L) :- member_set(H, L), subset(T,L).`
- Liệt kê tất cả các giá trị của cùng thuộc tính trong lớp sử dụng lệnh findall với cú pháp là  
`findall(Property_name, Predicate_name, List).`

**Ví dụ :**

`predicates`

`person(Name, Address, Age)`

`clauses`

`person("Alan", "22 Bstreet", 42).`

`person("Peter", "21 C street", 36).`

`goal`

`findall(Age, person(_,_,Age), L).`