

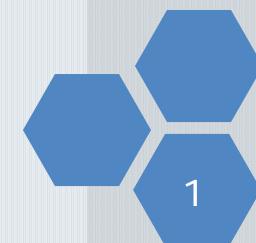
# Bài giảng Trí tuệ nhân tạo

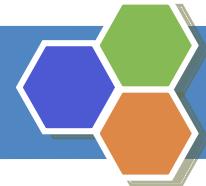


## Chương 1 Tổng quan về Trí tuệ nhân tạo



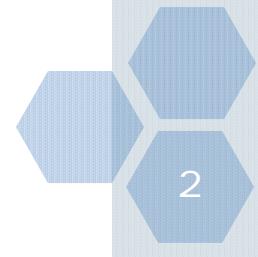
Bùi Đức Dương  
*Khoa Công nghệ Thông tin*

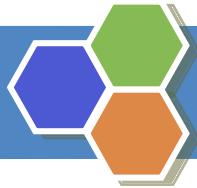




# Nội dung

- 1. Giới thiệu về AI (*Artificial Intelligence*)**
- 2. Đặc điểm của AI**
- 3. Lịch sử phát triển AI**
- 4. Các lĩnh vực nghiên cứu & ứng dụng AI**
- 5. Thuật toán - thuật giải**





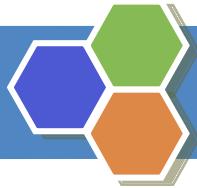
# 1. Giới thiệu về AI

## ❖ Trí tuệ

- Phản ứng một cách thích hợp những tình huống thông qua điều chỉnh hành vi hợp lý.
- Hiểu rõ mối quan hệ giữa các sự kiện của thế giới quan để đưa ra những hành vi phù hợp nhằm đạt được mục đích.

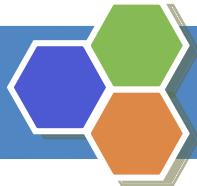
## ❖ Tri thức

- Là sự hiểu biết (thường là sâu) về một vấn đề (lĩnh vực). Ví dụ: Hiểu biết về y khoa, về máy tính, về thời tiết...
- Là thông tin chứa đựng 2 phần: Các khái niệm (cơ bản & phát triển) và các phương pháp nhận thức (quy luật, suy diễn...)
- Là rière kiện tiên quyết của các hành xử “thông minh”.
- Có được thông qua hành vi thu thập tri thức và sản sinh tri thức – hai quá trình song song và nối tiếp trong một thực thể “thông minh”.



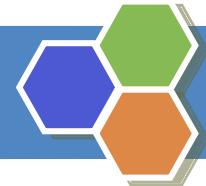
# 1. Giới thiệu về AI (tt)

- ❖ **Hành xử thông minh (intelligent behaviour)**
  - Là các hoạt động của một đối tượng như là kết quả của một quá trình thu thập, xử lý và điều khiển theo những tri thức đã có hay mới phát sinh.
  - Kết quả thường tốt theo mong đợi so với các hành xử thông thường
  - Khái niệm về tính thông minh của một đối tượng thường biểu hiện qua các hoạt động:
    - ✓ Sự hiểu biết và nhận thức được tri thức
    - ✓ Sự lý luận tạo ra tri thức mới dựa trên tri thức đã có
    - ✓ Hành động theo kết quả của các lý luận
    - ✓ Kỹ năng (Skill)



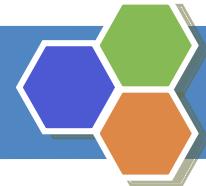
# 1. Giới thiệu về AI (tt)

- Hành xử thông minh không đơn thuần là các hành động như là kết quả của quá trình thu thập tri thức và suy luận trên tri thức.
- Hành xử thông minh còn bao hàm:
  - ✓ Sự tương tác với môi trường để nhận các phản hồi
  - ✓ Sự tiếp nhận các phản hồi để điều chỉnh hành động - Skill
  - ✓ Sự tiếp nhận các phản hồi để hiệu chỉnh và cập nhật tri thức
- Tính chất thông minh là sự tổng hợp của cả 3 yếu tố: thu thập tri thức, suy luận và hành xử của đối tượng trên tri thức thu thập được.
- Không thể hành giá riêng lẽ bất kỳ một khía cạnh nào để nói về tính thông minh.



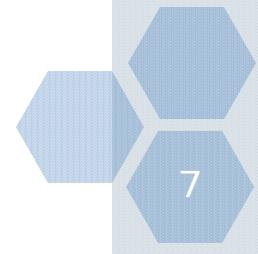
# 1. Giới thiệu về AI (tt)

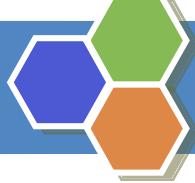
- ❖ AI là gì?
  - “**An AI approach problem-solving is one which** (George Luger):
    - ✓ uses domain-specific knowledge
    - ✓ to find a good-enough solution
    - ✓ to a hard problem
    - ✓ in a reasonable amount of time.”



## 1. Giới thiệu về AI (tt)

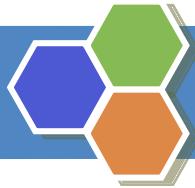
- ❖ AI là ngành nghiên cứu về các hành xử thông minh bao gồm: thu thập, lưu trữ tri thức, suy luận, hoạt động và kỹ năng.
- ❖ Đối tượng nghiên cứu là các “hành xử thông minh” chứ không phải là “sự thông minh”.
- ❖ ‘Không có’ Sự Thông Minh mà Chỉ có Biểu hiện thông minh qua hành xử





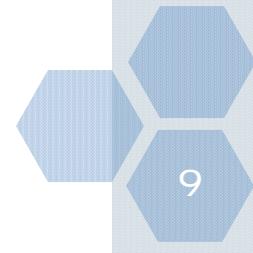
# 1. Giới thiệu về AI (tt)

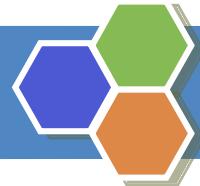
- ❖ Trí tuệ nhân tạo nhằm tạo ra “Máy người”?
- ❖ Mục tiêu
  - Xây dựng lý thuyết về thông minh hể giải thích các hoạt động thông minh
  - Tìm hiểu cơ chế sự thông minh của con người
  - Cơ chế lưu trữ tri thức
  - Cơ chế khai thác tri thức
  - Xây dựng cơ chế hiện thực sự thông minh
  - Áp dụng các hiểu biết này vào các máy móc phục vụ con người.



## 2. Một số i ặc điểm của AI

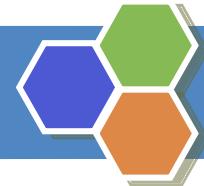
- ❖ Sử dụng máy tính vào suy luận trên các ký hiệu, nhận dạng qua mẫu, học, và các suy luận khác...
- ❖ Tập trung vào các vấn đề “khó” không thích hợp với các lời giải mang tính thuật toán.
- ❖ Quan tâm đến các kỹ thuật giải quyết vấn đề sử dụng các thông tin không chính xác, không đầy đủ, mơ hồ...
- ❖ Cho lời giải ‘đủ tốt’ chứ không phải là lời giải chính xác hay tối ưu.
- ❖ Sử dụng heuristics – “bí quyết”, “mẹo”
- ❖ Sử dụng tri thức chuyên môn
- ❖ ...



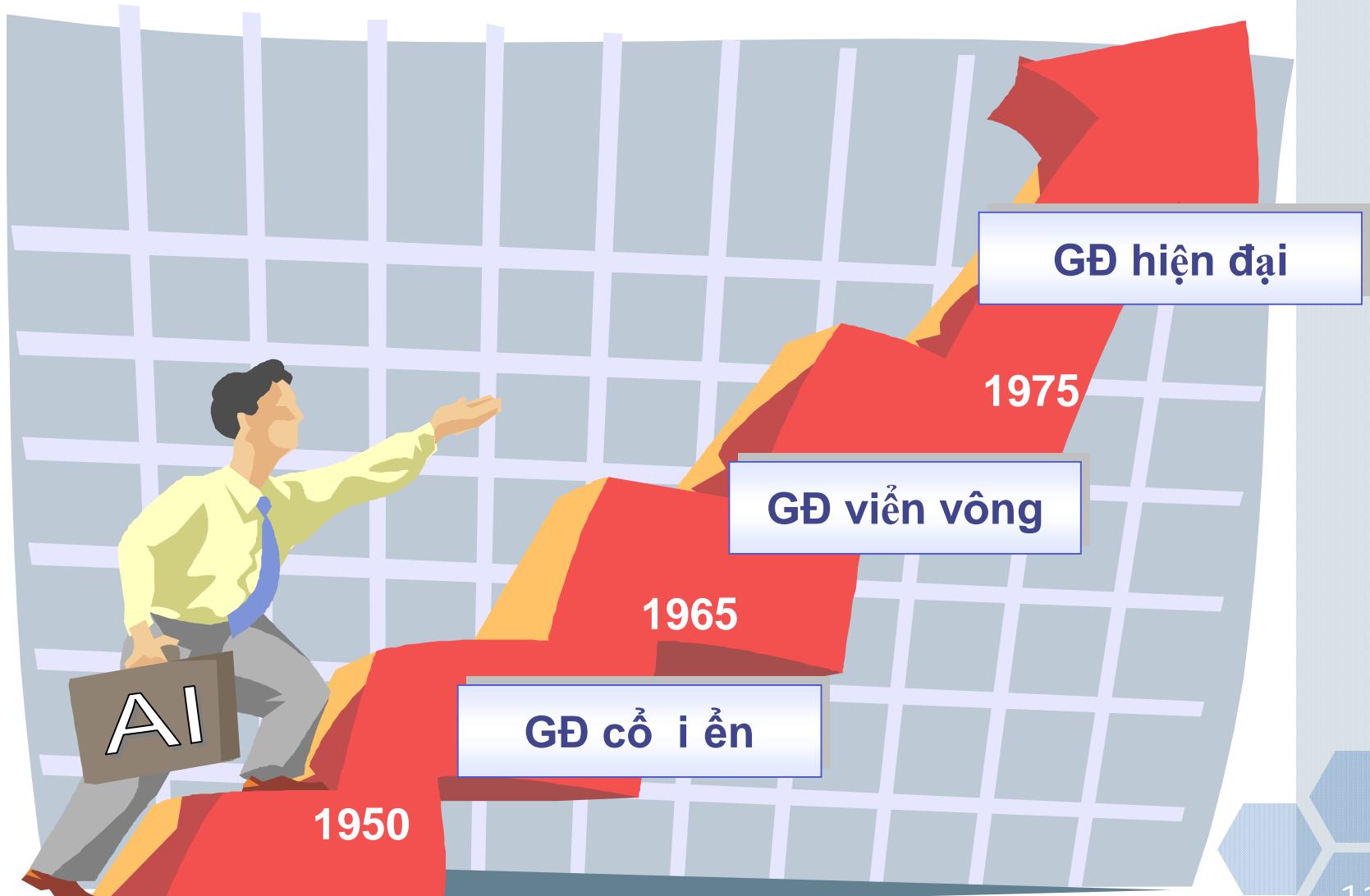


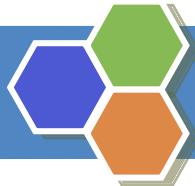
## 2. Một số nhược điểm của AI (tt)

- ❖ Chương trình chưa tự sinh ra được heuristic
- ❖ Chưa có khả năng xử lý song song của con người
- ❖ Chưa có khả năng diễn giải một vấn đề theo nhiều phương pháp khác nhau như con người.
- ❖ Chưa có khả năng xử lý thông tin trong môi trường liên tục như con người.
- ❖ Chưa có khả năng học như con người.
- ❖ Chưa có khả năng tự thích nghi với môi trường.



### 3. Lịch sử phát triển của AI





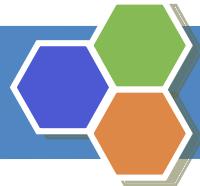
### 3. Lịch sử phát triển của AI (tt)

#### ❖ Giai đoạn cổ iễn (1950 – 1965)

Đây là giai đoạn của 2 lĩnh vực chính: Game Playing (Trò chơi) và Theorem Proving (Chứng minh định lý)

##### ➤ Game Playing

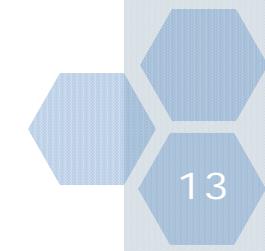
- ✓ Dựa trên kỹ thuật State Space Search với trạng thái (State) là các tình huống của trò chơi. Đáp án cần tìm là trạng thái thắng hay con đường dẫn tới trạng thái thắng. Áp dụng với các trò chơi loại đối kháng. Ví dụ: Trò chơi đánh cờ vua.
- ✓ Có 2 kỹ thuật tìm kiếm cơ bản
  - Kỹ thuật generate and test : chỉ tìm được 1 , áp án/ chưa chắc tối ưu.
  - Kỹ thuật Exhaustive search (vét cạn): Tìm tất cả các nghiệm, chọn lựa phương án tốt nhất.

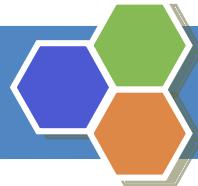


### 3. Lịch sử phát triển của AI (tt)

#### ➤ Theorem Proving

- ✓ Dựa trên tập tiên đề cho trước, chương trình sẽ thực hiện chuỗi các suy diễn để đạt tới biểu thức cần chứng minh.
- ✓ Nếu có nghĩa là ã ch ứng minh được. Ngược lại là không chứng minh được.
- ✓ Ví dụ: Chứng minh các định lý tự . ộng, giải toán,...
- ✓ Vẫn dựa trên kỹ thuật state space search nhưng khó khăn hơn do mức độ và quan hệ của các phép suy luận: song song, đồng thời, bắc cầu,..
- ✓ Có các kết quả khá tốt và vẫn còn phát triển đến ngày nay



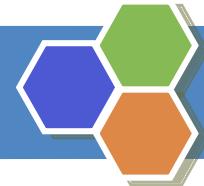


### 3. Lịch sử phát triển của AI (tt)

#### ❖ Giai đoạn viễn vông (1965 – 1975)

- Đây là giai đoạn phát triển với tham vọng làm cho máy hiểu được con người qua ngôn ngữ tự nhiên.
- Các công trình nghiên cứu tập trung vào việc biểu diễn tri thức và phương thức giao tiếp giữa người & máy bằng ngôn ngữ tự nhiên.
- Kết quả không mấy khả quan nhưng cũng tìm ra được các phương thức biểu diễn tri thức vẫn còn được dùng đến ngày nay tuy chưa thật tốt như:
  - Semantic Network (mạng ngữ nghĩa)
  - Conceptual graph (đồ th
  - Frame (khung)
  - Script (kịch bản)

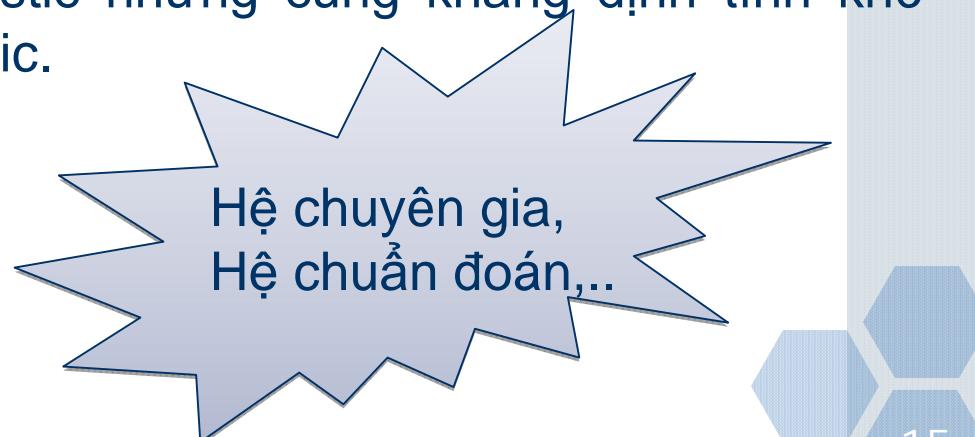


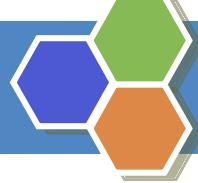


### 3. Lịch sử phát triển của AI (tt)

#### ❖ Giai đoạn hiện đại (từ 1975)

- Xác định lại mục tiêu mang tính thực tiễn hơn của AI là:
  - Tìm ra lời giải tốt nhất trong khoảng thời gian chấp nhận được.
  - Không cần toàn tìm ra lời giải tối ưu
- Tinh thần heuristic ra đời và được áp dụng mạnh mẽ để khắc phục bùng nổ tổ hợp.
- Khẳng định vai trò của tri thức đồng thời xác định 2 trở ngại lớn là biểu diễn tri thức và bùng nổ tổ hợp.
- Nêu cao vai trò của heuristic nhưng cũng khẳng định tính khó khăn trong đánh giá heuristic.





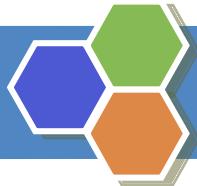
## 4. Các ứng dụng của AI

- ❖ **Game Playing:** Tìm kiếm / Heuristic
- ❖ **Automatic reasoning & Theorem proving:** Tìm kiếm / Heuristic
- ❖ **Expert System:** Là hướng phát triển mạnh mẽ nhất và có giá trị ứng dụng cao nhất
- ❖ **Planning & Robotic:** Các hệ thống dự báo, tự động hóa
- ❖ **Machine learning:** Trang bị khả năng học tập để giải quyết vấn đề kho tri thức:
  - **Supervised:** Kiểm soát được tri thức học được. Không tìm ra cái mới.
  - **UnSupervised:** Tự học, không kiểm soát. Có thể tạo ra tri thức mới nhưng cũng nguy hiểm vì có thể học những điều không mong muốn.



## 4. Các ứng dụng của AI (tt)

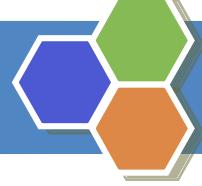
- ❖ **Natural Language Understanding & Semantic modelling:** Không được phát triển mạnh do mức độ phức tạp của bài toán cả về tri thức & khả năng suy luận.
- ❖ **Modeling Human performance:** Nghiên cứu cơ chế tổ chức trí tuệ của con người để áp dụng cho máy.
- ❖ **Language and Environment for AI:** Phát triển công cụ và môi trường để xây dựng các ứng dụng AI.
- ❖ **Neural network/Parallel distributed processing:** giải quyết vấn đề năng lực tính toán và tốc độ tính toán bằng kỹ thuật song song và mô phỏng mạng thần kinh của con người.



## 5. Thuật toán - thuật giải

- ❖ Bài toán/Vấn đề + Thuật toán = Kết quả???
- Có nhiều bài toán:
  - ✓ Chưa tìm ra thuật toán để giải
  - ✓ Không xác định được có hay không
- Một số bài toán có thuật giải nhưng không áp dụng được:
  - ✓ Độ phức tạp tính toán quá lớn
  - ✓ Các điều kiện đầu vào khó áp ứng
- Một số bài toán giải theo cách vi phạm thuật toán nhưng chấp nhận được.
- ❖ Thuật giải:

Cách giải chấp nhận được nhưng không hoàn toàn đáp ứng đầy đủ các yêu cầu của thuật toán.



# Bài tập Chương 1

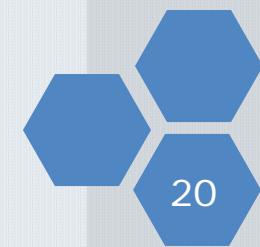
1. Phân biệt trí tuệ, trí thức và sự thông minh?
2. AI là gì? Các đặc điểm của AI?
3. Lịch sử phát triển của AI?
4. Các ứng dụng chính của AI?
5. Sự khác nhau giữa thuật toán & thuật giải?  
Lấy ví dụ?



# End of Chapter 1



Bùi Đức Dương  
*Khoa Công nghệ Thông tin*





# Bài giảng Trí tuệ nhân tạo

## Chương 2. Tìm kiếm trên không gian trạng thái

Bùi Đức Dương

Khoa Công nghệ Thông tin



# Nội dung

- 1. Không gian trạng thái**
- 2. Một số phương pháp tìm kiếm trên KGTT**



# 1. Không gian trạng thái

- ❖ Là một hệ thống gồm 4 thành phần [N,A,S,GD].

Trong đó:

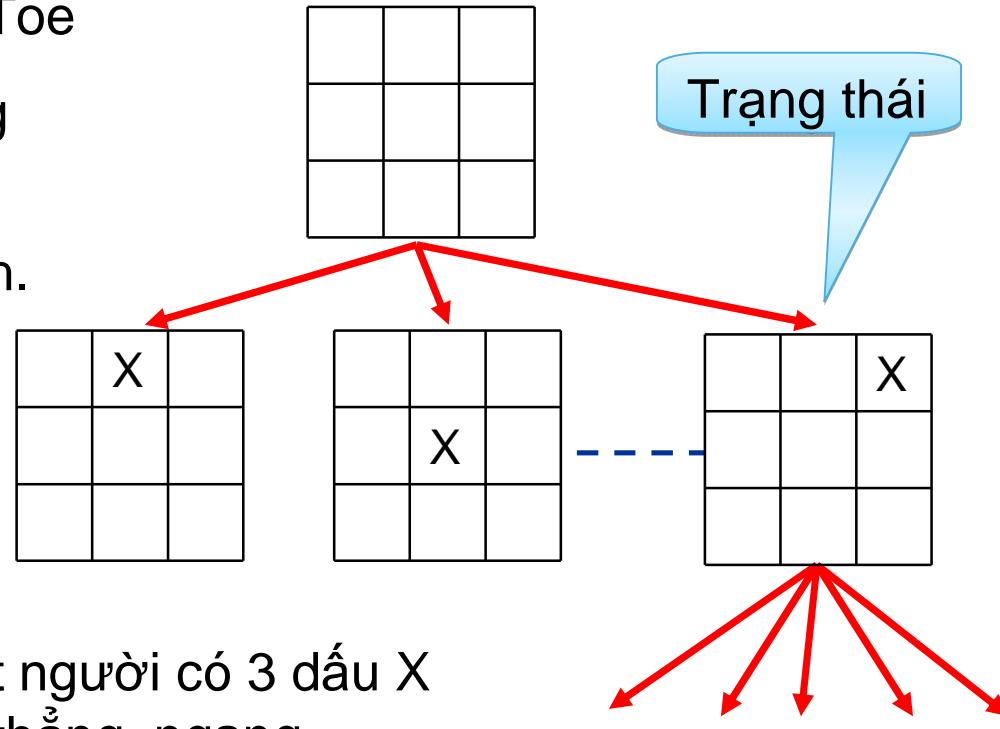
- N là tập nút của Graph. Mỗi nút là một trạng thái của quá trình giải quyết vấn đề
- A: Tập các cung nối giữa các nút N. Mỗi cung là một bước trong giải quyết vấn đề. Cung có thể có hướng
- S: Tập các trạng thái bắt đầu. S khác rỗng.
- GD: Tập các trạng thái đích. GD Không rỗng.
- ❖ Solution path: Là một path đi từ một nút bắt đầu  $S_i$  đến một nút kết thúc  $GD_j$ .
- ❖ Mục tiêu của các giải thuật tìm kiếm là tìm ra một solution path và/hay solution path tốt nhất.



# 1. Không gian trạng thái (tt)

## Ví dụ 1: Trò chơi Tic – Tac – Toe

- Trạng thái là một tình huống của bàn cờ
- Số trạng thái bùng nổ nhanh.
- Biểu diễn trạng thái
- Biểu diễn không gian



- Trạng thái kết thúc: có một người có 3 dấu X liên tục theo đường chéo, thẳng, ngang.
- Số trạng thái kết thúc=???

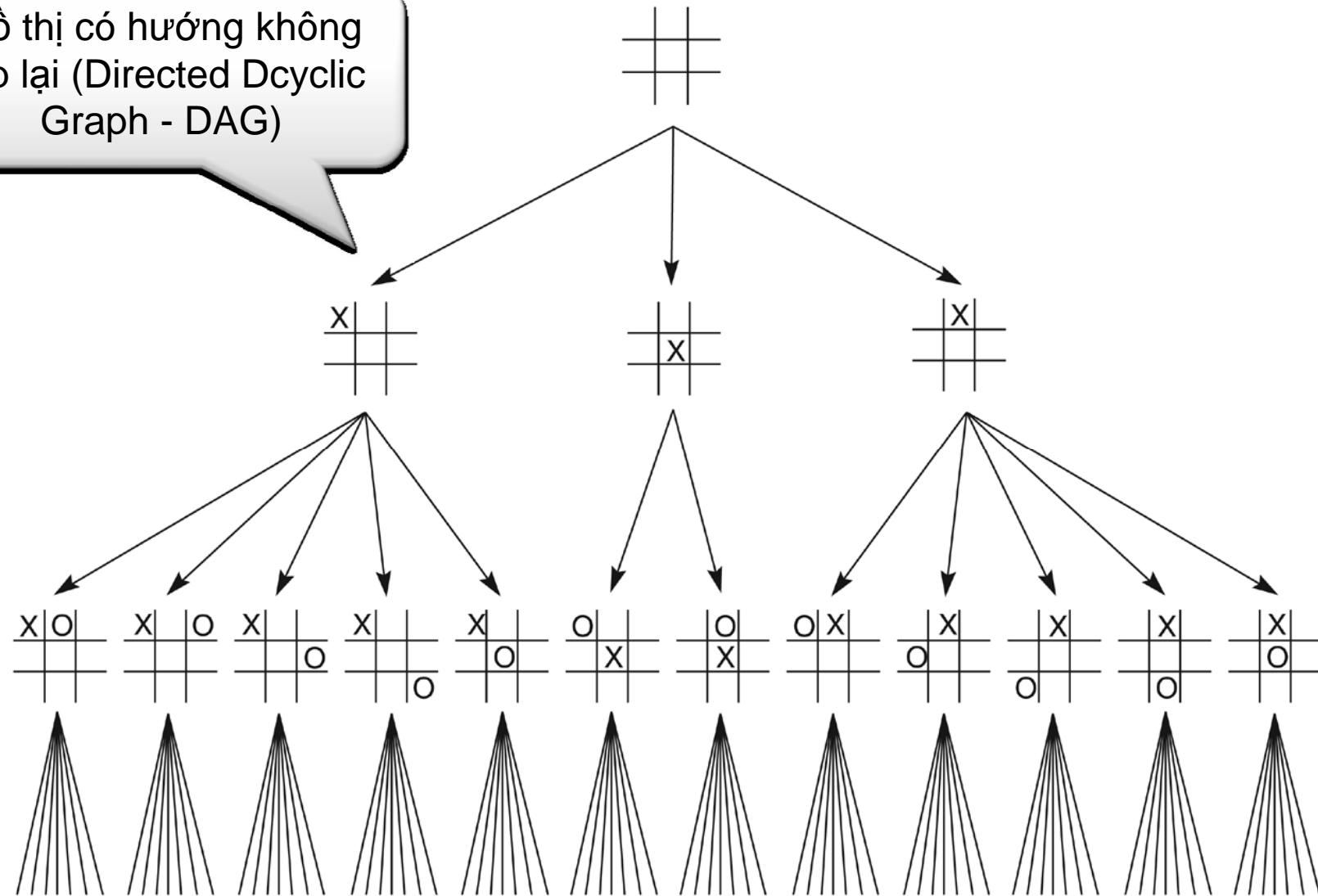
Trạng thái  
kết thúc

		X
	X	
X		



# 1. Không gian trạng thái (tt)

Đồ thị có hướng không  
lặp lại (Directed Acyclic  
Graph - DAG)





# 1. Không gian trạng thái (tt)

Ví dụ 2: Trò chơi (đố) Puzzle

❖ Trò đố 8 ô

Trạng thái đầu

	2	8
3	5	7
6	2	1

Trạng thái đích

1	2	3
8		4
7	6	5

❖ Trò đố 16 ô

Trạng thái đầu

11	14	4	7
10	6		5
1	2	13	15
9	12	8	3

Trạng thái đích

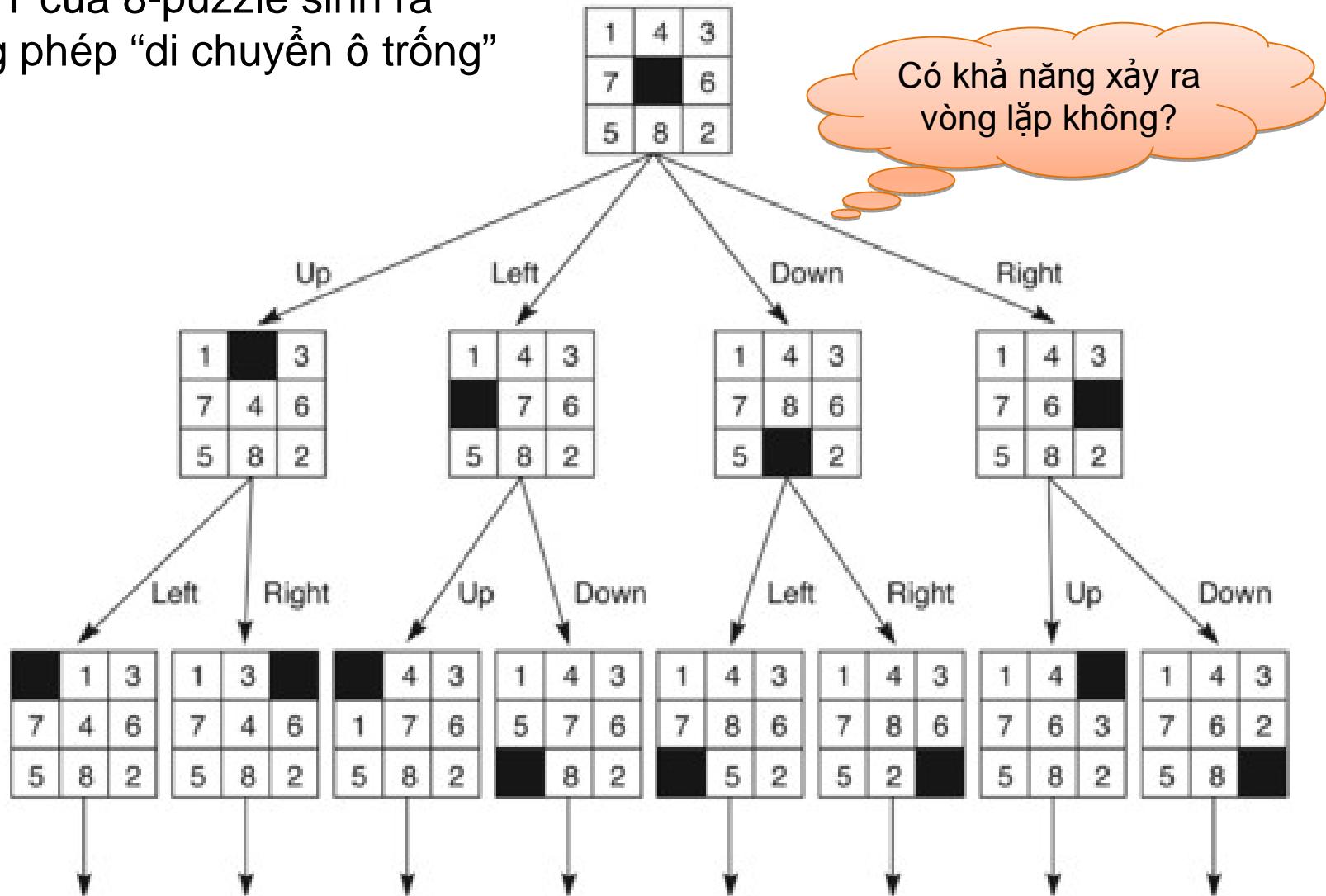
1	2	3	4
12	13	14	5
11		15	6
10	9	8	7

❖ Cần biểu diễn KGTT cho bài toán này như thế nào?



# 1. Không gian trạng thái (tt)

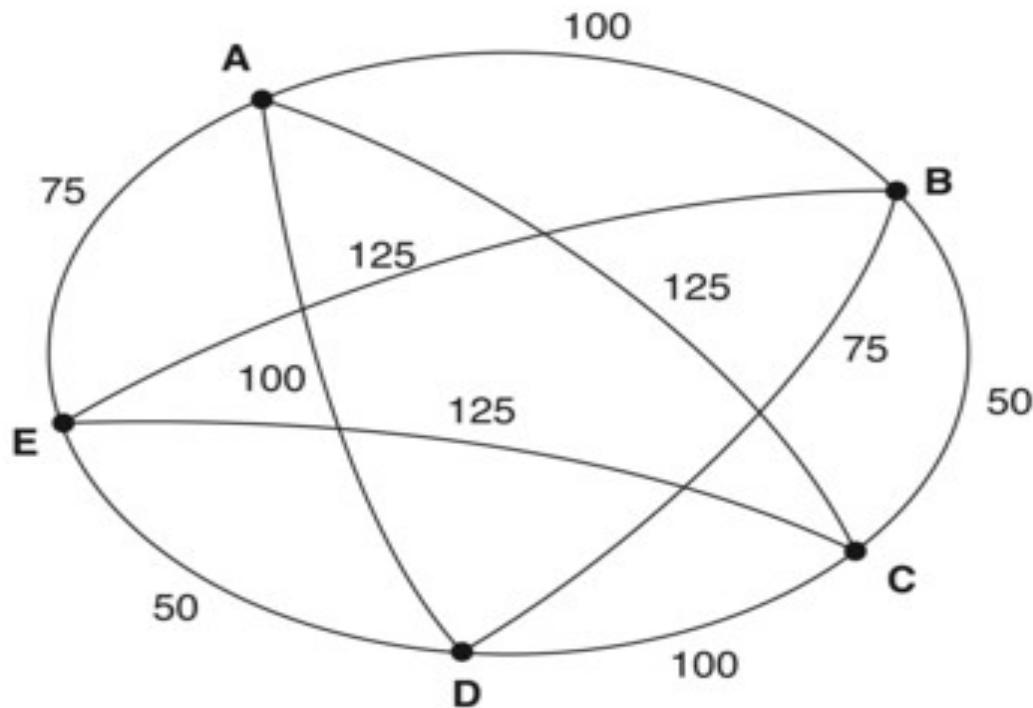
KGTT của 8-puzzle sinh ra  
bằng phép “di chuyển ô trống”





# 1. Không gian trạng thái (tt)

Ví dụ 3: Bài toán TSP

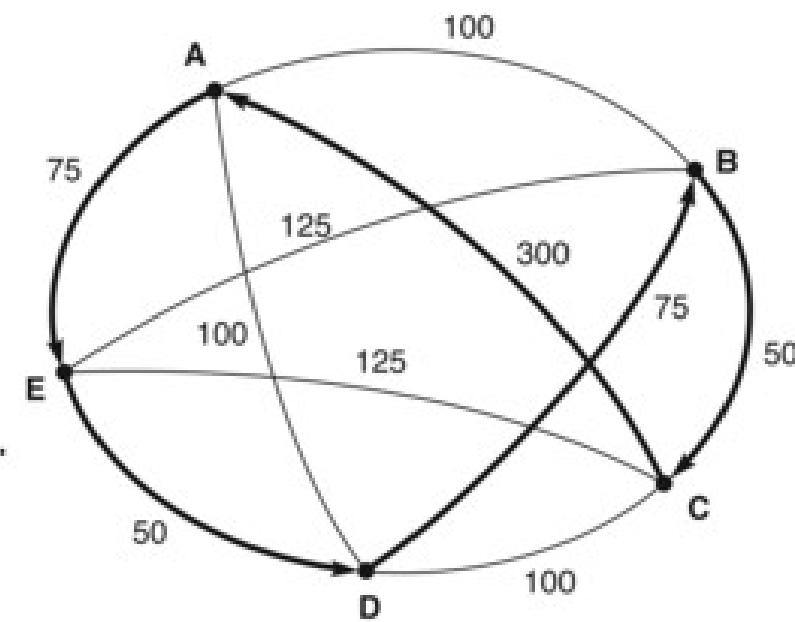
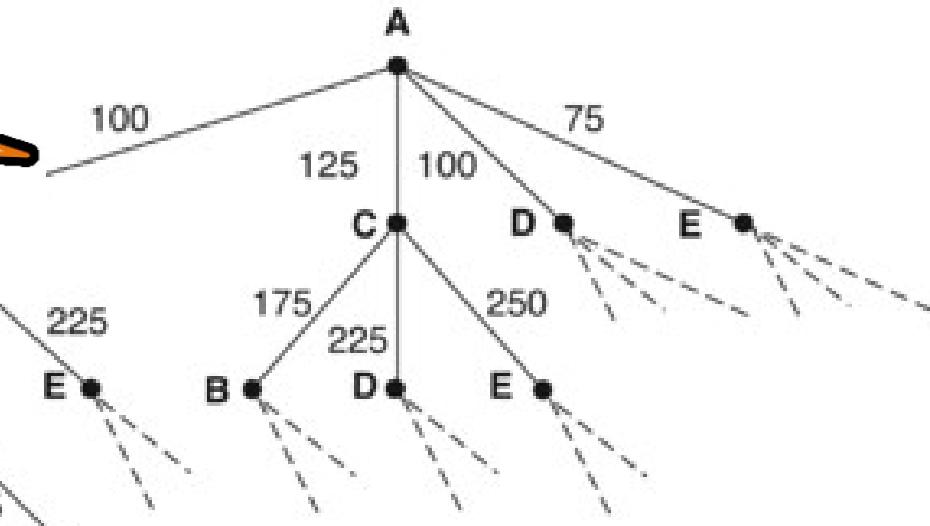
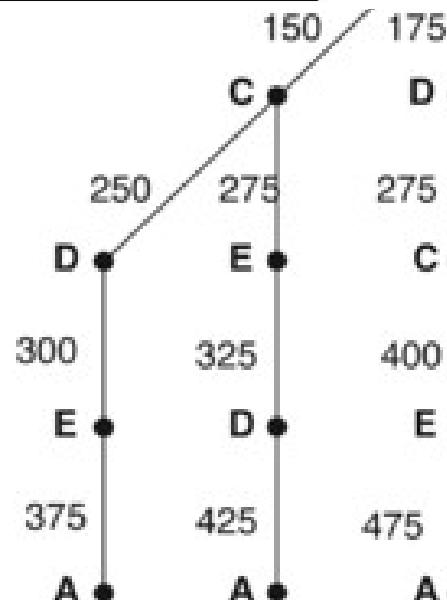


- ❖ Cần biểu diễn KGTT cho bài toán này như thế nào?



## 2. Không gian trạng thái (tt)

Mỗi cung được đánh dấu bằng tổng giá của con đường từ nút bắt đầu đến nút hiện tại.





## 2. Tìm kiếm trên không gian trạng thái

❖ TTNT = Biểu diễn + Tìm kiếm

➤ Sự biểu diễn phải:

- ✓ Cung cấp một cơ cấu tự nhiên để thể hiện tri thức/thông tin/ dữ liệu một cách đầy đủ (*Tính biểu đạt*).
- ✓ Hỗ trợ việc thực thi một cách hiệu quả việc tìm kiếm đáp án cho một vấn đề (*Tính hiệu quả*).

➤ Liệu việc tìm kiếm:

- ✓ Có kết thúc không?
- ✓ Có chắc chắn sẽ tìm được lời giải không?
- ✓ Có chắc chắn sẽ tìm được lời giải tối ưu không?



## 2. Tìm kiếm trên không gian trạng thái (tt)

❖ Các vấn đề khó khăn trong tìm kiếm với các bài toán AI

- Đặc tả vấn đề phức tạp
- Không gian tìm kiếm lớn
- Đặc tính đối tượng tìm kiếm thay đổi
- Đáp ứng thời gian thực
- Meta knowledge và kết quả “tối ưu”
- Khó khăn về kỹ thuật



## 2. Tìm kiếm trên không gian trạng thái (tt)

### ❖ State Space Search vs Database Search

#### ➤ Database

- ✓ Không gian tìm kiếm là một list hay tree
- ✓ Tìm kiếm một record/nút
- ✓ Phần tử eã duyệt qua là không còn dùng tới
- ✓ Không gian tìm kiếm là cỗ hịnh trong quá trình tìm kiếm
- ✓ Thuộc tính của một record/nút là cố rịnh

#### ➤ State Space

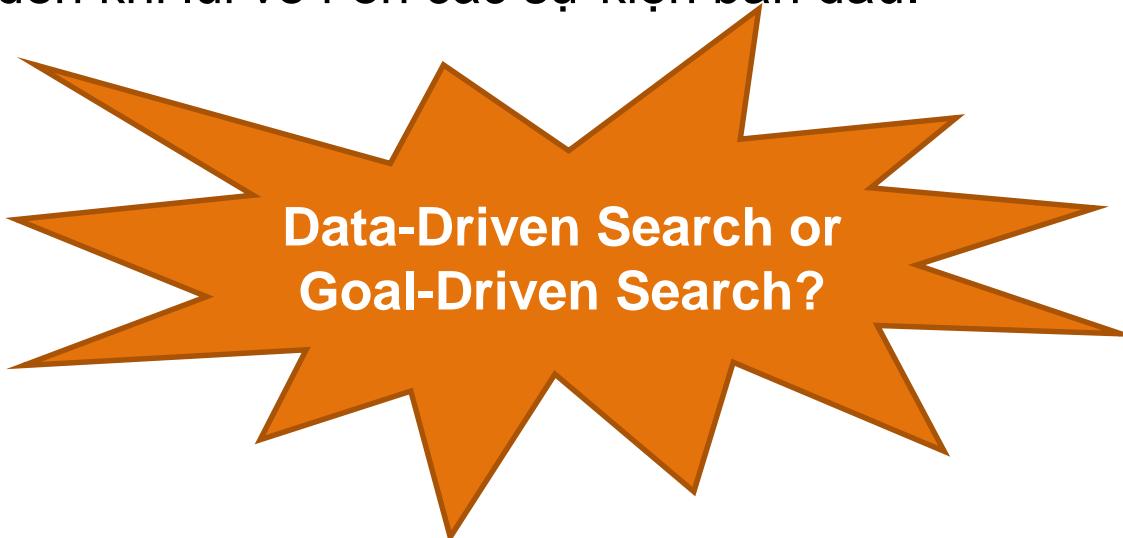
- ✓ Không gian tìm kiếm thường là một graph
- ✓ Mục tiêu tìm kiếm là một path
- ✓ Phải lưu trữ toàn bộ không gian trong quá trình tìm kiếm
- ✓ Không gian tìm kiếm biến động liên tục trong quá trình tìm kiếm
- ✓ Đặc tính của trạng thái/nút là phức tạp & biến động



## 2. Tìm kiếm trên không gian trạng thái (tt)

### ❖ Chiến lược tìm kiếm

- **Data-Driven-Search:** Quá trình search sẽ từ trạng thái hiện thời áp dụng các luật để gi hến trạng thái kế tiếp và cứ thế cho nến khi đạt được một goal.
- **Goal-Driven-Search:** Quá trình search sẽ gi từ trạng thái hiện tại (goal tạm thời) tìm xem luật nào có thể sinh ra trạng thái này. Các điều kiện để áp dụng được các luật đó trở thành subgoal. Quá trình lặp lại cho đền khi lui về r ến các sự kiện ban đầu.





## 2. Tìm kiếm trên không gian trạng thái (tt)

- ❖ Cả hai chiến lược cùng làm việc trên không gian trạng thái nhưng thứ tự và số các sự kiện duyệt qua khác nhau do cơ chế sinh ra các trạng thái khác nhau.
- ❖ Quyết định chọn lựa chiến lược tùy thuộc vào:
  - Độ phức tạp của các luật
  - Độ phân chia của không gian trạng thái
  - Sự hiện hữu của dữ liệu
    - ✓ Goal đã có hay chưa, nhiều hay ít
    - ✓ Goal được đặc tả như thế nào: state cụ thể hay mô tả mang tính đặc tính
  - Cơ sở thông tin Žể chọn lựa chiến lược hợp lý là một META KNOWLEDGE



## 2. Tìm kiếm trên không gian trạng thái (tt)

- Tìm kiếm đi từ dữ liệu đến mục tiêu thích hợp khi:
  - ✓ Tất cả hoặc một phần dữ liệu được cho từ đầu.
  - ✓ Có nhiều mục tiêu, nhưng chỉ có một số ít các phép toán có thể áp dụng cho một trạng thái bài toán.
  - ✓ Rất khó đưa ra một mục tiêu hoặc giả thuyết ngay lúc đầu.
- Tìm kiếm đi từ mục tiêu trở về dữ liệu thích hợp khi:
  - ✓ Có thể đưa ra mục tiêu hoặc giả thuyết ngay lúc đầu.
  - ✓ Có nhiều phép toán có thể áp dụng trên 1 trạng thái của bài toán ⇒ sự bùng nổ số lượng các trạng thái.
  - ✓ Các dữ liệu của bài toán không được cho trước, nhưng hệ thống phải đạt được trong quá trình tìm kiếm.



## 2. Tìm kiếm trên không gian trạng thái (tt)

### ❖ Tìm kiếm theo chiều rộng (BFS)

Procedure Breath\_First\_Search;

Begin

    Open:=[start]; Close:=[];

    While (Open <>[]) do

        begin

            remove X which is the leftmost of Open;

            If (X=goal) the return (Success)

            else begin

                generate children of X; Put X to Close;

                put remain children on RIGHT end of Open;

            end;

        end;

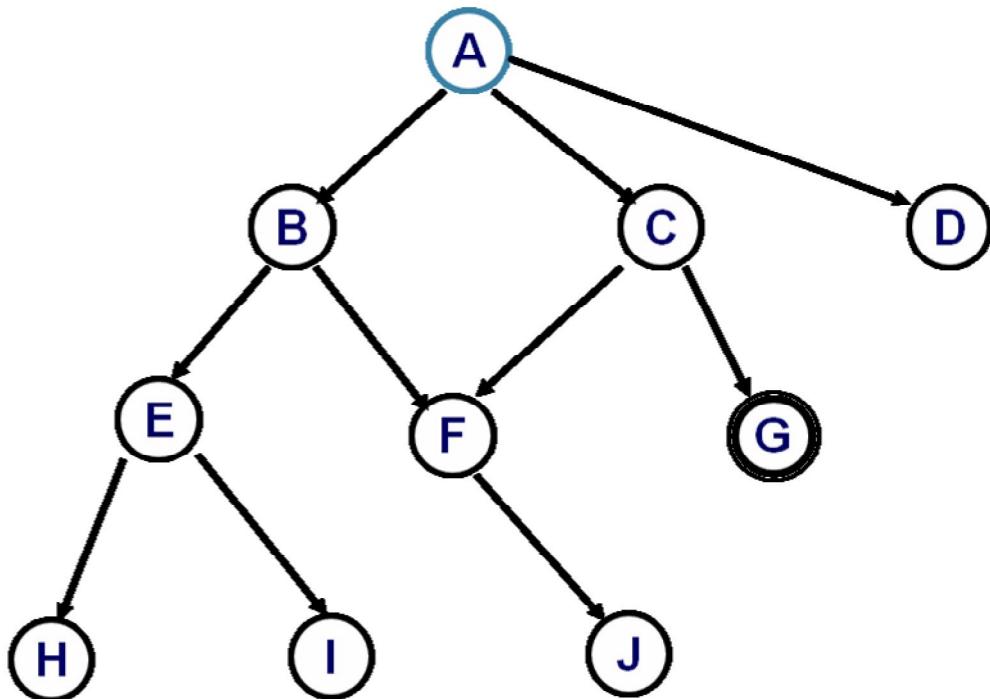
    Return (Fail);

End;



## 2. Tìm kiếm trên không gian trạng thái (tt)

### ❖ Tìm kiếm theo chiều rộng (BFS)



S=[A] bắt đầu

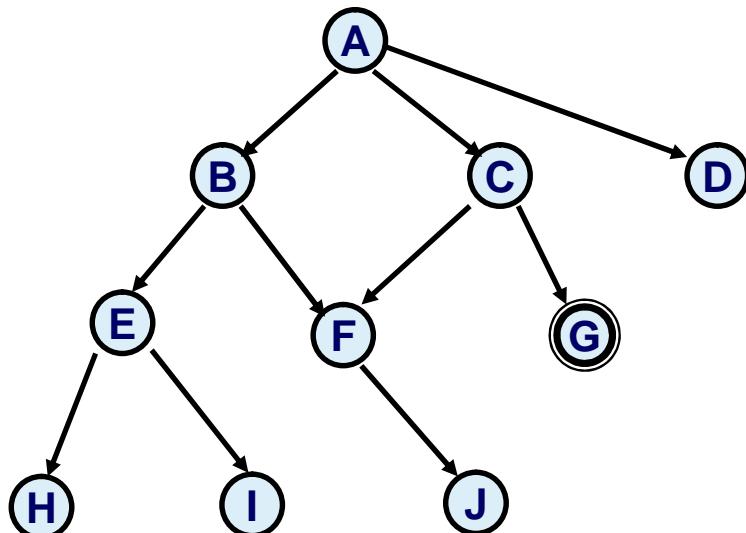
GD=[G] là goal. Kết thúc

Cung: Đường đi



## 2. Tìm kiếm trên không gian trạng thái (tt)

### ❖ Tìm kiếm theo chiều rộng (BFS)



Lần lặp	X	Open	Close
0		[A]	[]
1	A	[B C D ]	[A]
2	B	[C D E F ]	[A B]
3	C	[D E F G ]	[A B C]
4	D	[E F G ]	[A B C D]
5	E	[F G H I ]	[A B C D E]
6	F	[G H I J ]	[A B C D E F]
7	G	[H I J ]	[A B C D E F]



## 2. Tìm kiếm trên không gian trạng thái (tt)

### ❖ Tìm kiếm theo chiều sâu (DFS)

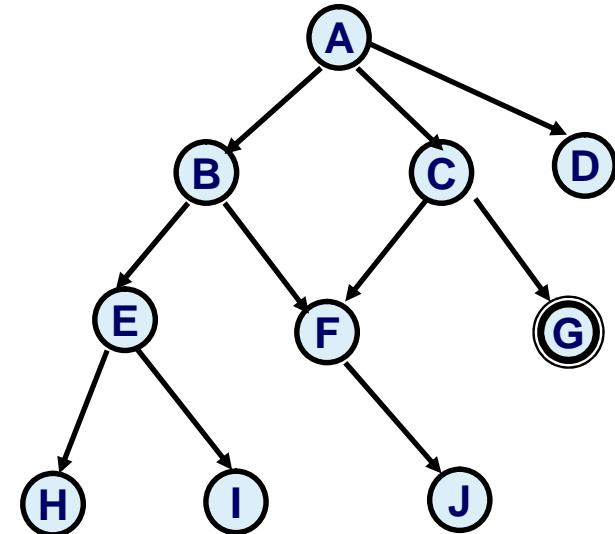
```
Procedure Depth_First_Search;  
Begin  
    Open := [start]; Close := [];  
    While (Open <> []) do  
        Begin  
            remove X which is the leftmost of Open;  
            If (X=goal) then return (Success)  
            else begin  
                generate children of X; Put X to close;  
                put remain children on LEFT end of open;  
            end;  
        End;  
    Return (Fail);  
End;
```



## 2. Tìm kiếm trên không gian trạng thái (tt)

### ❖ Tìm kiếm theo chiều sâu (DFS)

Lần lặp	X	Open	Close
0		[A]	[]
1	A	[B C D ]	[A]
2	B	[E F C D ]	[A B]
3	E	[H I F C D ]	[A B E ]
4	H	[I F C D ]	[A B E H ]
5	I	[F C D ]	[A B E H I ]
6	F	[J C D ]	[A B E H I F ]
7	J	[C D ]	[A B E H I F J ]
8	C	[G D ]	[A B E H I F J C ]
9	G		





## 2. Tìm kiếm trên không gian trạng thái (tt)

### ❖ BFS vs DFS

Tiêu chí	BFS	DFS
CTDL	Open:QUEUE	Open:STACK
Hiệu quả	Luôn tìm ra nghiệm có số cung nhỏ nhất	“Thường” cho kết quả nhanh hơn
Kết quả	Chắc chắn tìm ra kết quả nếu có	Có thể bị lặp vô tận
Bùng nổ tổ hợp	X	X





## 2. Tìm kiếm trên không gian trạng thái (tt)

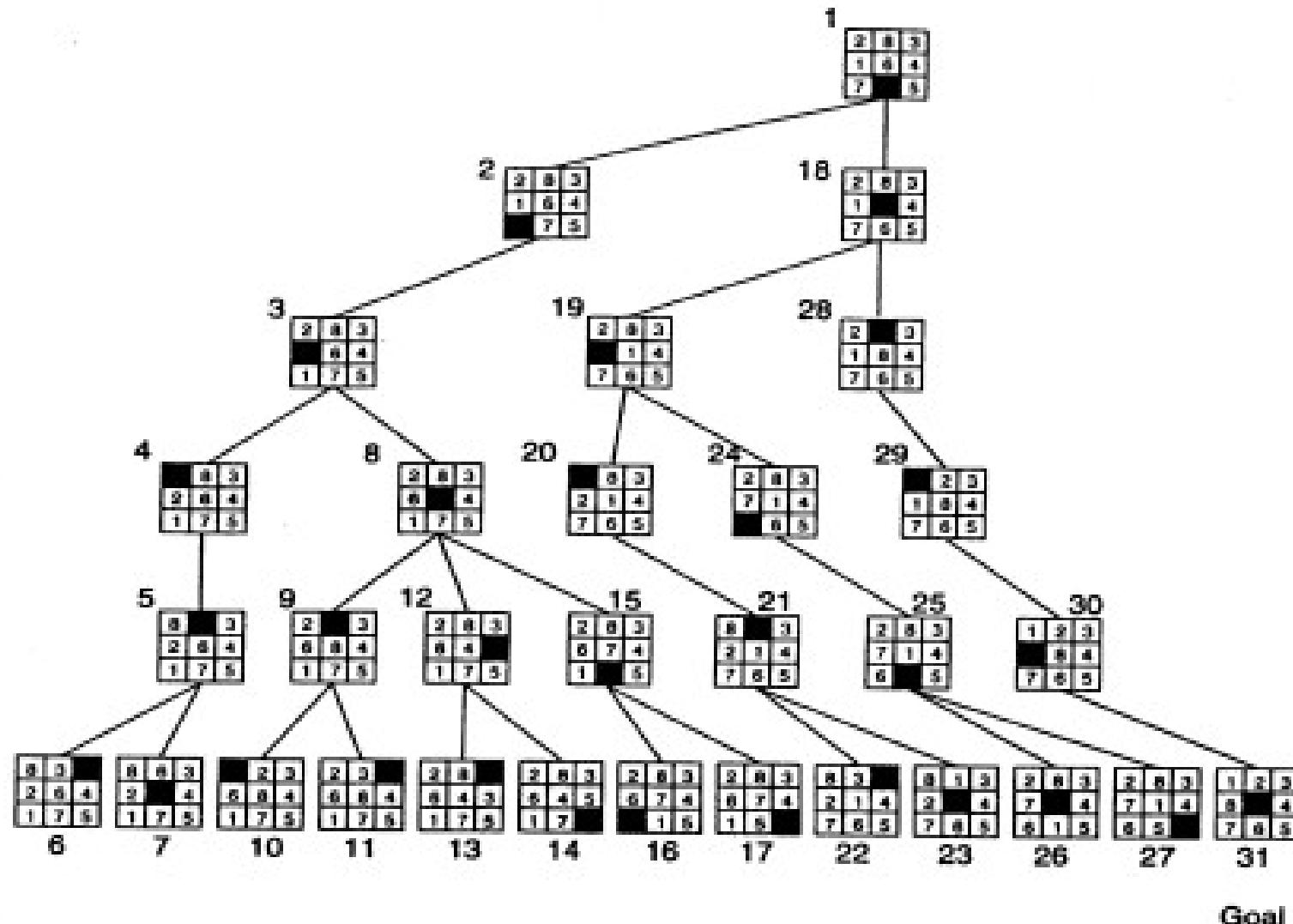
### ❖DFS có giới hạn (Depth Bound Search)

- Depth First Search có khả năng lặp vô tận do các trạng thái con sinh ra liên tục. Độ sâu tăng vô tận.
- Khắc phục bằng cách giới hạn độ sâu của giải thuật: quay lui khi trạng thái đang xét đạt đến độ sâu giới hạn đã định → Sâu bao nhiêu thì vừa?
- Chiến lược giới hạn
  - ✓ Cố định một độ sâu MAX, như các danh thủ chơi cờ tính trước được số nước nhất định
  - ✓ Theo cấu hình resource của máy tính
  - ✓ Meta knowledge trong việc định giới hạn độ sâu.
  - ✓ Giới hạn độ sâu → co hẹp không gian trạng thái → có thể mất nghiệm.



## 2. Tìm kiếm trên không gian trạng thái (tt)

DFS có giới hạn (Depth Bound =5) trong trò chơi 8-puzzle

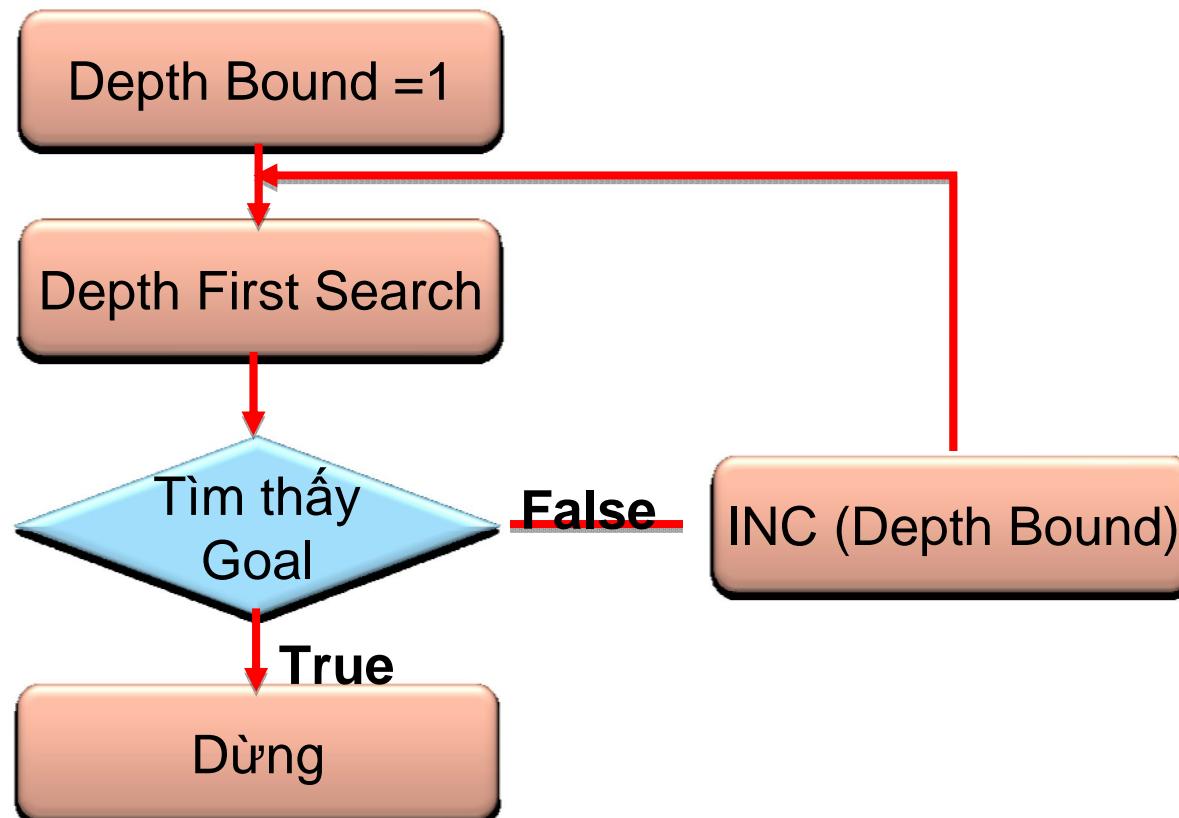


Goal



## 2. Tìm kiếm trên không gian trạng thái (tt)

- ❖ DFS đào sâu nhiều lần (Depth-first Iterative Deepening Search)



- ❖ Độ phức tạp về thời gian cùng bậc với BFS và DFS.



## 2. Tìm kiếm trên không gian trạng thái (tt)

### ❖ Backtrack Search

- Graph Search (GS) phải có khả năng tìm kiếm ra tất cả các Path có thể có hể tìm được nghiệm: Path từ trạng thái khởi đầu đến Goal.
- GS thực hiện bằng cách “lần” theo các nhánh của Graph. Từ một trạng thái, sinh ra các trạng thái con, chọn một trạng thái con, xem , ó là trạng thái xét kế tiếp. Lặp lại cho đến khi tìm thấy một trạng thái đích.
- “Lần” theo các trạng thái → Vào ngõ cụt?
- Khi gặp nhánh không đi tiếp được, giải thuật phải có khả năng quay lui lại trạng thái trước đó hể ni sang nhánh khác: Back Tracking.



## 2. Tìm kiếm trên không gian trạng thái (tt)

### ➤ Một số ký hiệu

- ✓ **SL** (State list) : chứa danh sách các trạng thái trên path hiện đang xét. Nếu tìm ra goal thì SL chính là nghiệm.
- ✓ **NSL** (New State List): chứa danh sách các trạng thái đang đợi xét.
- ✓ **DE** (Dead End): chứa các trạng thái mà con cháu của chúng không chứa đích.
- ✓ **CS** (Current State): chứa trạng thái đang xét.
- ✓ Hướng phát triển của quá trình search tùy theo cơ cấu tổ chức của NSL: FIFO, FILO hay Evaluated.



## 2. Tìm kiếm trên không gian trạng thái (tt)

```
Procedure Backtrack_Search;  
BEGIN  
    S:=[start]; NLS:=[start];  DE:=[ ]; CS:=start;  
    While (NSL<>[ ]) do  
        Begin  
            If (CS = Goal) then return(SL);  
            If (CS has no children (Except node in [DE, SI, NSL]) )then  
                Begin  
                    While ((SL<>[ ]) and CS=First Element of SL)) do  
                        Begin  
                            Add CS to DE  
                            Remove first element from SL;  
                            Remove first element from NSL;  
                            Cs:= first element of NSL;  
                        End;  
                        Add CS to SL;  
                    End;  
                End;  
            End;
```



## 2. Tìm kiếm trên không gian trạng thái (tt)

Else

Begin

Add children of CS (Except node in DE,SL and NSL) to NSL

CS:= first element of NSL;

Add CS to SL;

End;

End;

End; {end while}

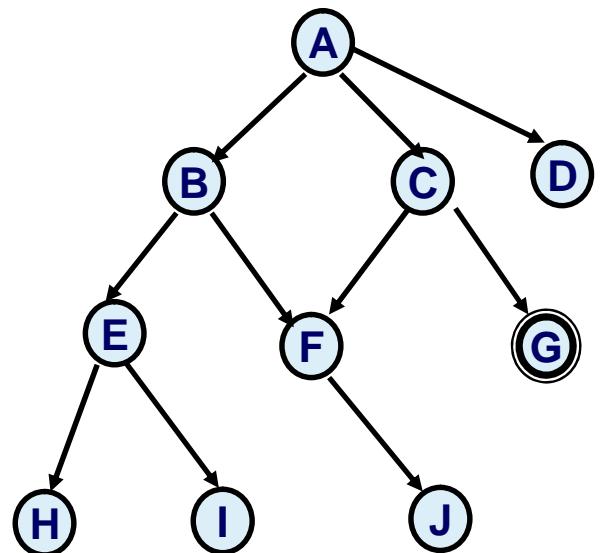
Return Fail;

END.



## 2. Tìm kiếm trên không gian trạng thái (tt)

Backtrack Search - Ví dụ

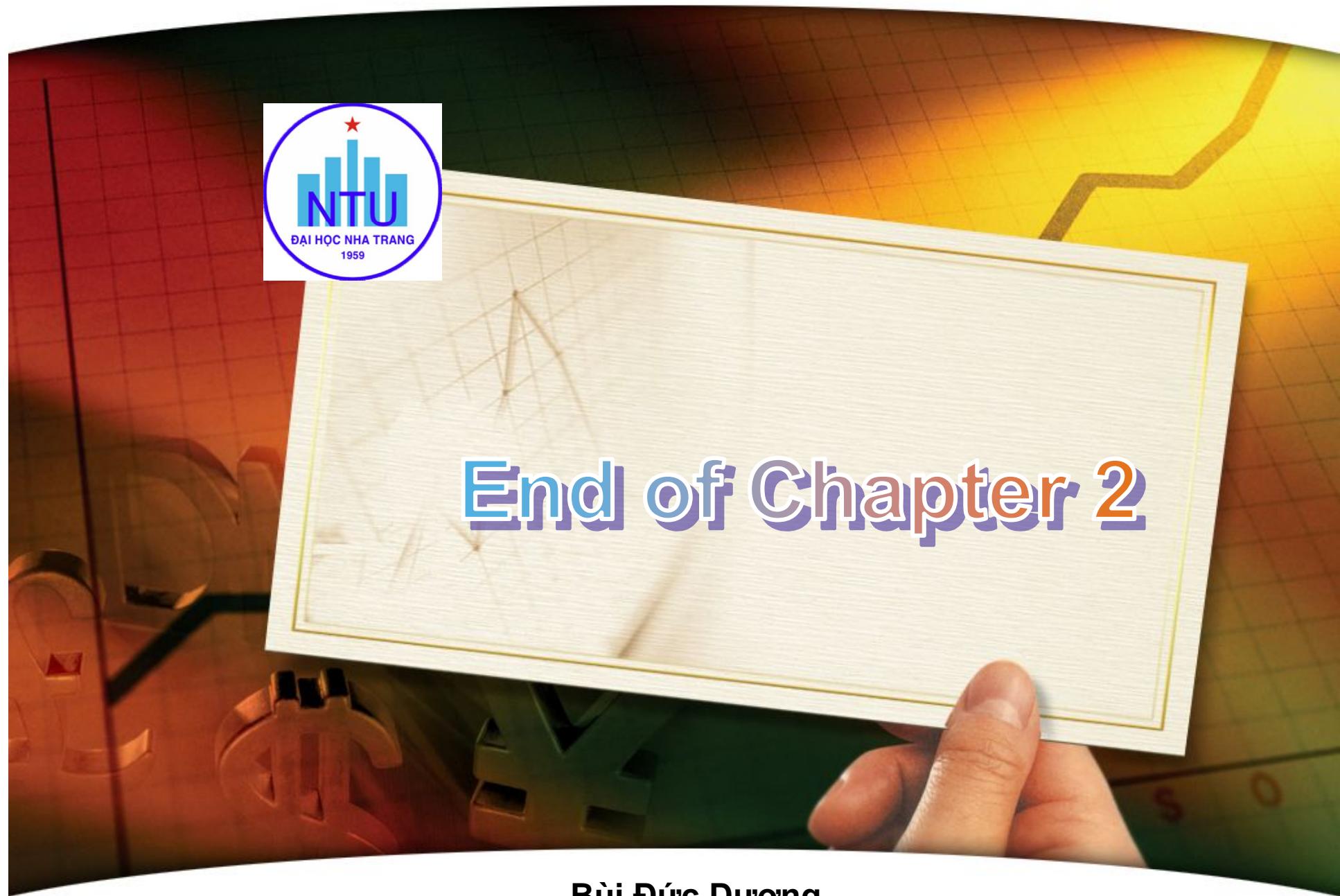


Lần lặp	CS	SL	NSL	DE
0	A	[A]	[A]	[]
1	B	[B A]	[B C D A]	[]
2	E	[E B A]	[E F B C D A]	[]
3	H	[H E B A]	[H I E F B C D A]	[]
4	I	[I E B A]	[I E F B C D A]	[H]
5	F	[F B A]	[F B C D A]	[E I H]
6	J	[J F B A]	[J F B C D A]	[E I H]
7	C	[C A]	[C D A]	[B F J E I H]
8	G	<b>[G C A]</b>	[G C D A]	[B F J E I H]



# Bài tập Chương 2

1. Biểu diễn không gian trạng thái cho các bài toán đơn giản.
2. Lấy ví dụ và minh họa các thuật toán BFS, DFS, DFIDS, Backtrack Search.
3. Cài đặt các thuật toán BFS, DFS, DFIDS, Backtrack Search.



Bùi Đức Dương  
*Khoa Công nghệ Thông tin*



# Bài giảng Trí tuệ nhân tạo

## Chương 3. Tìm kiếm Heuristic

*Bùi Đức Dương – Khoa Công nghệ Thông tin*

# 1. Giới thiệu

## ❖ Heuristic là gì?

- Heuristic là những tri thức được rút tinh túng từ những kinh nghiệm, “trực giác” của con người (mẹo).
- Heuristic có thể là những tri thức “đúng” hay “sai”.
- Heuristic là những meta knowledge và “thường đúng”.

## ❖ Heuristic dùng để làm gì?

- Trong những bài toán tìm kiếm trên không gian trạng thái, có 2 trường hợp cần đến heuristic:
  - ✓ Vấn đề có thể không có nghiệm chính xác do các mệnh đề không phát biểu chặt chẽ hay thiếu dữ liệu để khẳng định kết quả.
  - ✓ Vấn đề có nghiệm chính xác nhưng phí tổn tính toán để tìm ra nghiệm là quá lớn (hệ quả của bùng nổ tổ hợp)
- Heuristic giúp tìm kiếm đạt kết quả với chi phí thấp hơn

## 1. Giới thiệu (tt)

❖ Giải thuật heuristic thường gồm 2 phần:

➤ Phép đo heuristic:

Thể hiện qua *hàm đánh giá heuristic (evaluation function)*,  
dùng để ánh giá các đặc điểm của một trạng thái trong KGTT.

➤ Giải thuật tìm kiếm heuristic:

- ✓ *Tìm kiếm tốt nhất (best-first search)*
- ✓ *Giải thuật leo đồi (hill-climbing)*

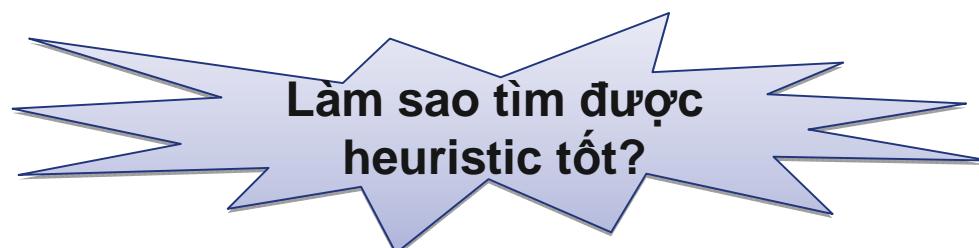
## 1. Giới thiệu (tt)

### ❖ Heuristic dùng như thế nào trong SSS?

- Tìm kiếm trên không gian trạng thái theo chiều nào? BFS hay DFS?
- Tìm theo heuristic: heuristic Định hướng quá trình tìm kiếm theo hướng mà “nó” cho rằng khả năng đạt tới nghiệm là cao nhất. Không “sâu” cũng không “rộng”

### ❖ Kết quả của tìm kiếm với heuristic

- Việc tìm kiếm theo định hướng của heuristic có kết quả tốt hay xấu tùy theo heuristic “đúng” hay “sai”.
- Heuristic có khả năng bỏ sót nghiệm
- Heuristic càng tốt càng dẫn đến kết quả nhanh và tốt.



## 1. Giới thiệu (tt)

- ❖ Thuật giải heuristic là sự mở rộng của khái niệm thuật toán. Nó thể hiện cách giải bài toán với các đặc tính sau:
  - Thường tìm được lời giải tốt (nhưng không chắc là tốt nhất)
  - Giải bài toán theo thuật giải heuristic thường dễ dàng và nhanh chóng đưa ra kết quả hơn so với giải thuật tối ưu (chi phí sẽ thấp hơn).
  - Thuật giải heuristic thường thể hiện khá tự nhiên, gần với cách suy nghĩ và hành động của con người.

# 1. Giới thiệu (tt)

- ❖ Một số nguyên lý cơ bản
- **Nguyên lý vét cạn thông minh:** Giới hạn không gian tìm kiếm dựa vào đặc thù bài toán
- **Nguyên lý tham lam (Greedy):** Lấy tiêu chuẩn tối ưu của bài toán làm tiêu chuẩn cho việc lựa chọn ở từng bước giải
- **Nguyên lý thứ tự:** Hành động dựa trên một cấu trúc thứ tự hợp lý của không gian khảo sát
- **Hàm heuristic:** Hàm đánh giá dựa “kinh nghiệm”, phụ thuộc vào trạng thái hiện tại của mỗi bước giải. Từ hay chọn ra phương án hành động.

# 1. Giới thiệu (tt)

## ❖ Một số ví dụ

- **Bài toán 1 (*Đổi tiền*)**. Đổi số tiền T thành các loại tiền có mệnh giá  $L_1, L_2, \dots, L_n$  ( $L_1 > L_2 > \dots > L_n$ ) sao cho số tờ là ít nhất.

Procedure DoiTien;

Begin

i=1;

Repeat

Soto[i]=T/L[i];

T=T%L[i];

i++;

Until (T=0);

End;

- **Bài toán 2 (*Tìm nghiệm*)**. Tìm tất cả các nghiệm không âm của phương trình:

$$x_1 + x_2 + \dots + x_n = M$$

# 1. Giới thiệu (tt)

- **Bài toán 3 (*Hành trình ngắn nhất -TSP*).** Cho  $n$  thành phố (1, 2,... $n$ ) và khoảng cách giữa chúng ( $c_{i,j}$ ). Hãy tìm hành trình của một người đưa thư, đi qua tất cả các thành phố rồi quay về thành phố xuất phát, sao cho tổng chiều dài đường đi là ngắn nhất.
  - ✓ Vết cạn:  $(n-1)!$ . Với  $n$  lớn?
  - ✓ Greedy 1: Mỗi bước chọn  $i \rightarrow j$  sao cho  $j$  gần  $i$  nhất trong những thành phố chưa đến nối với  $i$ .
  - ✓ Greedy 2: Mỗi bước chọn  $i \rightarrow j$  sao cho  $i$  gần  $j$  nhất trong những thành phố chưa đến nối với  $j$ .

# 1. Giới thiệu (tt)

➤ Ví dụ: TSP với n=7 cho bởi ma trận sau

	A	B	C	D	E	F	G
A	0	150	178	350	215	655	472
B	150	0	550	730	196	285	328
C	178	550	0	125	485	348	226
D	350	730	125	0	925	147	630
E	215	196	485	925	0	352	475
F	655	285	348	147	352	0	548
G	472	328	226	630	475	548	0

✓ Greedy 1: A → B → E → F → D → C → G → A

✓ Greedy 2: A → B → E → F → D → C → G → A

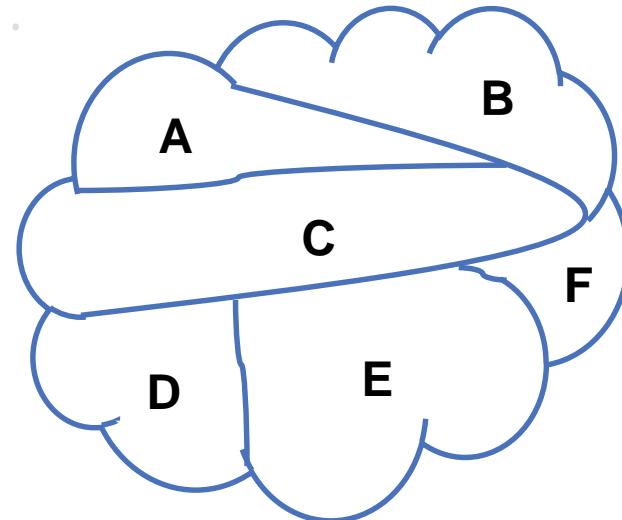
# 1. Giới thiệu (tt)

➤ **Bài toán 4 (*Tô màu bản đồ*).** Cho  $n$  quốc gia ( $1, 2, \dots, n$ ). Hai quốc gia được gọi là láng giềng nếu có đường biên giới chung. Hãy tìm số màu ít nhất để tô cho các quốc gia sao cho 2 quốc gia láng giềng bất kỳ phải khác màu.

- ✓ Bậc của quốc gia: số QG láng giềng

QG	A	B	C	D	E	F
Bậc	2	3	5	2	3	3

- ✓ Đưa về bài toán tô màu đồ thị



# 1. Giới thiệu (tt)

## ❖ Thuật toán tô màu đồ thị

**Input:** Ô thị  $G = (V, E)$ .

**Output:** hò thị  $G = (V, E)$  có các đỉnh đã được gán màu.

### Method:

➤ **B1:** Sắp xếp các đỉnh của đồ thị  $E'=[v_1, v_2, \dots, v_n]$  theo thứ tự bậc giảm dần:  $d(v_1) \geq d(v_2) \geq \dots \geq d(v_n)$

Đặt  $i := 1$ ;

➤ **B2:** Tô màu  $i$  cho đỉnh đầu tiên. Duyệt lần lượt các đỉnh tiếp theo và tô màu  $i$  cho đỉnh không kề đỉnh đã được tô màu  $i$ .

➤ **B3:** Nếu tất cả các đỉnh đã được tô màu thì kết thúc, ô thị được tô bằng  $i$  màu. Ngược lại, sang bước 4.

➤ **B4:** Loại khỏi  $E'$  các đỉnh đã tô màu. Sắp xếp lại các đỉnh trong  $E'$  theo thứ tự bậc giảm dần. Đặt  $i := i + 1$  và quay lại **B2**.

# 1. Giới thiệu (tt)

## Áp dụng thuật toán tô màu giải bài toán xếp lịch thi đấu thể thao:

1. Có 6 đội tham gia: A, B, C, D, E và F. Các trận đấu diễn ra theo như bảng dưới đây:

	A	B	C	D	E	F
A		AB	AC	AD	AE	AF
B			BC	BD	BE	BF
C				CD	CE	CF
D					DE	DF
E						EF
F						

Các trận đấu đã xảy ra: AB, BC, CD, DE, EF, và AF

Hãy sắp xếp các trận đấu còn lại sao cho số buổi tổ chức là ít nhất.  
Lưu ý mỗi đội không thể tham gia thi đấu cùng một lúc 2 trận; Mỗi buổi có thể diễn ra nhiều trận đấu.

# 1. Giới thiệu (tt)

## Áp dụng thuật toán tô màu giải bài toán xếp lịch thi đấu thể thao:

2. Có 6 đội tham gia: A, B, C, D, E và F. Các trận đấu diễn ra theo như bảng dưới đây:

	A	B	C	D	E	F
A		AB	AC	AD	AE	AF
B			BC	BD	BE	BF
C				CD	CE	CF
D					DE	DF
E						EF
F						

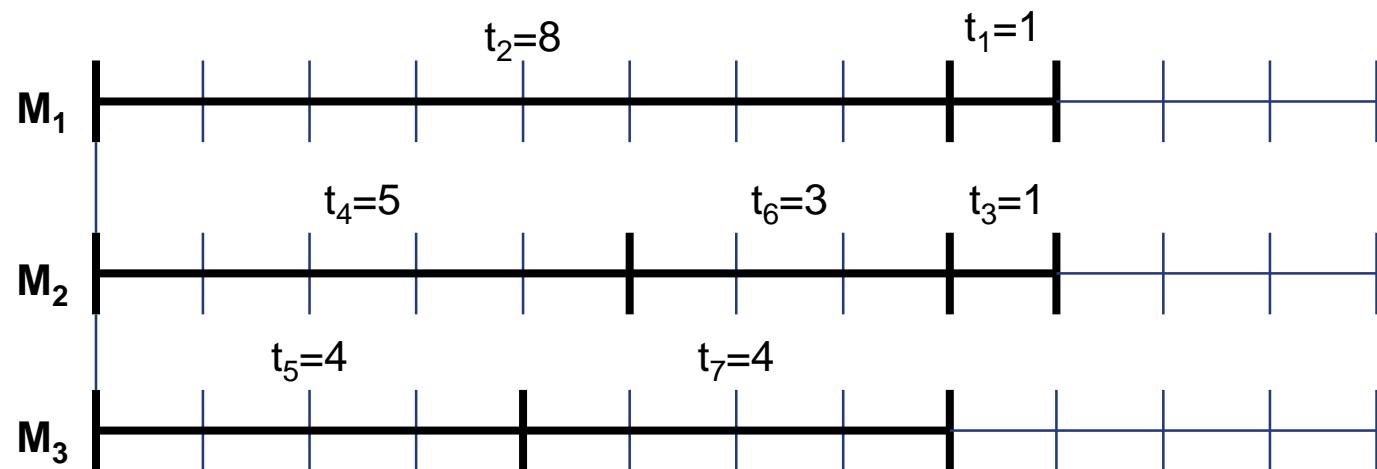
Các trận đấu đã xảy ra: AB, AC, CD, DE, DF, và AF

Hãy sắp xếp các trận đấu còn lại sao cho số buổi tổ chức là ít nhất.  
Lưu ý mỗi đội không thể tham gia thi đấu cùng một lúc 2 trận; Mỗi buổi có thể diễn ra nhiều trận đấu.

# 1. Giới thiệu (tt)

➤ **Bài toán 5 (Bài toán phân việc).** Phân xưởng có n máy cưa công suất bằng nhau  $M_1, M_2, \dots, M_n$ . Có m thanh gỗ cần được xẻ  $G_1, G_2, \dots, G_m$ , biết thời gian xẻ của thanh gỗ  $G_i$  là  $t_i$ . Hãy tìm phương án để xẻ m thanh gỗ trong thời gian sớm nhất.

- ✓ Sắp xếp các thanh gỗ theo chiều giảm dần của thời gian xẻ.
- ✓ Lần lượt bố trí các thanh gỗ vào máy còn dư nhiều thời gian nhất.
- ✓ Ví dụ: Xét trường hợp 3 máy  $M_1, M_2, M_3$  và 7 thanh gỗ có thời gian xẻ  $t_1=1; t_2=8; t_3=1; t_4=5; t_5=4; t_6=3; t_7=4;$



## 2. Thuật toán HCS

- ❖ Tìm kiếm leo đồi – Hill Climbing Search (Pearl, 1984)
  - Mở rộng trạng thái hiện tại và ánh giá các trạng thái con của nó bằng hàm đánh giá heuristic.
    - Con “tốt” sẽ được chọn để tiếp. Chọn một trạng thái tốt hơn: leo đồi đơn giản, trạng thái tốt nhất: leo đồi dốc đứng.
  - Khác với tìm kiếm sâu, leo đồi không lưu tất cả các con mà chỉ lưu đúng một t.thái được chọn nếu có.
- ❖ Giới hạn
  - Giải thuật có khuynh hướng bị sa lầy ở những cực đại cục bộ:
    - ✓ Lời giải tìm được không tối ưu
    - ✓ Không tìm được lời giải mặc dù có tồn tại lời giải
  - Giải thuật có thể gấp vòng lặp vô hạn do không lưu giữ thông tin về các trạng thái đã duyệt.

## 2. Thuật toán HCS (tt)



**Procedure Hill\_Climbing\_Search;**

Begin

Open:={Start}; Close:=∅;

While (Open<> ∅) do

Begin

X= Get(Open,leftmost);// Chọn trạng thái ngoài cùng của Open

If (X=Goal) then Return True

Else Begin

Close = Close  $\cup$  {X};

Sinh ra các nút con của X;

C=Children(X)\{Open  $\cup$  Close}

Y = Retrieve(C); // Chọn trạng thái tốt hơn X(nhất)

Open = Open  $\cup$  {Y};

End;

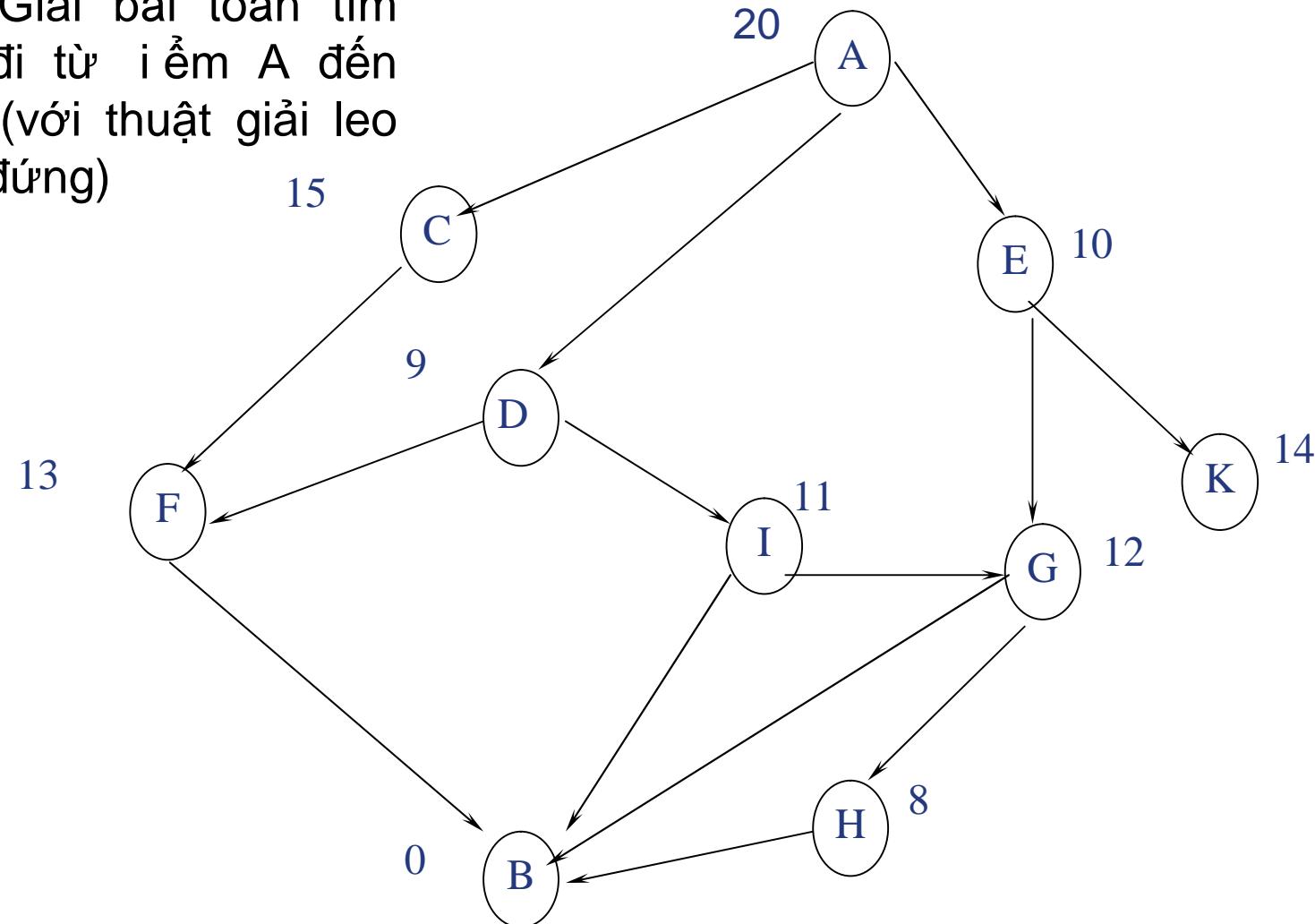
End;

Return False;

End;

## 2. Thuật toán HCS (tt)

Ví dụ: Giải bài toán tìm đường đi từ điểm A đến điểm B (với thuật giải leo đồi dốc đứng)



### 3. Tìm kiếm tối ưu

**Procedure Best\_First\_Search;**

Begin

Open:={Start}; Close:=∅;

While (Open<> ∅) do

Begin

X= Retrieve(Open); // Chọn trạng thái tốt nhất

If (X=Goal) then Return True

Else Begin

Sinh ra các nút con của X;

For mỗi nút con Y của X do

Begin

Gán giá trị heuristic cho Y;

Open = Open  $\cup$  {Y};

End;

End;

End;

Return False;

End;

### 3. Tìm kiếm tối ưu (tt)

- ❖ Best First Search vs Breath First & Depth First Search
  - Best First search tương tự như Depth First & Breath First nhưng phần tử được xét tiếp là phần tử có giá trị heuristic tốt nhất.
  - Cần có một hàm đánh giá các trạng thái để xác định giá trị heuristic cho các trạng thái.
  - Không gian trạng thái vẫn không thay đổi về “toàn cục” tuy nhiên thường Heuristic search có không gian trạng thái làm việc nhỏ hơn Depth First và Breath First. **Tại sao??**
    - Do sự ôngh hưởng các trạng thái kế tiếp theo hướng có khả năng tìm ra nghiệm nhanh hơn nên số trạng thái xét dư thừa sẽ hạn chế -> sinh ít trạng thái con hơn
    - Điều này cũng là nguyên nhân làm cho Best First Search có thể dẫn đến kết quả là “nghiệm phụ” thay vì “nghiệm tối ưu”.

### 3. Tìm kiếm tối ưu (tt)

**Procedure A<sup>T</sup>;**

Begin

Open:={Start};

While (Open $\neq \emptyset$ ) do

Begin

X= Retrieve(Open); // Chọn X sao cho  $g(X)$  ) ạt min(Open)

If (X=Goal) then Return True

Else Begin

Sinh ra các nút con của X;

For mỗi nút con Y của X do

If(Y  $\notin$  Open)

Begin

$g(Y) := g(X) + cost(X, Y);$

Open = Open  $\cup$  {Y};

End;

Else So sánh  $g(Y)$  với  $g_{New}(Y)$  và cập nhật

End;

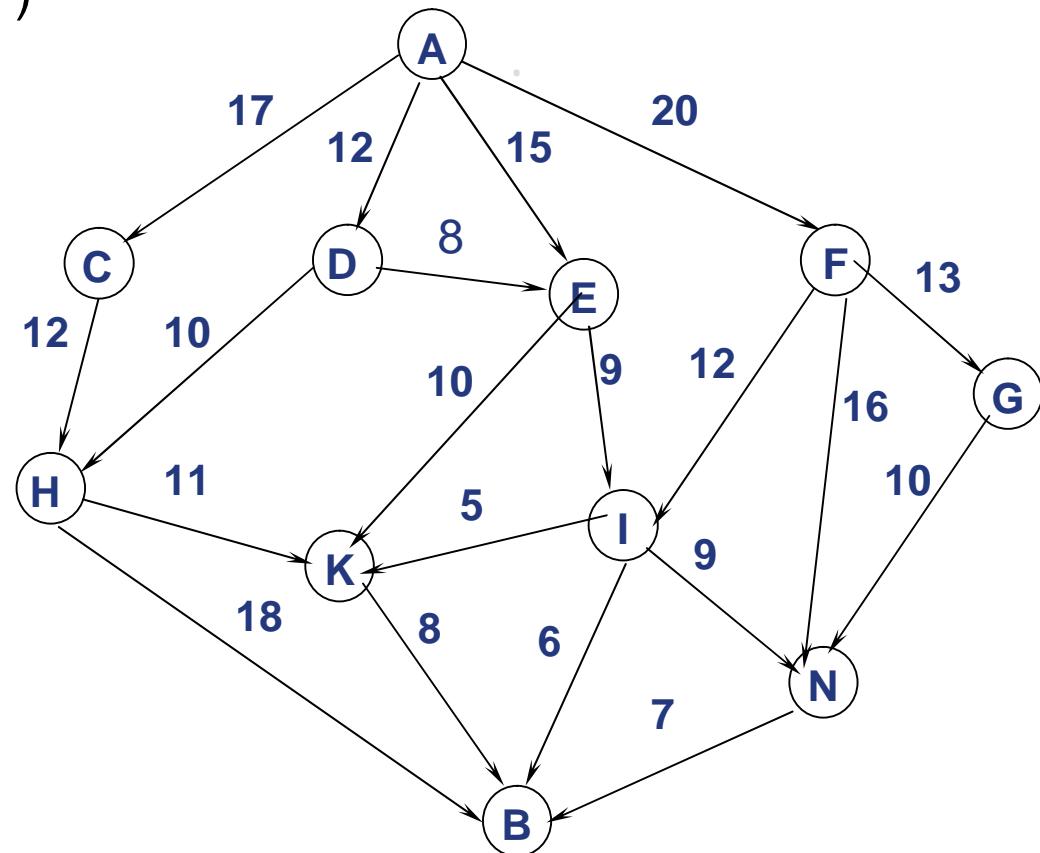
End;

Return False;

End;

### 3. Tìm kiếm tối ưu (tt)

Ví dụ: Giải bài toán tìm đường đi từ iểm A đến điểm B (với thuật toán A<sup>T</sup>)



### 3. Tìm kiếm tối ưu (tt)

**Procedure A<sup>KT</sup>;**

Begin

Open:={Start};

While (Open $\neq \emptyset$ ) do

Begin

X= Retrieve(Open); // Chọn X sao cho  $f(X)$  là min(Open)

If (X=Goal) then Return True

Else Begin

Sinh ra các nút con của X;

For mỗi nút con Y của X do

Begin

$g(Y)=g(X)+cost(X,Y);$

$f(Y)=g(Y)+h'(Y);$

Open = Open  $\cup \{Y\};$

End;

End;

End;

Return False;

End;



### 3. Tìm kiếm tối ưu (tt)

**Procedure A<sup>\*</sup>;**

Begin

Open:={Start}; Close:= Ø;

While (Open<>Ø) do

Begin

X= Retrieve(Open); // Chọn X sao cho  $f(x)$  ) ạt min(Open)

If (X=Goal) then Return True

Else Begin

Sinh ra các nút con của X;

For mỗi nút con Y của X do

If ( $Y \notin \text{Open}$ ) và ( $Y \notin \text{Close}$ )

Begin

Gán heuristic -  $f(Y)$ ;

$\text{Open} = \text{Open} \cup \{Y\}$ ;

End;

### 3. Tìm kiếm tối ưu (tt)



```
If (Y ∈ Open)
    If ( $f_{\text{new}}(Y) < f_{\text{old}}(Y)$ ) cập nhật lại Y trong Open;
If (Y ∈ Close)
    If ( $f_{\text{new}}(Y) < f_{\text{old}}(Y)$ )
        Begin
            Close=Close \ {Y};
            Open = Open  $\cup$  {Y};
        End;
    End;
Close=Close  $\cup$  {X};
End;
Return False;
End;
```

## 4. Hàm lượng giá heuristic

$$f(n) = g(n) + h(n)$$

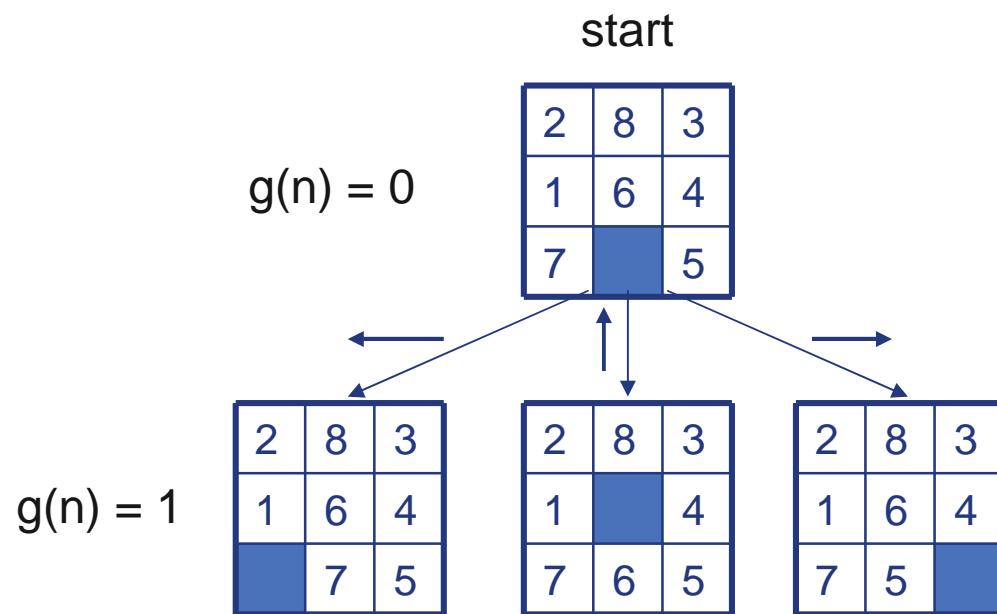
Trong đó:

- ✓  $f(n)$ : Hàm lượng giá heuristic tại trạng thái  $n$
- ✓  $g(n)$ : Khoảng cách từ  $n$  đến trạng thái bắt đầu
- ✓  $h(n)$ : Hàm heuristic đánh giá khoảng cách từ t.thái  $n$  đến mục tiêu

**Ví dụ:** Xét trò chơi 8-puzzle

1	2	3
8		4
7	6	5

goal



## 4. Hàm lượng giá heuristic

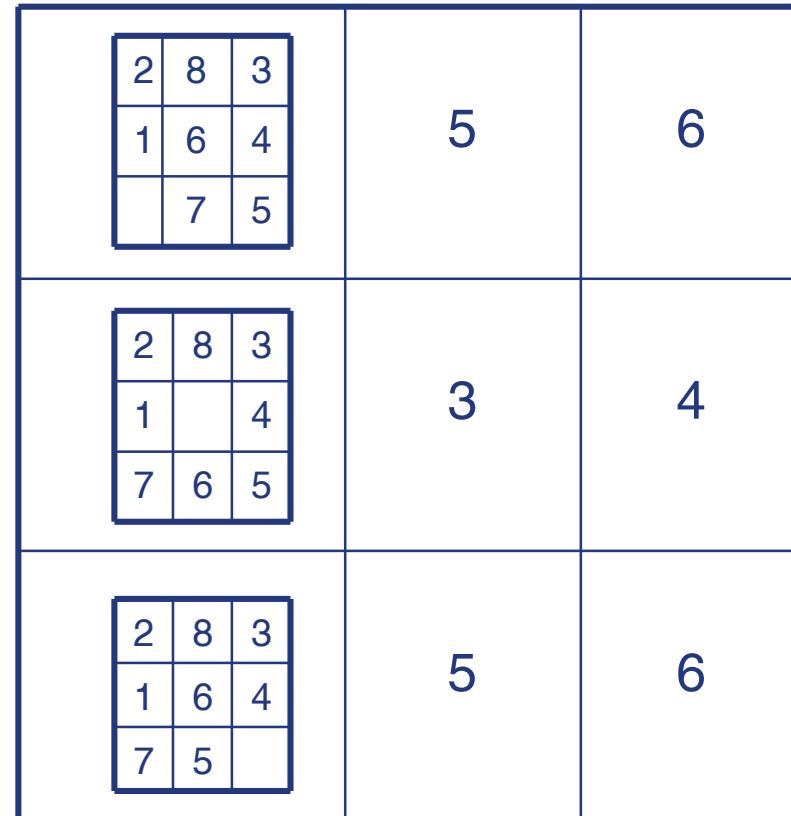


1	2	3
8		4
7	6	5

goal

Heuristic 1: Tổng số miếng sai vị trí

Heuristic 2: Tổng khoảng cách sai vị trí của từng miếng.



## 4. Hàm lượng giá heuristic

$$f(n) = g(n) + h(n)$$

Trong đó:

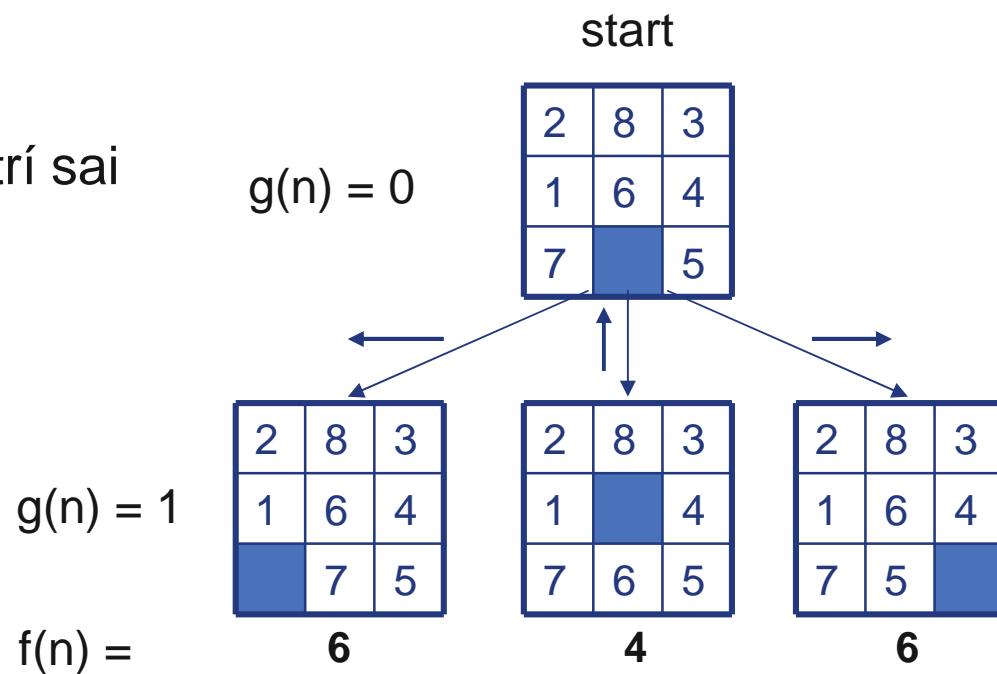
- ✓  $f(n)$ : Hàm lượng giá heuristic tại trạng thái  $n$
- ✓  $g(n)$ : Khoảng cách từ  $n$  đến trạng thái bắt đầu
- ✓  $h(n)$ : Hàm heuristic đánh giá khoảng cách từ t.thái  $n$  đến mục tiêu

**Ví dụ:** Xét trò chơi 8-puzzle.

$h(n)$ : số lượng các vị trí sai

1	2	3
8		4
7	6	5

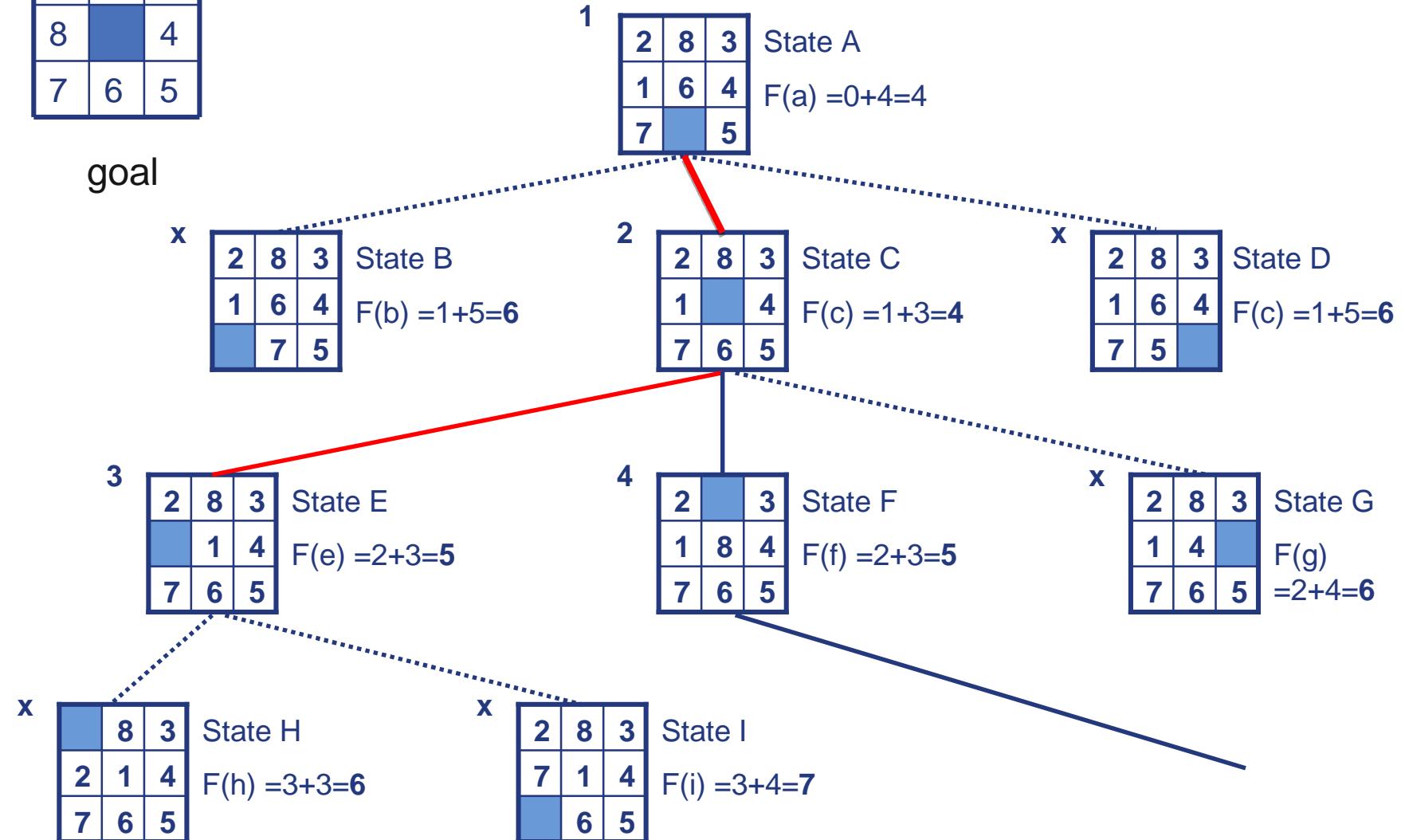
goal



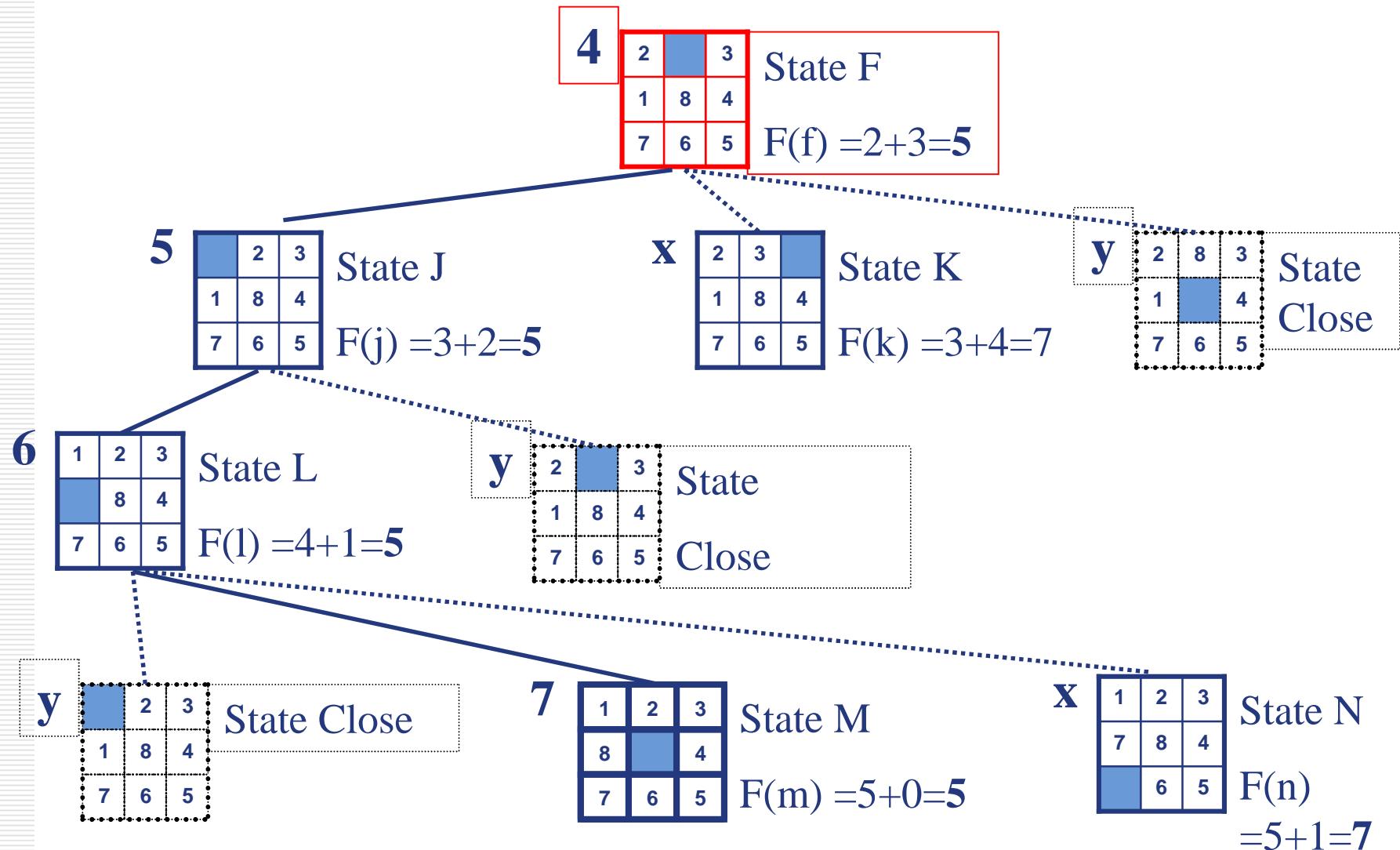
## 4. Hàm lượng giá heuristic (tt)

1	2	3
8		4
7	6	5

goal



## 4. Hàm lượng giá heuristic (tt)

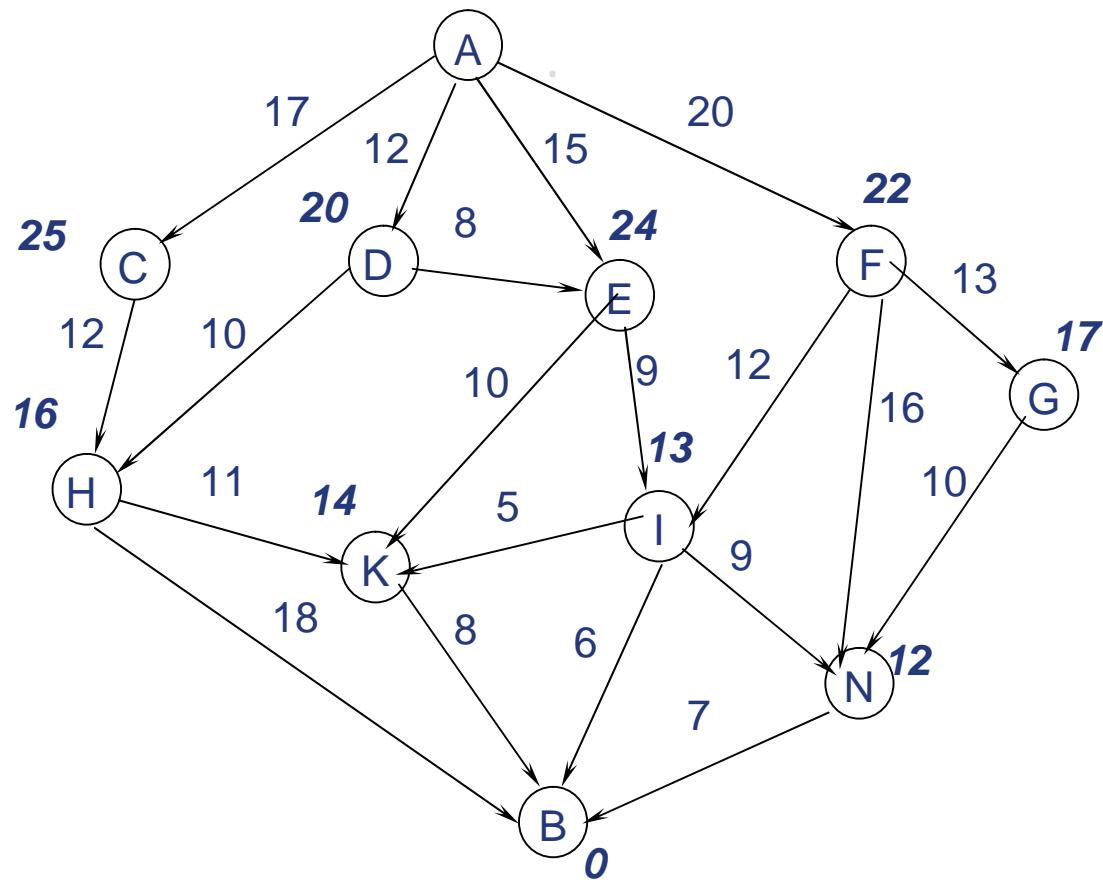


## 4. Hàm lượng giá heuristic (tt)

Lần	X	Open	Close
0		[a4]	[]
1	A4	[c4,b6,d6]	[a4]
2	C4	[e5,f5,g6,b6,d6]	[a4,c4]
3	E5	[f5,h6,g6,b6,d6,i7]	[a4,c4,e5]
4	F5	[j5,h6,g6,b6,d6,k7,i7]	[a4,c4,e5,f5]
5	J5	[l5,h6,g6,b6,d6,k7,i7]	[a4,c4,e5,f5,j5]
6	I5	[m5,h6,g6,b6,d6,k7,i7,n7]	[a4,c4,e5,f5,j5,I5]
7	m5		

## 4. Hàm lượng giá heuristic (tt)

Ví dụ: Giải bài toán tìm đường đi từ điểm A đến điểm B (với thuật toán A\*). Giá trị các đỉnh là hàm đánh giá  $h(n)$



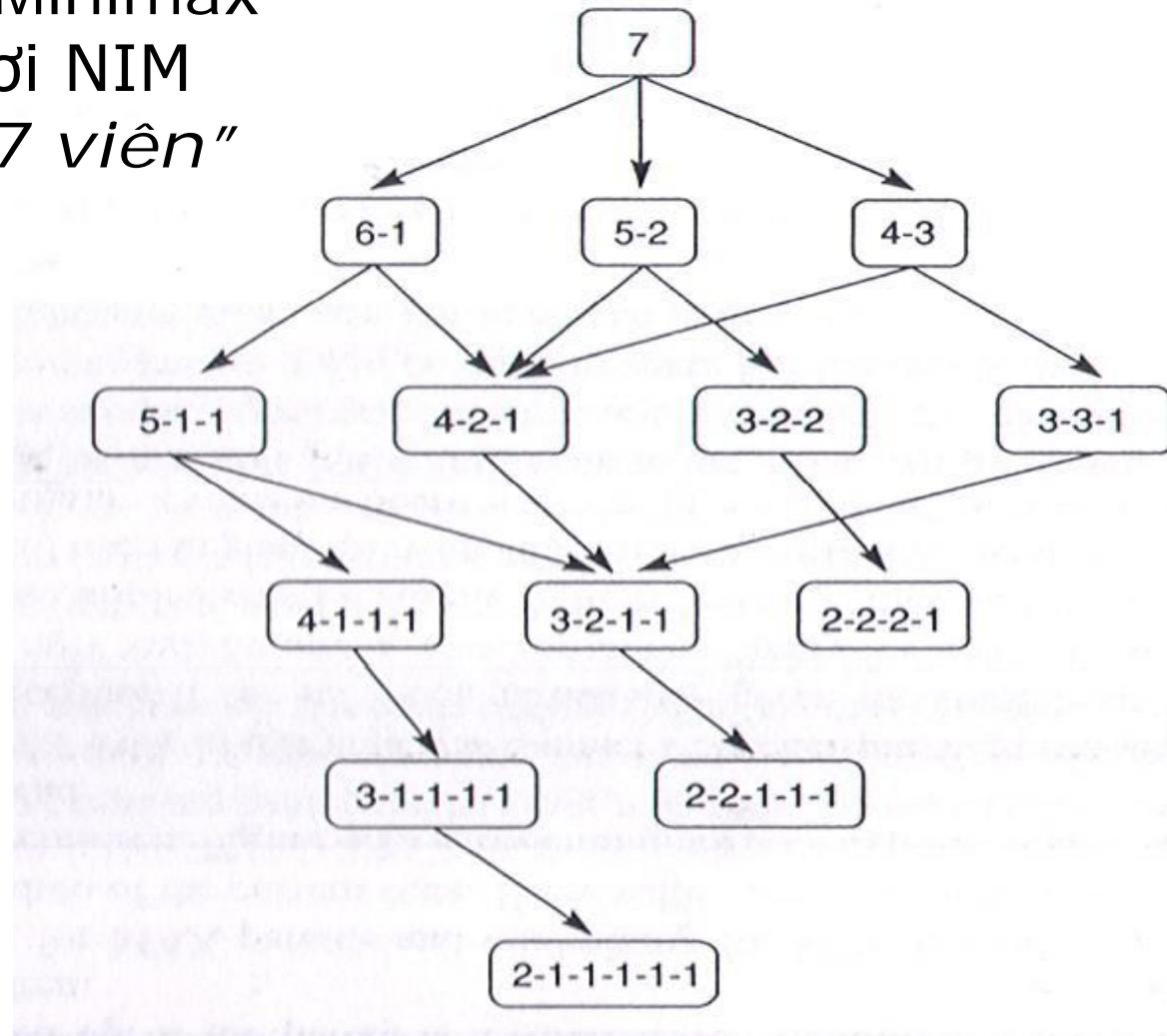
## 5. Heuristic trong trò chơi đối kháng

### ❖ Giải thuật minimax:

- Hai đấu thủ trong trò chơi được gọi là **MIN** và **MAX**.
- Mỗi nút lá có giá trị:
  - **1** nếu là MAX thắng,
  - **0** nếu là MIN thắng.
- Minimax sẽ truyền các giá trị này lên cao dần trên đồ thị, qua các nút cha mẹ kế tiếp theo các luật sau:
  - Nếu trạng thái cha mẹ là **MAX**, gán cho nó giá trị **lớn nhất** có trong các trạng thái con.
  - Nếu trạng thái cha, mẹ là **MIN**, gán cho nó giá trị **nhỏ nhất** có trong các trạng thái con.

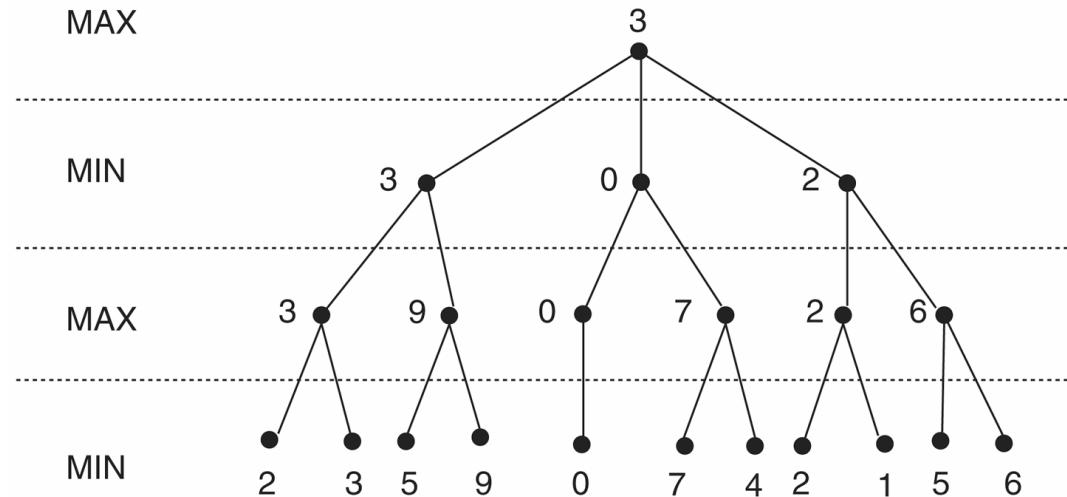
## 5. Heuristic trong trò chơi đối kháng (tt)

Áp dụng GT Minimax  
vào trò chơi NIM  
**“ĐốNg sỎi 7 viÊn”**



## 5. Heuristic trong trò chơi đối kháng (tt)

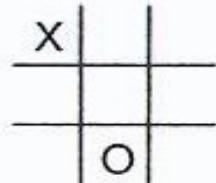
- ❖ Minimax với **độ sâu lớp cõi cịnh**
- ❖ Minimax đối với một KGTT giả hịnh.



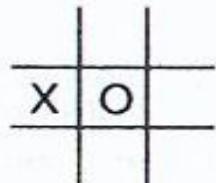
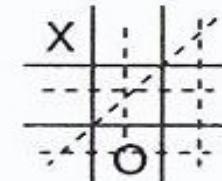
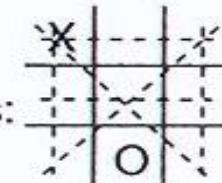
- ❖ Các nút lá được gán các giá trị *heuristic*
- ❖ Còn giá trị tại các nút trong là các giá trị nhận được dựa trên giải thuật Minimax

## 5. Heuristic trong trò chơi đối kháng (tt)

- Heuristic trong trò chơi tic-tac-toe



X has 6 possible win paths:  
O has 5 possible wins:  
 $E(n) = 6 - 5 = 1$



X has 4 possible win paths;  
O has 6 possible wins  
 $E(n) = 4 - 6 = -2$

### Hàm Heuristic:

Trong đó:

$$E(n) = M(n) - O(n)$$

M(n) là tổng số đường thắng có thể của tôi

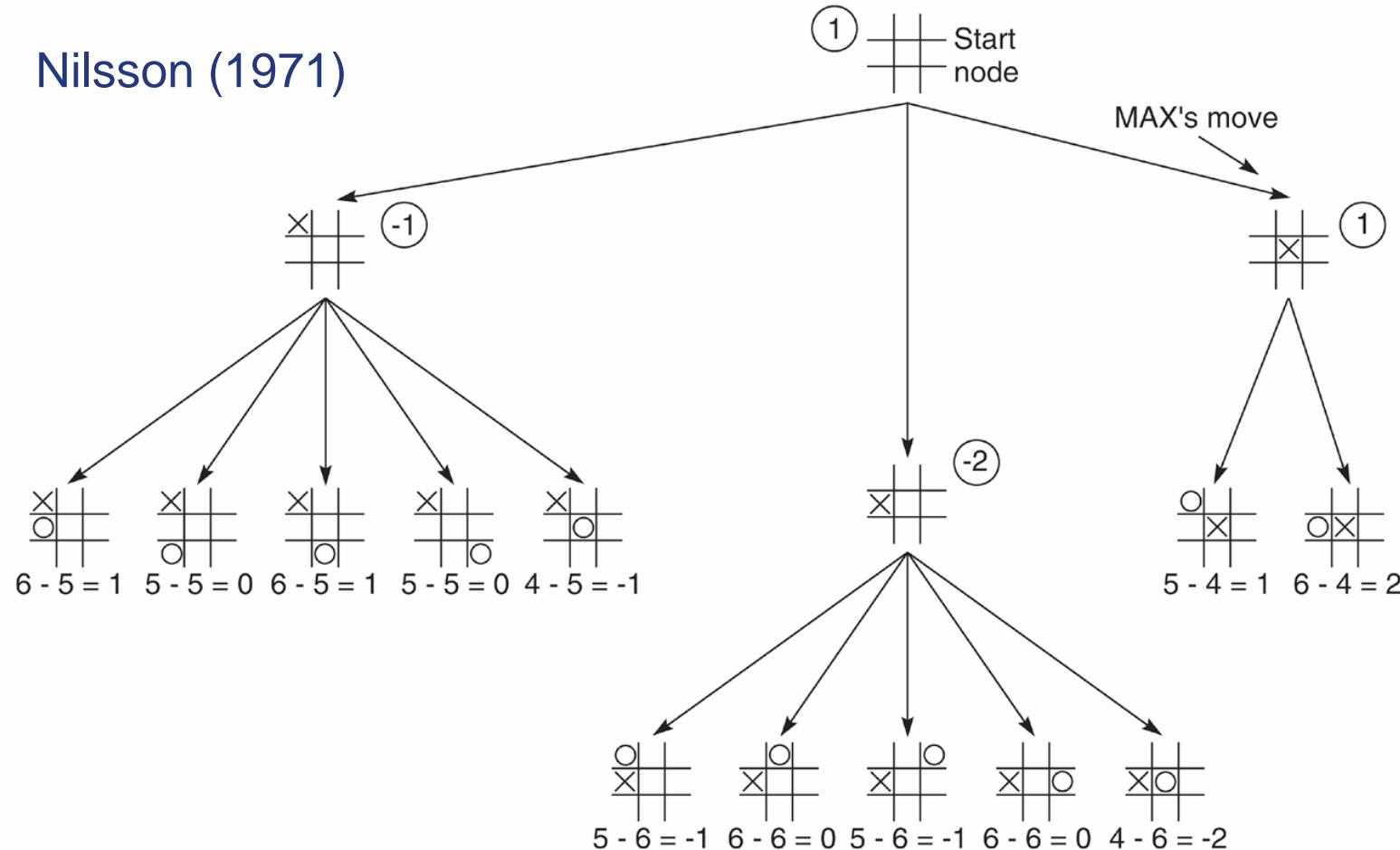
O(n) là tổng số đường thắng có thể của đối thủ

E(n) là trị số đánh giá tổng cộng cho trạng thái n

## 5. Heuristic trong trò chơi đối kháng (tt)

Minimax 2 lớp được áp dụng vào nước đi mở đầu trong tic-tac-toe

Nilsson (1971)



## 5. Heuristic trong trò chơi đối kháng (tt)

- **Giải thuật cắt tỉa  $\alpha$ - $\beta$**

- ❖ Tìm kiếm theo kiểu depth-first.
- ❖ Nút MAX có 1 giá trị  $\alpha$  (luôn tăng)
- ❖ Nút MIN có 1 giá trị  $\beta$  (luôn giảm)
- ❖ TK có thể kết thúc dưới bất kỳ:
  - Nút MIN nào có  $\beta \leq \alpha$  của bất kỳ nút cha MAX nào.
  - Nút MAX nào có  $\alpha \geq \beta$  của bất kỳ nút cha MIN nào.
- ❖ Giải thuật cắt tỉa  $\alpha$ - $\beta$  thể hiện *mối quan hệ giữa các nút ở lớp  $n$  và  $n+2$* , mà tại đó toàn bộ cây có gốc tại lớp  $n+1$  có thể cắt bỏ.

## 5. Heuristic trong trò chơi đối kháng (tt)



### Cắt tỉa $\alpha$

MAX

MIN

$S = \alpha$

$A = \alpha$

Z

$\geq \alpha$

$\alpha$  - cut

= Z

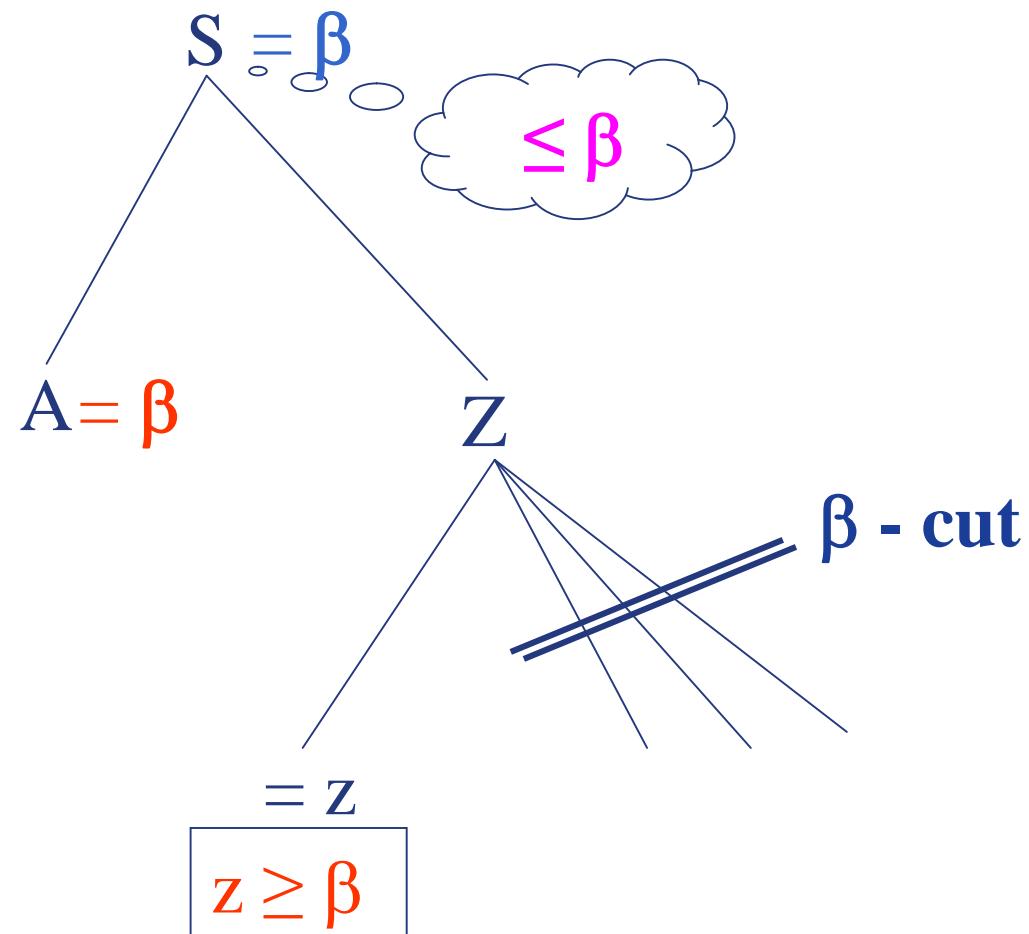
$Z \leq \alpha$

## 5. Heuristic trong trò chơi đối kháng (tt)

Cắt tỉa  $\beta$

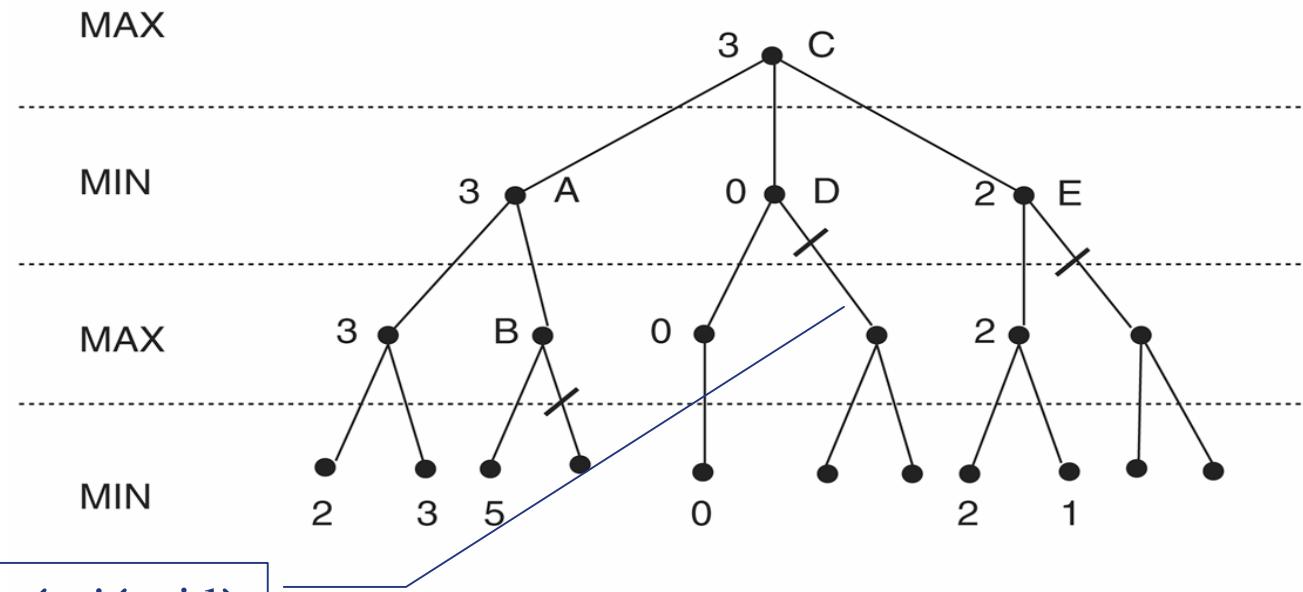
MIN

MAX



## 5. Heuristic trong trò chơi đối kháng (tt)

- Áp dụng cho KGTT giả cinh



Các nút không có giá trị là các nút không được duyệt qua

A has  $\beta = 3$  (A will be no larger than 3)  
 B is  $\beta$  pruned, since  $5 > 3$   
 C has  $\alpha = 3$  (C will be no smaller than 3)  
 D is  $\alpha$  pruned, since  $0 < 3$   
 E is  $\alpha$  pruned, since  $2 < 3$   
 C is 3

# Bài tập Chương 3

## Tìm hiểu các bài toán

- |                       |                               |
|-----------------------|-------------------------------|
| 1. Đổi tiền           | 8. Bài toán TSP               |
| 2. Bài toán phân việc | 9. Puzzle                     |
| 3. BFS, DFS           | 10. Cờ vua, sẮp hậu           |
| 4. Tic tac toe        | 11. Cờ tướng                  |
| 5. Trò chơi NIM       | 12. Người nông dân qua sông   |
| 6. Đong dầu           | 13. Con thỏ và con cáo        |
| 7. Tô màu bản đồ      | 14. Con khỉ và nǎi chuối      |
|                       | 15. Tu sỹ và kẻ ăn thịt người |



# End of Chapter 3

*Bùi Đức Dương – Khoa Công nghệ Thông tin*

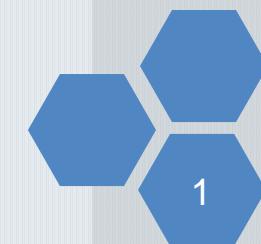
# *Bài giảng Trí tuệ nhân tạo*

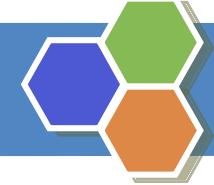


## Chương 4 Biểu diễn & suy diễn tri thức



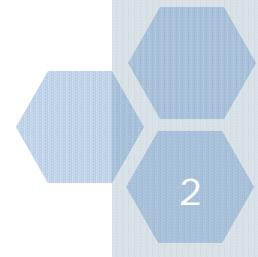
Bùi Đức Dương  
*Khoa Công nghệ Thông tin*

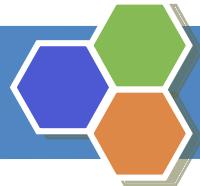




# Nội dung

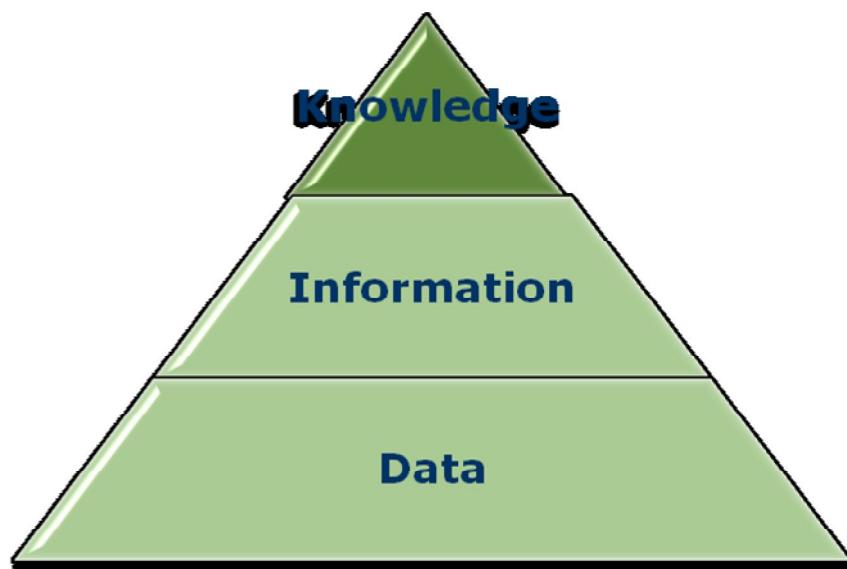
- 1. Giới thiệu**
- 2. Các loại tri thức**
- 3. Các kỹ thuật biểu diễn tri thức**
- 4. Suy diễn dữ liệu**
- 5. Chứng minh mệnh đề**



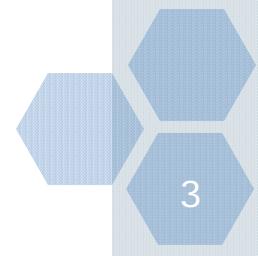


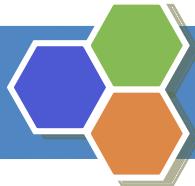
## 1. Giới thiệu

- ❖ Tri thức
  - Là sự hiểu biết về một vấn đề nào đó



- ❖ Lĩnh vực
- ❖ Biểu diễn tri thức





## 2. Các loại tri thức

### ❖ Tri thức thủ tục

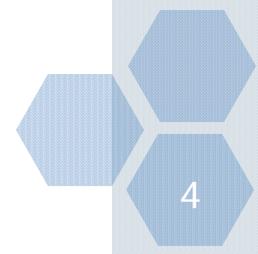
- Mô tả cách thức giải quyết một vấn đề
- Đề xuất các giải pháp thực hiện công việc
- Tiêu biểu là các luật, chiến lược, lịch trình & thủ tục

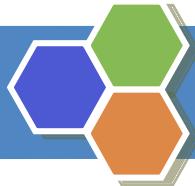
### ❖ Tri thức khai báo

- Mô tả vấn đề trực quan
- Tiêu biểu là các phát biểu đơn giản (True/False); danh sách các khẳng định

### ❖ Siêu tri thức (*meta – knowledge*)

- Mô tả tri thức về tri thức
- Giúp lựa chọn tối ưu tri thức





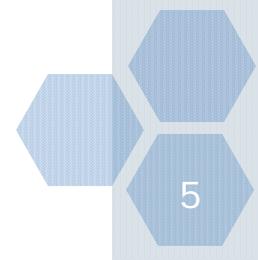
## 2. Các loại tri thức (tt)

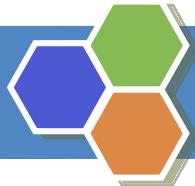
### ❖ Tri thức heuristic

- Mô tả các “mẹo” dẫn dắt tiến trình lập luận
- Không đảm bảo tối ưu về kết quả (chính xác dưới 100%)
- Chuyển các luật, sự kiện... thành tri thức heuristic để thuận tiện trong xử lý

### ❖ Tri thức có cấu trúc

- Mô tả tri thức theo cấu trúc
- Mô tả mô hình tổng quan hệ thống theo quan điểm của chuyên gia: khái niệm, khái niệm con & các đối tượng
- Diễn tả chức năng và mối liên hệ giữa các tri thức dựa theo cấu trúc xác định.





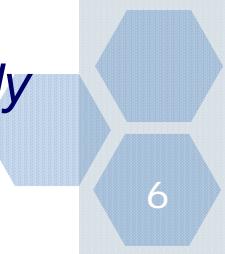
### 3. Các kỹ thuật biểu diễn tri thức

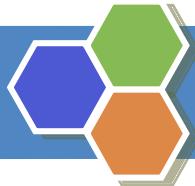
#### ❖ Bộ ba Đối tượng – Thuộc tính – Giá trị

- O – A – V: Object-Attribute-Value
- Ví dụ: *Quả cam màu vàng*



- Trong các sự kiện O – A – V:
  - ✓ Một đối tượng có thể có nhiều thuộc tính
  - ✓ Một thuộc tính có một (*đơn trị/single - valued*) hay nhiều giá trị (*đa trị/multi - valued*): *Cho phép biểu diễn tri thức linh hoạt*
  - ✓ Xác định độ tin cậy: *Nhân tố chắc chắn CF (certainly factor)*

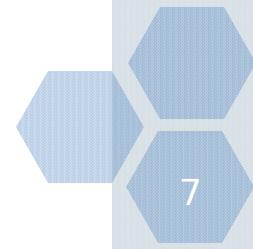


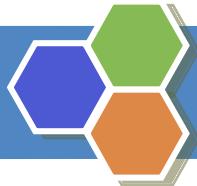


### 3. Các kỹ thuật biểu diễn tri thức (tt)

#### ❖ Các luật dẫn

- Là cấu trúc tri thức dùng để liên kết thông tin đã biết với thông tin khác giúp đưa ra các suy luận, kết luận
- Bộ suy diễn: *Module xử lý các luật trong hệ thống dựa trên các luật được quản lý trong hệ thống (CSTT & thông tin trong bộ nhớ)*
- 7 dạng luật cơ bản
- ✓ *Quan hệ*
  - IF** Không cài đặt HĐH
  - THEN** Máy tính sẽ không khởi động được
- ✓ *Lời khuyên*
  - IF** Máy tính không khởi động được
  - THEN** Xử lý dữ liệu bằng tay





### 3. Các kỹ thuật biểu diễn tri thức (tt)

- ✓ *Hướng dẫn*

**IF** Máy tính không khởi động AND Phần cứng tốt

**THEN** Kiểm tra hệ điều hành

- ✓ *Chiến lược*

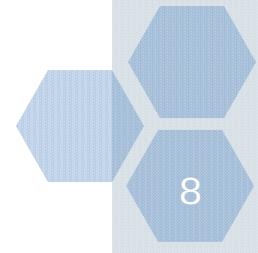
**IF** Máy tính không khởi động được

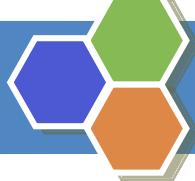
**THEN** Đầu tiên kiểm tra có internet không, sau đó hển phần cứng và phần mềm

- ✓ *Điễn giải*

**IF** Máy tính khởi động được

**THEN** Hệ điều hành bình thường





### 3. Các kỹ thuật biểu diễn tri thức (tt)

✓ *Chẩn đoán*

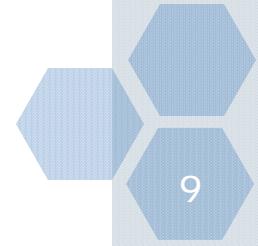
**IF** Phần cứng tốt AND Máy chạy chậm AND hay shutdown đột ngột

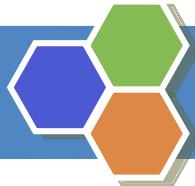
**THEN** Bị virus

✓ *Thiết kế:*

**IF** Là nam giới AND làm ngành CNTT

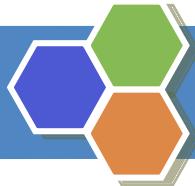
**THEN** Nên mua máy Dell OR máy IBM





### 3. Các kỹ thuật biểu diễn tri thức (tt)

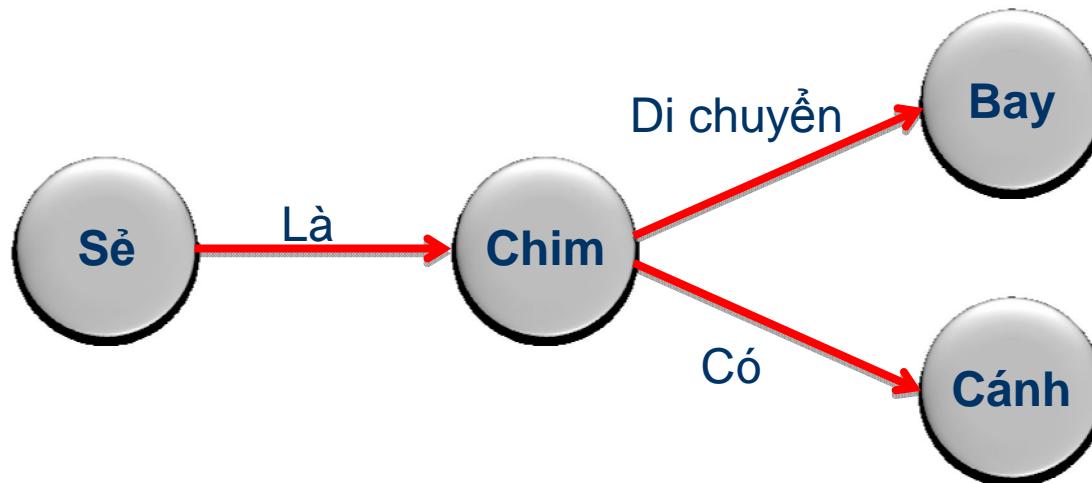
- Mở rộng cho các luật
- ✓ Luật có biến
  - Áp dụng cần thực hiện cùng một phép toán trên một tập hay các đối tượng.
  - Ví dụ: IF X là viên chức AND X là nam AND Tuổi X>60  
THEN X có thể nghỉ hưu
- ✓ Luật không chắc chắn
  - Sự kiện có thể không chấp chắc chắn: Mệnh đề phát biểu (luật) đưa thêm hệ số chắc chắn CF.
  - Ví dụ: IF Giá vàng tăng THEN Giá hắt tăng, CF=0.9
- ✓ Siêu luật (meta-rules)
  - Luật mô tả cách thức dùng các luật khác (Không có thông tin mới)
  - Ví dụ: IF Máy không khởi động AND Phần cứng tốt  
THEN Sử dụng các luật liên quan đến HĐH



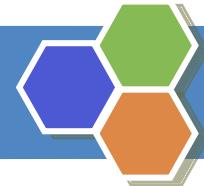
### 3. Các kỹ thuật biểu diễn tri thức (tt)

- ❖ Mạng ngũ nghĩa

- Là phương pháp biểu diễn tri thức dùng đồ thị: Nút biểu diễn đối tượng; cung biểu diễn quan hệ.

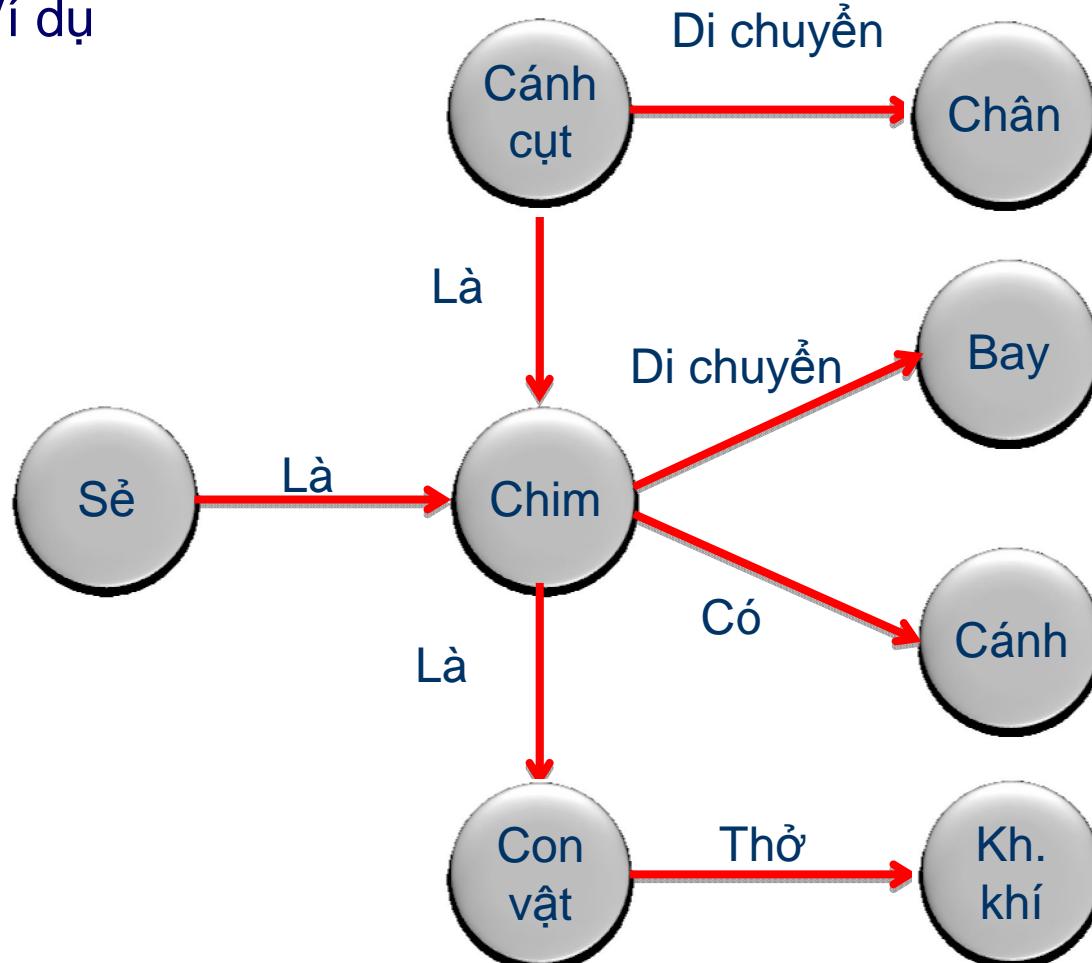


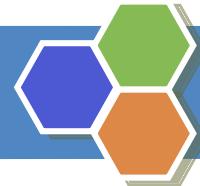
- Mở rộng mạng ngũ nghĩa: Thêm các nút (đối tượng bổ sung) và nối vào đồ thị bằng các cung:
  - ✓ Thêm một đối tượng tương tự
  - ✓ Thêm đối tượng đặc biệt hơn
  - ✓ Thêm một đối tượng tổng quát hơn



### 3. Các kỹ thuật biểu diễn tri thức (tt)

➤ Ví dụ

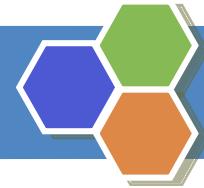




### 3. Các kỹ thuật biểu diễn tri thức (tt)

#### ❖ Frames

- Có hình thức như bảng mẫu, tờ khai cho phép điền vào chỗ trống
- Cấu trúc cơ bản:
  - ✓ Tên frame
  - ✓ Lớp
  - ✓ Các thuộc tính: Biểu diễn như O – A - V

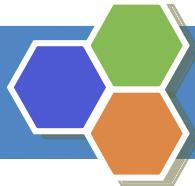


### 3. Các kỹ thuật biểu diễn tri thức (tt)

#### ❖ Frames

Tên frame	
Lớp	<input type="text"/>
	<input type="text"/>
Thuộc tính	
Thuộc Tính	Giá trị
TT1	GT1
TT2	GT2
...	...

PHIẾU ĐIỂM	
Họ tên	<input type="text"/>
MSSV	<input type="text"/>
Môn	Điểm
CTDL	10
TTNT	7
CSTT	9



### 3. Các kỹ thuật biểu diễn tri thức (tt)

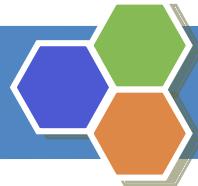
#### ❖ Logic

- Sử dụng ký hiệu để thể hiện tri thức & toán tử

Phép toán	NOT	AND	OR	Kéo theo	Tương đương
Ký hiệu	$\neg, \sim$	$\wedge, \cap, \&$	$\vee, \cup, +$	$\supset, \rightarrow$	$\equiv, \leftrightarrow$

- Logic mệnh đề

- Logic vị từ

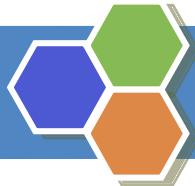


### 3. Các kỹ thuật biểu diễn tri thức (tt)

- Logic mệnh đề
- ✓ Mệnh đề: Phát biểu/khẳng định đúng/sai
- ✓ Các phép toán logic

X	Y	$\neg X$	$X \wedge Y$	$X \vee Y$	$X \rightarrow Y$	$X \leftrightarrow Y$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

- ✓ Ví dụ:
  - IF Phần cứng hỏng (A) OR Chưa cài đặt HĐH (B)  
THEN Máy tính không khởi động được (C)  
Có thể biểu diễn là:  $A \vee B \rightarrow C$



### 3. Các kỹ thuật biểu diễn tri thức (tt)

#### ➤ Logic vị từ

- ✓ Phép toán mệnh đề → suy diễn tự hông nhưng **chưa đủ** khi cần phải truy cập vào thành phần nhỏ trong câu, dùng biến số trong câu.

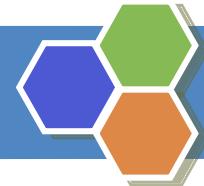
**Ví dụ:**

***“Mọi sinh viên trường ĐHNT đều có bằng tú tài. Lan không có bằng tú tài. Do vậy, Lan không là sinh viên trường ĐHNT”***

“Lan” là một đối tượng cụ thể của “SV trường ĐHNT” – không thể đặc tả được “quan hệ” này trong mệnh đề được mà chỉ có thể là:

***“LAN là sinh viên trường ĐHNT thì Lan có bằng tú tài. Lan không có bằng tú tài. Do vậy, Lan không là sinh viên trường ĐHNT”***

- ✓ Mệnh đề phải giải quyết bằng cách liệt kê tất cả các trường hợp  
→ Không khả thi
- ✓ Do đó, chúng ta cần một Logic khác hơn là phép toán mệnh đề:



### 3. Các kỹ thuật biểu diễn tri thức (tt)

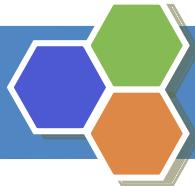
- ✓ *Vị từ là một phát biểu nói lên quan hệ giữa một đối tượng với các thuộc tính của nó hay quan hệ giữa các đối tượng với nhau.*
- ✓ Vị từ được **biểu diễn** bởi một tên được gọi là **tên vị từ**, sau nó là một danh sách các thông số.

**Ví dụ:**

Phát biểu: “**Nam là sinh viên trường ĐHNT**”

Biểu diễn: **sv\_NT(“Nam”)**

Ý nghĩa: đối tượng tên là “Nam” có thuộc tính là “sinh viên trường ĐHNT”.

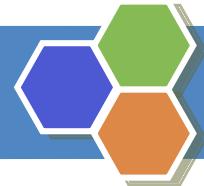


### 3. Các kỹ thuật biểu diễn tri thức (tt)

✓ **Biểu thức Vị từ:** là sự kết hợp của các vị từ bởi các phép toán vị từ.

✓ **Các phép toán:**

$\neg$	Phủ hịnh	- một ngôi.
$\forall X$	Với mọi	- một ngôi
$\exists X$	Tồn tại	- một ngôi
$\wedge$	Hội	- hai ngôi.
$\vee$	Tuyễn	- hai ngôi.
$\Rightarrow$	Suy ra	- hai ngôi.
$=$	Tương đương	- hai ngôi.



### 3. Các kỹ thuật biểu diễn tri thức (tt)

#### Ví dụ:

Chuyển các câu sau sang biểu thức vị từ:

*“Mọi sinh viên trường ĐHNT đều có bằng tú tài.*

*Lan không có bằng tú tài.*

*Do vậy, Lan không là sinh viên trường ĐHNT”*

Với  $sv\_NT(X)$  cho biết: “*X là sinh viên trường DHNT*”

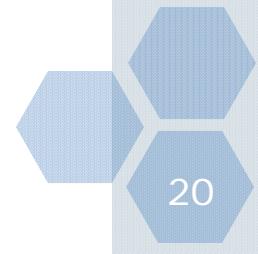
$tu\_tai(X)$  cho biết: “*X có bằng tú tài*”

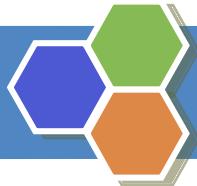
Các câu trên được chuyển qua vị từ là:

$\forall X(sv\_NT(X) \Rightarrow tu\_tai(X))$ .

$\neg tu\_tai(\text{"Lan"})$ .

*Do vậy,  $\neg sv\_NT(\text{"Lan"})$ .*





## 4. Suy diễn dữ liệu

### ❖ Suy lý

Là quá trình sử dụng các sự kiện riêng của bài toán và tri thức tổng thể của lĩnh vực để rút ra kết luận.

#### ➤ Suy lý theo cách suy diễn

✓ Sử dụng sự kiện (tiền đề) và tri thức chung liên quan ở các dạng luật kéo theo.

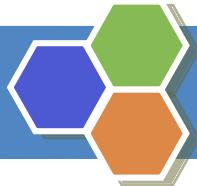
$$(A \rightarrow B) \wedge (A = \text{True}) \Rightarrow (B = \text{True})$$

#### Ví dụ:

**Kéo theo:** IF Phần cứng hỏng THEN Máy tính không khởi động  
được

**Tiền đề:** Phần cứng hỏng

**Kết luận:** Máy tính không khởi động được



## 4. Suy diễn dữ liệu (tt)

- Suy lý quy nạp
- ✓ Rút ra kết luận (tri thức) tổng quát từ một tập các sự kiện.

**Ví dụ:**

**Tiền đề:** Máy tính Dell có CPU

**Tiền đề:** Máy tính Sony có CPU

**Kết luận:** Máy tính có CPU

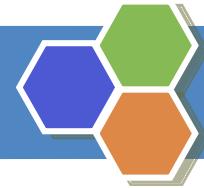
- Suy lý giả : ịnh
- ✓ Kết luận có thể húng hoặc sai.

**Ví dụ:**

**Kéo theo:** Máy tính không khởi động được nếu cúp điện

**Tiền đề:** Máy tính không khởi động.

**Kết luận:** Cúp điện?



## 4. Suy diễn dữ liệu (tt)

- Suy lý tương tự, loại suy
- ✓ Vạch ra điểm tương tự giữa 2 vật so sánh, rút ra điểm giống và khác nhau.

**Ví dụ:**

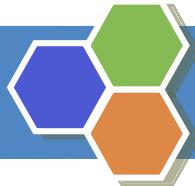
***Khung:*** Con hổ

***Chủng loại:*** Thú vật

***Ăn:*** Thịt

***Sống tại:*** Ấn Độ và Đông Nam Á

***Màu:*** Vàng có vạch



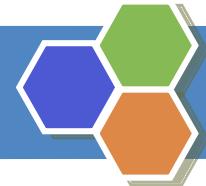
## 4. Suy diễn dữ liệu (tt)

- Suy lý theo lẽ thường
- ✓ Sử dụng kinh nghiệm để nhanh chóng rút ra kết luận.

### Ví dụ:

Siết quạt lồng thường gây tiếng ồn.

- Suy lý không đơn điệu
- ✓ Khi có sự kiện thay đổi, đưa vào sự kiện phụ thuộc khác để đưa ra kết luận mong muốn.



## 4. Suy diễn dữ liệu (tt)

### ❖ Suy diễn

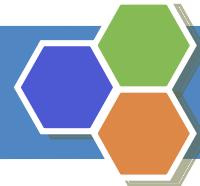
Là quá trình rút ra thông tin mới từ các thông tin cũ

#### ➤ Modus ponens

1. E1
2.  $E1 \rightarrow E2$
3. E2

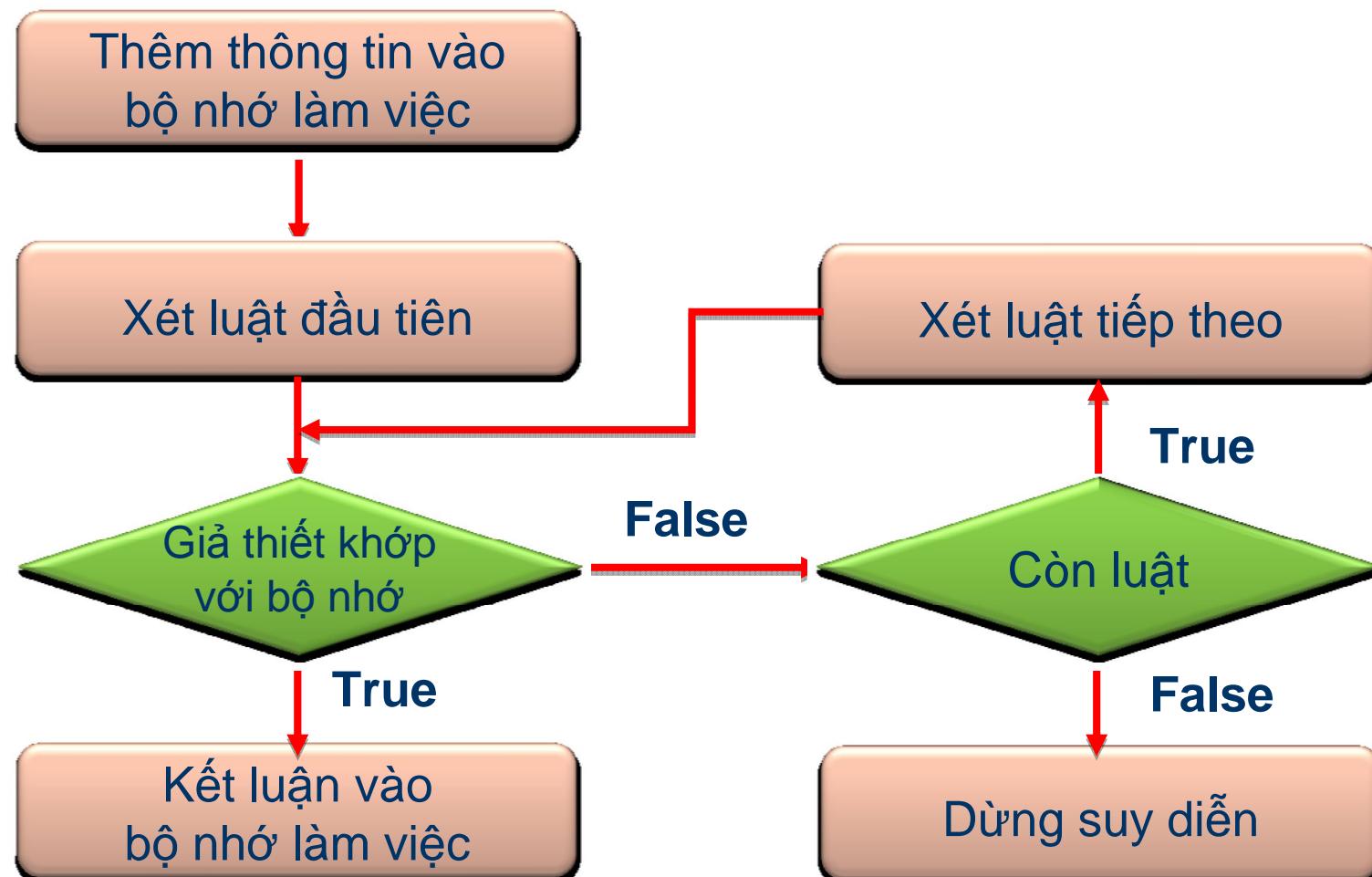
#### ➤ Modus tollens

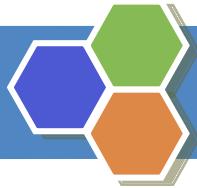
1.  $\neg E2$
2.  $E1 \rightarrow E2$
3.  $\neg E1$



## 4. Suy diễn dữ liệu (tt)

### ➤ Suy diễn tiến





## 4. Suy diễn dữ liệu (tt)

✓ Suy diễn tiên với logic mệnh đề

**Input:** Tập luật Rule=  $\{r_1, r_2, \dots, r_m\}$ ; GT; KL.

**Output:** Return “True” nếu  $GT \rightarrow KL$

Ngược lại, return “False”.

**Method:**

$TD = GT;$

$T = Filter(Rule, TD);$

while ( $KL \not\subset TD$ ) and ( $T \neq \emptyset$ ) do

{

$r = Get(T);$

$TD = TD \cup \{q\}; \quad // r: left \rightarrow q$

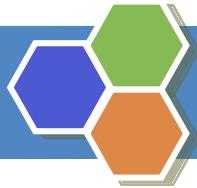
$Rule = Rule \setminus \{r\};$

$T = Filter(Rule, TD);$

}

if ( $KL \subset TD$ ) then return “True”

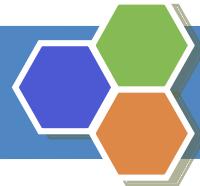
else return “False”



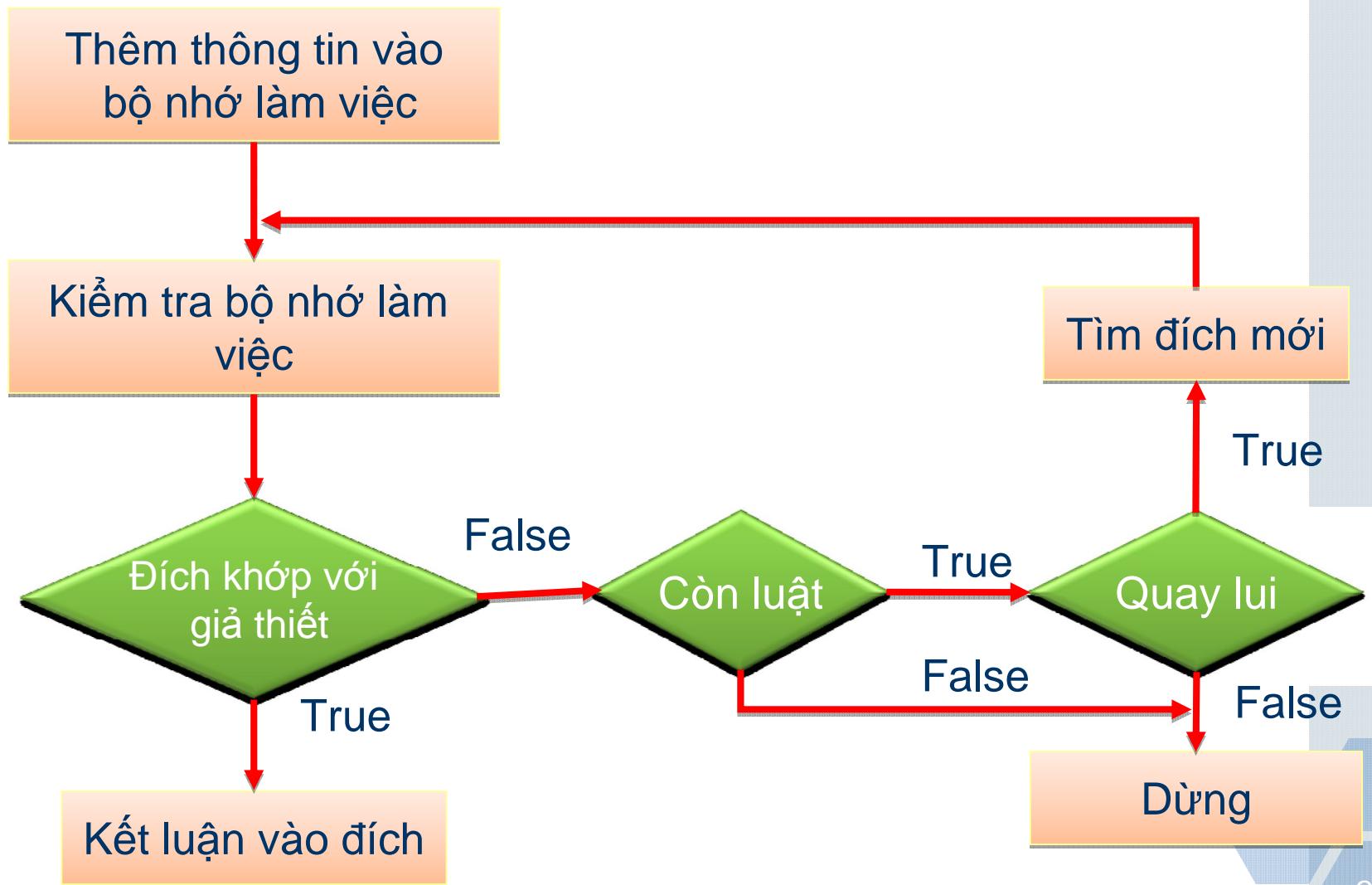
## 4. Suy diễn dữ liệu (tt)

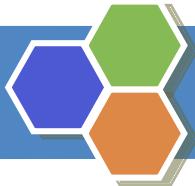
### ➤ Suy diễn lùi

- ✓ Thủ tục bắt đầu tìm kiếm từ dữ liệu đích của bài toán.
- ✓ Chọn tất cả các luật ứng với về kết luận hợp với dữ liệu đích, thiết lập dữ liệu ở về iều kiện phát sinh ra đích làm dữ liệu đích mới.
- ✓ Tại mỗi điểm dữ liệu đích mới, chọn tất cả các luật ứng với về kết luận hợp với đích mới, thiết lập dữ liệu ở iều kiện làm dữ liệu đích mới hơn.
- ✓ Thủ tục này lặp lại cho tất cả các đích mới cho đến khi nào dữ liệu ban đầu của bài toán được tìm thấy.



## 4. Suy diễn dữ liệu (tt)





## 4. Suy diễn dữ liệu (tt)

✓ Suy diễn lùi với loaic mēnh đē

**Input:** Tập luật Rule=  $\{r_1, r_2, \dots, r_m\}$ ; GT; KL.

**Output:** Return “True” nếu GT $\rightarrow$ KL  
Ngược lại, return “False”.

**Method:**

If (KL  $\subseteq$  GT) then return “True”

Else

{

TĐ= $\emptyset$ ; Vết =  $\emptyset$ ; First=1; QuayLui= False;

}

For (Each  $q \in KL$ ) do TĐ=TĐ $\cup\{(q,0)\}$ ;

Repeat

{

first ++;

(f,i)=Get(TĐ);



## 4. Suy diễn dữ liệu (tt)

```
If (fnotin GT) then
{
    j=TìmLuật(f,i,Rule); // rj: Leftj → f
    If (Tìm có rj) THEN
    {
        Vết = Vết ∪ {(f,j)};
        For (Each t ∈ (Leftj\GT)) do TD = TD ∪ {(t,0)};
    }
else
{
    QuayLui=True;
    While ((fnotin KL) and Quaylui) do
    {
        Repeat
        {
            (g,k)=Get(Vết);
            TD = TD \ Leftk;
        }
        Until (f∈Leftk);
```

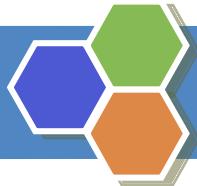


## 4. Suy diễn dữ liệu (tt)

```
I=Tìmluat(g,k,Rule);
If (Tìm có rI) Then
{
    TD = TD \ Leftk;
    For (Each t∈(LeftI\GT)) do
        TD= TD∪{(t,0)};
    Vết = Vết ∪ {(g,I)};
    Quaylui = False;
} //end if3

else f=g;
} //end while
} //end if2
} //end if1

Until (TD = ∅) or ((f ∈KL) and (First>2));
If (f ∈KL) then Return False
else Return True;
```



## 4. Suy diễn dữ liệu (tt)

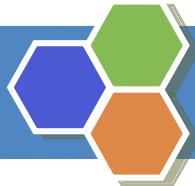
### ❖ So sánh ưu điểm

#### ➤ Suy diễn tiến

- ✓ Làm việc tốt đối với bài toán thu thập thông tin rồi tìm ra điều cần suy diễn.
- ✓ Cho khối lượng lớn các thông tin (mới) từ một số thông tin ban đầu.
- ✓ Lý tưởng cho các bài toán lập kế hoạch, hiều hành, hiều khiển và diễn dịch.

#### ➤ Suy diễn lùi

- ✓ Phù hợp với bài toán đưa ra giả thuyết rồi kiểm chứng giả thuyết đó đúng hay không.
- ✓ Tập trung vào đích đã cho. Tạo ra một loạt câu hỏi liên quan đến vấn đề, hoàn cảnh đang xét.
- ✓ Chỉ tìm trên một không gian con CSTT liên quan đến bài toán đang xét.



## 4. Suy diễn dữ liệu (tt)

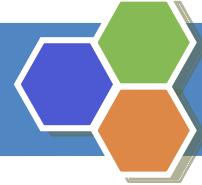
### ❖ So sánh nhược điểm

#### ➤ Suy diễn tiến

✓ Không cảm nhận được: Chỉ một số thông tin là quan trọng. Hệ thống có thể hỏi câu không liên quan và câu trả lời có thể không quan trọng.

#### ➤ Suy diễn lùi

✓ Thường phải tiếp dòng suy diễn: Không dừng đúng lúc.



## 5. Chứng minh mệnh đề

### ➤ Thuật giải Vương Hạo

Bước 1: Phát biểu lại giả thiết và kết luận của vấn đề dưới dạng chuẩn như sau:

$$GT_1, GT_2, \dots, GT_n \rightarrow KL_1, KL_2, \dots, KL_m$$

Trong đó các  $GT_i$  và  $KL_j$  được xây dựng từ các biến mệnh đề và các phép toán  $\wedge$ ,  $\vee$ ,  $\neg$ .

Bước 2: Chuyển về các  $GT_i$  và  $KL_j$  có dạng phủ định.

Ví dụ:

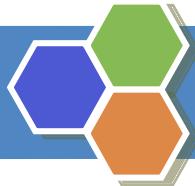
$$(p \rightarrow q, \neg(r \rightarrow s), \neg q, p \vee r) \rightarrow (s, \neg p)$$

$$\Leftrightarrow (p \rightarrow q, p \vee r, p) \rightarrow (s, r \rightarrow s, q)$$

Bước 3: Thay dấu “ $\wedge$ ” ở trong  $GT_i$  và dấu “ $\vee$ ” ở trong  $KL_j$  bằng dấu “,” (phẩy).

Ví dụ:

$$p \wedge q, r \wedge (p \vee s) \rightarrow q \vee r \Leftrightarrow p, q, r, (p \vee s) \rightarrow q, r$$



## 5. Chứng minh mệnh đề (*tt*)

Bước 4: Nếu GTi còn dấu “ $\vee$ ” và KLj còn dấu “ $\wedge$ ” thì dòng vó được tách thành hai dòng con.

Ví dụ:  $p, q, r, (p \vee s) \rightarrow q, r \Leftrightarrow p, q, r, p \rightarrow q, r$  và  $p, q, r, s \rightarrow q, r$

Bước 5: Nếu một dòng được chứng minh: nếu tồn tại chung một mệnh đề ở cả 2 vế thì coi như đúng.

Ví dụ:  $p, q \rightarrow p$ : mệnh đề húng

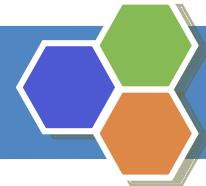
Bước 6:

+ Nếu một dòng không còn dấu liên kết tuyển và hội mà cả ở hai vế đều không có chung biến mệnh đề nào thì dòng ó không được chứng minh.

Ví dụ:  $p, \neg q \rightarrow q$

+ Một vấn đề được giải quyết một cách trọn vẹn nếu mọi dòng dẫn xuất từ dạng chuẩn được chứng minh.

Lưu ý: Từ bước 2 uến bước 4 không cần làm theo thứ tự.



## 5. Chứng minh mệnh đề (*tt*)

### Ví dụ: CMR

$$(p \vee \neg q) \wedge (\neg s \vee q) \wedge (r \vee s) \wedge (\neg p) \Rightarrow r \vee u$$

Giải:

$$(p \vee \neg q) \wedge (\neg s \vee q) \wedge (r \vee s) \wedge (\neg p) \Rightarrow r \vee u$$

$$\Leftrightarrow p \vee \neg q, \neg s \vee q, r \vee s, \neg p \Rightarrow r, u$$

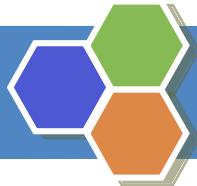
$$\Leftrightarrow p \vee \neg q, \neg s \vee q, r \vee s \Rightarrow r, u, p$$

Tách phép  $\vee$ :  $(p \vee \neg q)$  thành 2 dòng con

$$1: p, \neg s \vee q, r \vee s \Rightarrow r, u, p \text{ (đcm vì có } p \text{ ở hai phía)}$$

$$2: \neg q, \neg s \vee q, r \vee s \Rightarrow r, u, p$$

$$\Leftrightarrow \neg s \vee q, r \vee s \Rightarrow r, u, p, q$$



## 5. Chứng minh mệnh đề (*tt*)

Tách phép  $\vee$ :  $(\neg s \vee q)$  thành 2 dòng con

2.1:  $q, r \vee s \Rightarrow r, u, p, q$  (đcm vì có  $q$  ở hai phía)

2.2:  $\neg s, r \vee s \Rightarrow r, u, p, q$

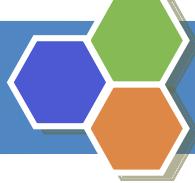
$$\Leftrightarrow r \vee s \Rightarrow r, u, p, q, s$$

Tách phép  $\vee$ :  $(r \vee s)$  thành 2 dòng con

2.2.1:  $r \Rightarrow r, u, p, q, s$  (đcm vì có  $r$  ở cả hai phía)

2.2.2:  $s \Rightarrow r, u, p, q, s$  (đcm vì có  $s$  ở cả hai phía)

Các dòng dẫn xuất từ dạng chuẩn ban đầu đều được chứng minh, vậy vấn đề được chứng minh.



## 5. Chứng minh mệnh đề (*tt*)

### ➤ Thuật giải Robinson

Thuật giải Robinson hành động dựa trên phương pháp chứng minh bằng phản chứng.

Bước 1: Đưa vấn đề về dạng chuẩn như sau:

$$GT_1, GT_2, \dots, GT_n \Rightarrow KL_1, KL_2, \dots, KL_m$$

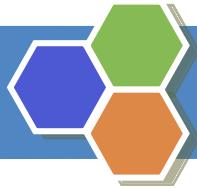
Trong đó các  $GT_i$  và  $KL_j$  được xây dựng từ các biến mệnh đề và các phép logic:  $\wedge, \vee, \neg$ .

Bước 2: Nếu  $GT_i$  có phép  $\wedge$  thì thay bằng dấu ",". Nếu  $KL_j$  có phép  $\vee$  thì thay bằng dấu ";".

Bước 3: Biến đổi dạng chuẩn ở Bước 1 về dạng sau:

$$GT_1, GT_2, \dots, GT_n, \neg KL_1, \neg KL_2, \dots, \neg KL_m$$

Bước 4: Nếu trong danh sách mệnh đề ở Bước 3 có mệnh đề ối ngẫu ( $p$  và  $\neg p$ ) thì mệnh đề được chứng minh. Ngược lại thì chuyển sang Bước 5.



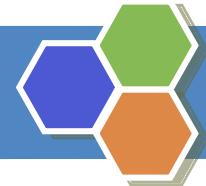
## 5. Chứng minh mệnh đề (*tt*)

Bước 5: Xây dựng một mệnh đề mới bằng cách tuyển một cặp mệnh đề trong danh sách mệnh đề. Nếu mệnh đề mới có các biến mệnh đề đối ngẫu thì loại bỏ các biến đó.

Bước 6: Thay thế hai mệnh đề vừa tuyển trong danh sách mệnh đề bằng mệnh đề mới, áp dụng phép hợp giải:

- i)  $p \wedge (\neg p \vee q) \Rightarrow q$
- ii)  $(p \vee q) \wedge (\neg p \vee r) \Rightarrow q \vee r$

Bước 7: Nếu không xây dựng được thêm một mệnh đề mới nào và trong danh sách mệnh đề không có hai mệnh đề nào đối ngẫu nhau thì vấn đề không được chứng minh. Nếu danh sách mệnh đề không còn mệnh đề nào (danh sách rỗng), vấn đề được chứng minh.



## 5. Chứng minh mệnh đề (*tt*)

Ví dụ: CMR

$$\neg p \vee q, (s \vee \neg q) \wedge (r \vee \neg s), p \wedge u \Rightarrow r, u$$

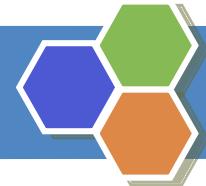
Giải:

$$\neg p \vee q, (s \vee \neg q) \wedge (r \vee \neg s), p \wedge u \Rightarrow r, u$$

$$\neg p \vee q, s \vee \neg q, r \vee \neg s, p, u \Rightarrow r, u$$

$$\neg p \vee q, s \vee \neg q, r \vee \neg s, p, u, \neg r, \neg u$$

ĐPCM



## 5. Chứng minh mệnh đề (*tt*)

Ví dụ: CMR

$$\neg p \vee q, (s \vee \neg q) \wedge (r \vee \neg s), p \Rightarrow r, u$$

Giải:

$$\neg p \vee q, (s \vee \neg q) \wedge (r \vee \neg s), p \Rightarrow r, u$$

$$\neg p \vee q, s \vee \neg q, r \vee \neg s, p \Rightarrow r, u$$

$$\neg p \vee q, s \vee \neg q, r \vee \neg s, p, \neg r, \neg u$$

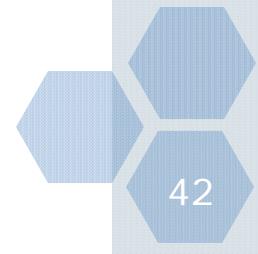
$$(\neg p \vee q, s \vee \neg q), r \vee \neg s, p, \neg r, \neg u$$

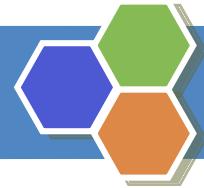
$$(\neg p \vee s, r \vee \neg s), p, \neg r, \neg u$$

$$(\neg p \vee r, p), \neg r, \neg u$$

$$r, \neg r, \neg u$$

ĐPCM





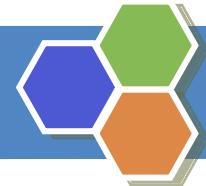
## 5. Chứng minh mệnh đề (*tt*)

Ví dụ: CMR

$$\neg p \vee s, q \vee t, \neg q \vee \neg p, s \Rightarrow t, \neg p$$

Giải:

$$\neg p \vee s, q \vee t, \neg q \vee \neg p, s \Rightarrow t, \neg p$$



## Bài tập Chương

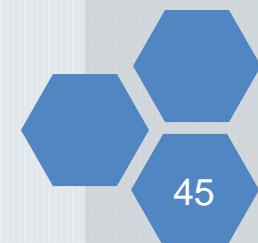
1. Giải bài toán tam giác (Mạng ngũ nghĩa)
2. Cài đặt thuật toán suy diễn tiến
3. Cài đặt thuật toán suy diễn lùi
4. Cài đặt thuật toán Vương Hạo
5. Cài đặt thuật toán Robinson
6. Tìm hiểu cơ chế biểu diễn và suy diễn tri thức trong hệ MYCIN, EMYCIN

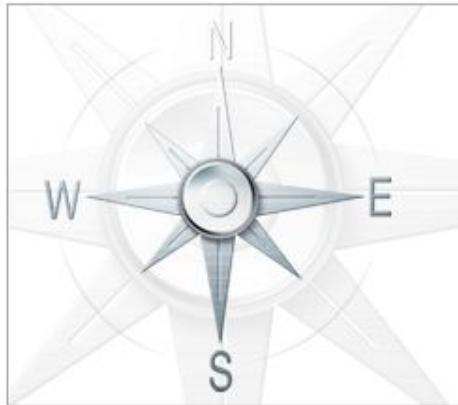


# End of Chapter 4



Bùi Đức Dương  
*Khoa Công nghệ Thông tin*





# Bài giảng Trí tuệ nhân tạo

Chương 5. Hệ học



Bùi Đức Dương

Khoa Công nghệ Thông tin



# 1. Giới thiệu

- ❖ Trong nhiều trường hợp, ta không có sẵn tri thức
  - Kỹ sư tri thức cần thu nhận tri thức từ các chuyên gia
  - Cần biết các luật mô tả lĩnh vực cụ thể
  - Bài toán không được biểu diễn tường minh theo luật, sự kiện hay quan hệ.
- ❖ Hai hướng tiếp cận cho hệ thống học:
  - Học từ ký hiệu
  - Học từ dữ liệu số



## 2. Các hình thức học

- ❖ Học vẹt
- ❖ Học bằng cách chỉ dẫn
- ❖ Học bằng quy nạp
- ❖ Học bằng tương tự
- ❖ Học dựa trên giải thích
- ❖ Học dựa trên tình huống
- ❖ Học không giám sát (Khám phá)



### 3. Cây định danh

- ❖ Xây dựng cây định danh dựa trên sự phân hoạch của các thuộc tính
- ❖ Phân hoạch:
  - Nút cha là thuộc tính được phân hoạch
  - Các nút con: Các giá trị phân biệt ứng với thuộc tính được phân hoạch.
- ❖ Là công cụ phổ biến trong một số ứng dụng



### 3. Cây định danh

#### ❖ Ví dụ 1:

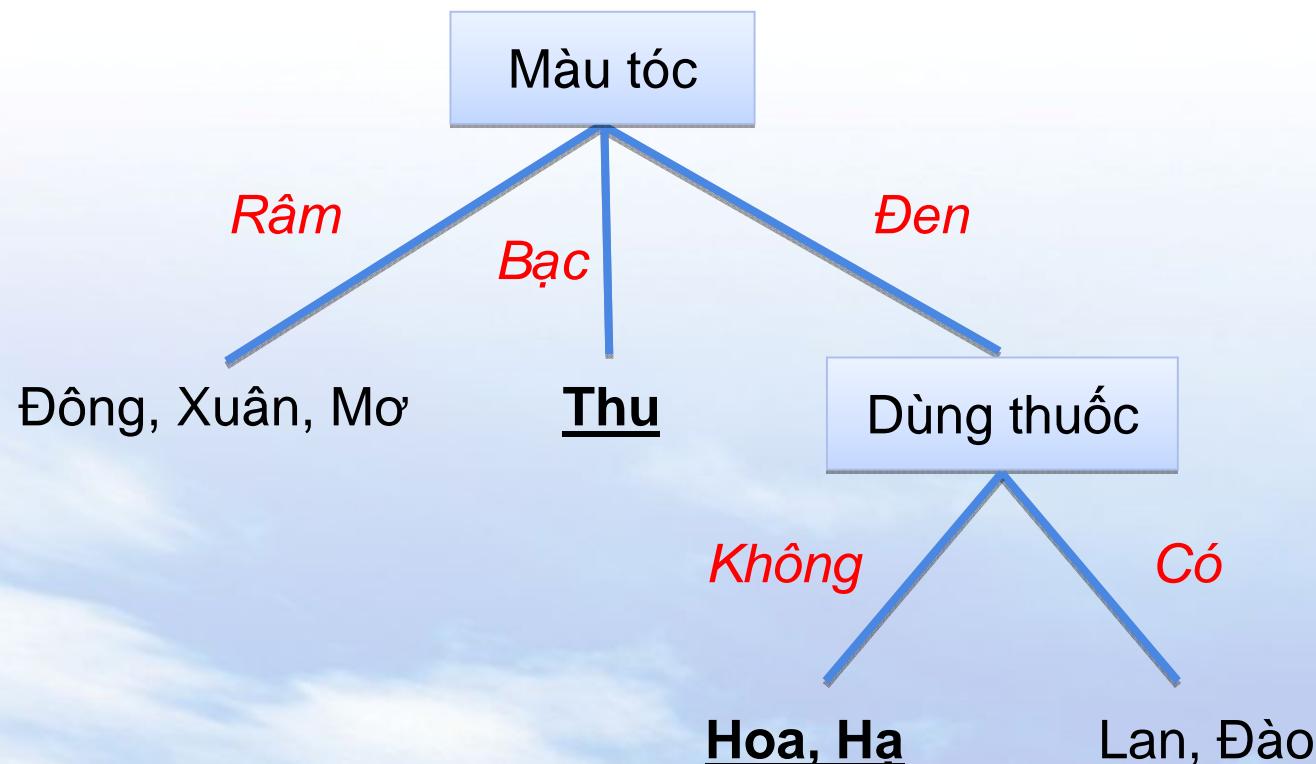
Bảng quan sát người đi tắm biển có/không bị rám nắng

TT	Tên	Màu tóc	Chiều cao	Cân nặng	Dùng thuốc	Kết quả
1	Hoa	Đen	TB	Nhẹ	Không	Bị rám
2	Lan	Đen	Cao	Vừa	Có	Không
3	Xuân	Râm	Thấp	Vừa	Có	Không
4	Hạ	Đen	Thấp	Vừa	Không	Bị rám
5	Thu	Bạc	TB	Nặng	Không	Bị rám
6	Đông	Râm	Cao	Nặng	Không	Không
7	Mơ	Râm	TB	Nặng	Không	Không
8	Đào	Đen	Thấp	Nhẹ	Có	Không



### 3. Cây định danh (tt)

- ❖ Một cây định danh tương ứng





### 3. Cây định danh (tt)

❖ Chuyển tri thức học từ dạng cây sang luật:

- Đi từ nút gốc đến nút lá;
- Lấy nút trong làm giả thiết;
- Lấy nút lá làm kết luật .

❖ Với cây định danh Ví dụ 1, ta có luật:

If (Màu tóc = Đen) and (Dùng thuốc = Không) then (Bị rám)



## 4. Thuật toán Quinlan

### ❖ Xây dựng các vector đặc trưng

$$V_A(j) = (T(j, r_1), T(j, r_2), \dots, T(j, r_n))$$

Trong đó:

- A: Thuộc tính dẫn xuất;  $r_1, r_2, \dots, r_n$ : Các thuộc tính mục tiêu
- $T(j, r_i) = T_{Aij}/T_{Aj}$
- ✓  $T_{Aij}$ : tổng số phần tử trong phân hoạch có thuộc tính A là j và thuộc tính mục tiêu là  $r_i$ .
- ✓  $T_{Aj}$ : tổng số phần tử trong phân hoạch có thuộc tính A là j.
- Lưu ý:  $T(j, r_1) + T(j, r_2) + \dots + T(j, r_n) = 1$

Vector đơn vị: Có 1 thành phần bằng 1; còn lại bằng 0.

### ❖ Thuộc tính được chọn để phân hoạch là thuộc tính có nhiều vector đơn vị nhất



### 3. Cây định danh

#### ❖ Ví dụ 1:

Bảng quan sát người đi tắm biển có/không bị rám nắng

TT	Tên	Màu tóc	Chiều cao	Cân nặng	Dùng thuốc	Kết quả
1	Hoa	Đen	TB	Nhẹ	Không	Bị rám
2	Lan	Đen	Cao	Vừa	Có	Không
3	Xuân	Râm	Thấp	Vừa	Có	Không
4	Hạ	Đen	Thấp	Vừa	Không	Bị rám
5	Thu	Bạc	TB	Nặng	Không	Bị rám
6	Đông	Râm	Cao	Nặng	Không	Không
7	Mơ	Râm	TB	Nặng	Không	Không
8	Đào	Đen	Thấp	Nhẹ	Có	Không



## 4. Thuật toán Quinlan (tt)

**Ví dụ 2.** Dựa trên bảng cho trong Ví dụ 1

- Thuộc tính **Màu tóc** (3 giá trị)

$$\begin{aligned}V_{\text{màu tóc}}(\text{Đen}) &= (T(\text{Đen}/\text{Bị rám}), T(\text{Đen}/\text{Không bị rám})) \\&= (2/4, 2/4) = (0.5, 0.5)\end{aligned}$$

$$\begin{aligned}V_{\text{màu tóc}}(\text{Râm}) &= (T(\text{Râm}/\text{Bị rám}), T(\text{Râm}/\text{Không bị rám})) \\&= (0/3, 3/3) = (0.0, 1.0)\end{aligned}$$

$$\begin{aligned}V_{\text{màu tóc}}(\text{Bạc}) &= (T(\text{Bạc}/\text{Bị rám}), T(\text{Bạc}/\text{Không bị rám})) \\&= (1/1, 0/1) = (1.0, 0.0)\end{aligned}$$

⇒ Số vector đơn vị: 02



## 4. Thuật toán Quinlan (tt)

- Thuộc tính **Chiều cao** (3 giá trị)

$$V_{\text{Chiều cao}}(\text{Cao}) = (0/2, 2/2) = (0.0, 1.0)$$

$$V_{\text{Chiều cao}}(\text{TB}) = (2/3, 1/3)$$

$$V_{\text{Chiều cao}}(\text{Thấp}) = (1/3, 2/3)$$

⇒ Số vector đơn vị: 01

- Thuộc tính **Cân nặng** (3 giá trị)

$$V_{\text{Cân nặng}}(\text{Nặng}) = (1/3, 2/3)$$

$$V_{\text{Cân nặng}}(\text{Vừa}) = (1/3, 2/3)$$

$$V_{\text{Cân nặng}}(\text{Nhẹ}) = (1/2, 1/2)$$

⇒ Số vector đơn vị: 0



## 4. Thuật toán Quinlan (tt)

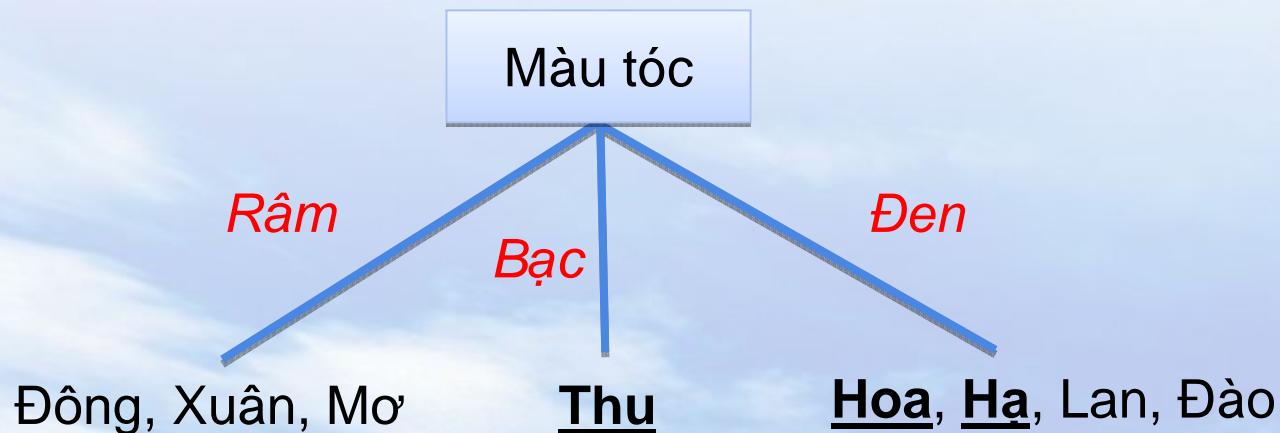
- Thuộc tính **Dùng thuốc** (2 giá trị)

$$V_{\text{Dùng thuốc}}(\text{Có}) = (0/3, 3/3) = (0.0, 1.0)$$

$$V_{\text{Dùng thuốc}}(\text{Không}) = (3/5, 2/5)$$

⇒ Số vector đơn vị: 01

Vậy, ta chọn thuộc tính màu tóc để phân hoạch!





## 4. Thuật toán Quinlan (tt)

Màu tóc	TT	Tên	Chiều cao	Cân nặng	Dùng thuốc	Kết quả
Đen	1	Hoa	TB	Nhẹ	Không	Bị rám
Đen	2	Lan	Cao	Vừa	Có	Không
Đen	3	Hạ	Thấp	Vừa	Không	Bị rám
Đen	4	Đào	Thấp	Nhẹ	Có	Không

➤ Thuộc tính **Chiều cao** (3 giá trị)

$$V_{\text{Chiều cao}}(\text{Cao}) = (0/1, 1/1) = (0.0, 1.0)$$

$$V_{\text{Chiều cao}}(\text{TB}) = (1/1, 0/1) = (1.0, 0.0)$$

$$V_{\text{Chiều cao}}(\text{Thấp}) = (1/2, 1/2)$$

⇒ Số vector đơn vị: 02



## 4. Thuật toán Quinlan (tt)

- Thuộc tính **Cân nặng** (2 giá trị)

$$V_{\text{Cân nặng}}(\text{Vừa}) = (1/2, 1/2)$$

$$V_{\text{Cân nặng}}(\text{Nhẹ}) = (1/2, 1/2)$$

⇒ Số vector đơn vị: 0

- Thuộc tính **Dùng thuốc** (2 giá trị)

$$V_{\text{Dùng thuốc}}(\text{Có}) = (0/2, 2/2) = (0.0, 1.0)$$

$$V_{\text{Dùng thuốc}}(\text{Không}) = (2/2, 0/2) = (1.0, 0.0)$$

⇒ Số vector đơn vị: 02

Có 2 thuộc tính cùng có 2 vector đơn vị:

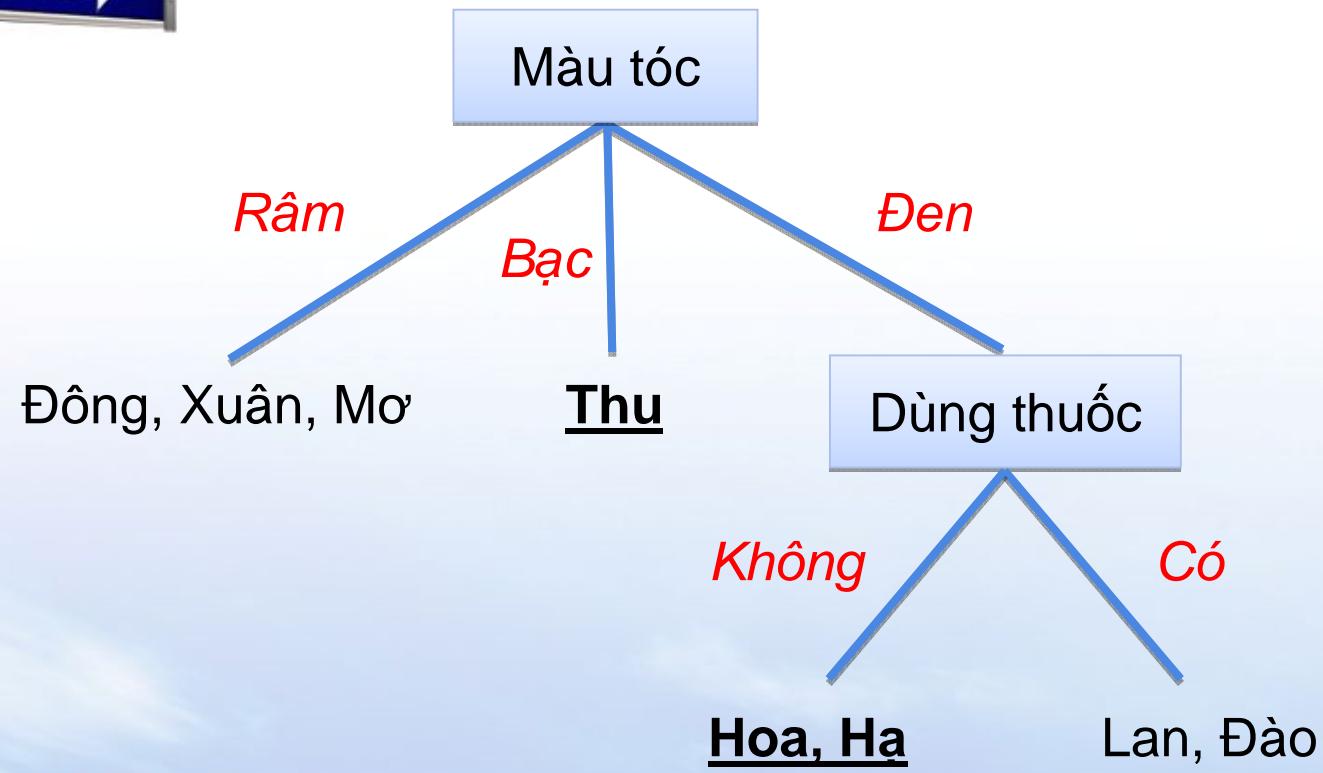
Chiều cao: Tỷ lệ 2/3

Dùng thuốc: Tỷ lệ 2/2

Chọn thuộc tính **Dùng thuốc** để phân hoạch

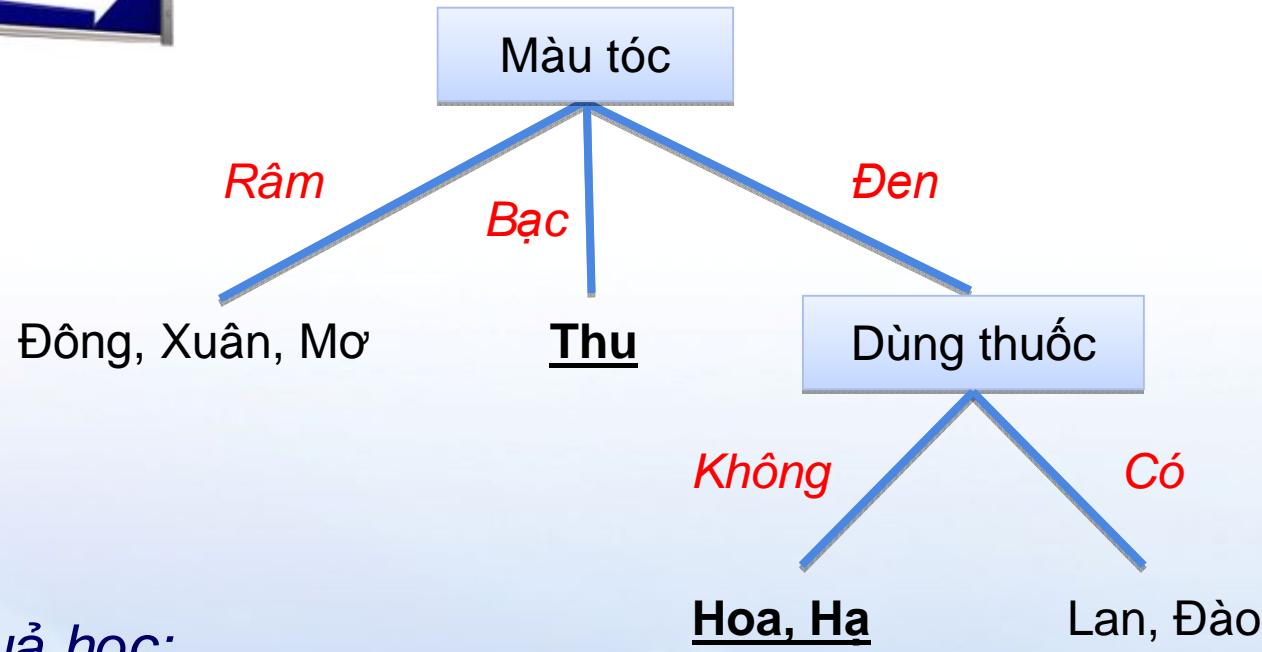


## 4. Thuật toán Quinlan (tt)



Cây định danh kết quả

## 4. Thuật toán Quinlan (tt)



Kết quả học:

- ✓ If (**Màu tóc = Râm**) then (**Bị rám = Không**)
- ✓ If (**Màu tóc = Bạc**) then (**Bị rám = Có**)
- ✓ If (**Màu tóc = Đen**) AND (**Dùng thuốc = Không**) then (**Bị rám = Có**)
- ✓ If (**Màu tóc = Đen**) AND (**Dùng thuốc = Có**) then (**Bị rám = Không**)



## 5. Học theo độ bất định

### ❖ Độ bất định (entropy) của thuộc tính X

$$E(X) = \sum_b \frac{n_b}{n_t} \sum_c -\frac{n_{bc}}{n_b} \log_a \frac{n_{bc}}{n_b}$$

Trong đó:

$n_b$ : số mẫu nhánh b;  $n_t$ : tổng số mẫu;  $n_{bc}$ : số mẫu trong nhánh b của lớp c; a: Số lượng giá trị của thuộc tính mục tiêu.

### ❖ Thuật toán

Repeat

- Tính độ bất định cho mọi thuộc tính
- Chọn thuộc tính có hộ bất định nhỏ nhất để phân hoạch Until (Hết thuộc tính phân hoạch)



## 5. Học theo độ bất định (tt)

### ❖ Ví dụ 3. Dựa trên bảng cho trong Ví dụ 1

- Thuộc tính **Màu tóc** (3 giá trị)
  - ✓ Tập tóc đen: 2/4 người bị rám; 2/4 người không bị rám
  - ✓ Tập tóc bạc: 1/1 người bị rám; 0/1 người không bị rám
  - ✓ Tập tóc râm: 0/3 người bị rám; 3/3 người không bị rám

$$\begin{aligned}\Rightarrow E(\text{Màu tóc}) &= 4/8(-2/4\log_2 2/4 - 2/4\log_2 2/4) \\ &\quad + 1/8(-1/1\log_2 1/1 - 0/1\log_2 0/1) \\ &\quad + 3/8(-0/3\log_2 0/3 - 3/3\log_2 3/3) \\ &= 0.5 + 0 + 0 = 0.5.\end{aligned}$$

Tương tự ta có:

$$E(\text{Chiều cao}) = 0.69; E(\text{Cân nặng}) = 0.94; E(\text{Dùng thuốc}) = 0.61.$$

Vậy, ta chọn **Màu tóc** làm thuộc tính phân hoạch



## 5. Học theo độ bất định (tt)

- Ở bước tiếp theo, ta sẽ tính được:
  - ✓  $E(\text{Chiều cao})=0.5$ ;
  - ✓  $E(\text{Cân nặng})=1.0$ ;
  - ✓  $E(\text{Dùng thuốc})=0.0$ ;

Vậy ta chọn **Dùng thuốc** làm thuộc tính phân hoạch và được cây định danh kết quả.



## 5. Học theo độ bất định (tt)

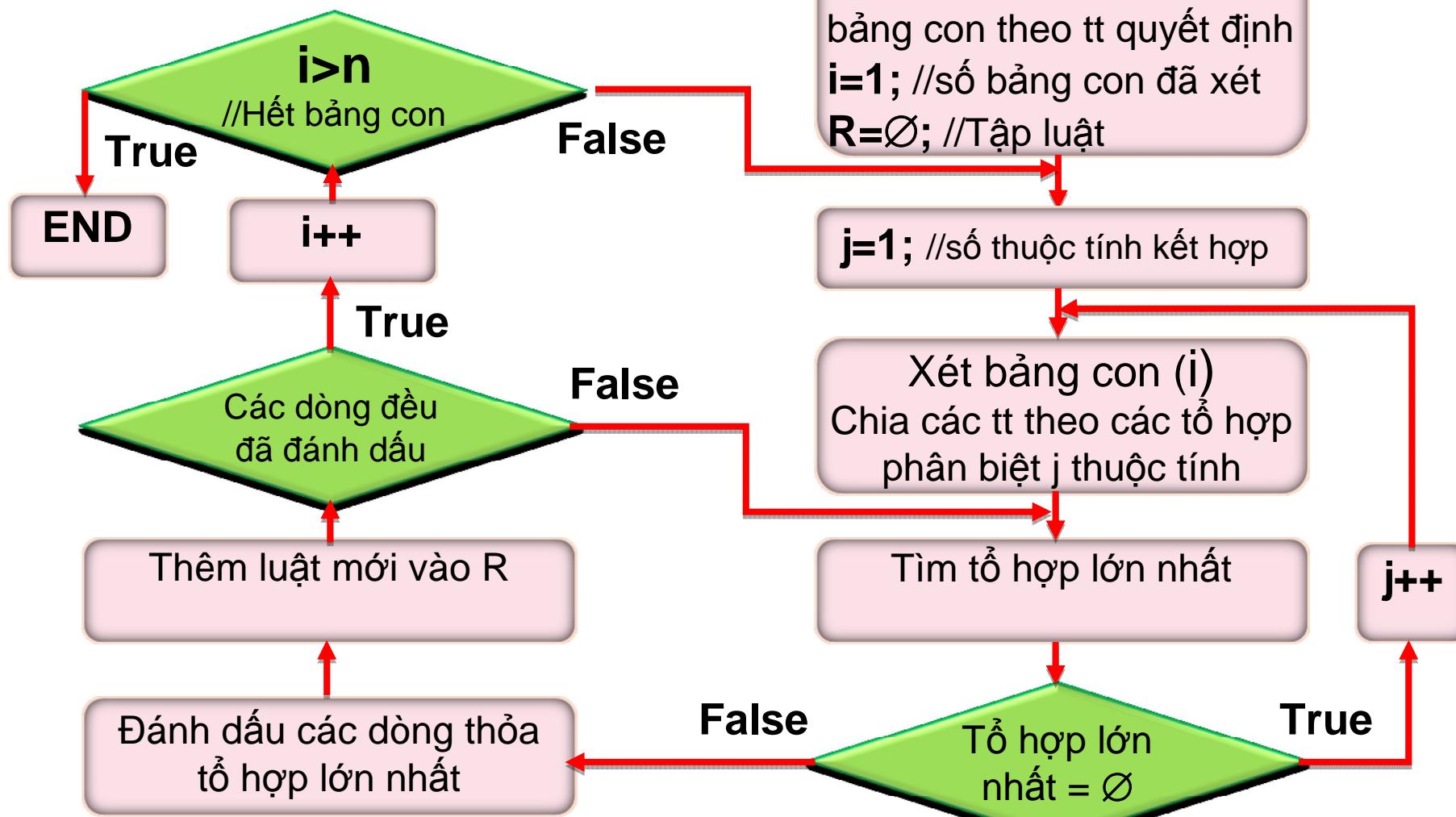
### ❖ Ví dụ 4:

Play golf dataset

Independent variables				Dep. var
OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
overcast	83	78	FALSE	Play
rain	70	96	FALSE	Play
rain	68	80	FALSE	Play
rain	65	70	TRUE	Don't Play
overcast	64	65	TRUE	Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
rain	75	80	FALSE	Play
sunny	75	70	TRUE	Play
overcast	72	90	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play



## 6. Thuật giảiILA (Inductive Learning A)





## 6. Thuật giảiILA (tt)

### ❖ Ví dụ 5:

Bảng quan sát người đi tắm biển cho trong Ví dụ 1

TT	Màu tóc	Chiều cao	Cân nặng	Dùng thuốc	Kết quả
1	Đen	TB	Nhẹ	Không	Bị rám
2	Đen	Cao	Vừa	Có	Không
3	Râm	Thấp	Vừa	Có	Không
4	Đen	Thấp	Vừa	Không	Bị rám
5	Bạc	TB	Nặng	Không	Bị rám
6	Râm	Cao	Nặng	Không	Không
7	Râm	TB	Nặng	Không	Không
8	Đen	Thấp	Nhẹ	Có	Không



## 6. Thuật giảiILA (tt)

➤ **Bước 1: Chia thành 2 bảng con**

<b>Bảng</b>	<b>TT</b>	<b>Màu tóc</b>	<b>Chiều cao</b>	<b>Cân nặng</b>	<b>Dùng thuốc</b>	<b>Kết quả</b>
1	1	Đen	TB	Nhẹ	Không	Bị rám
	2	Đen	Thấp	Vừa	Không	Bị rám
	3	Bạc	TB	Nặng	Không	Bị rám

<b>Bảng</b>	<b>TT</b>	<b>Màu tóc</b>	<b>Chiều cao</b>	<b>Cân nặng</b>	<b>Dùng thuốc</b>	<b>Kết quả</b>
2	1	Đen	Cao	Vừa	Có	Không
	2	Râm	Thấp	Vừa	Có	Không
	3	Râm	Cao	Nặng	Không	Không
	4	Râm	TB	Nặng	Không	Không
	5	Đen	Thấp	Nhẹ	Có	Không



## 6. Thuật giảiILA (tt)

➤ Xét bảng con i=1;

Bảng	TT	Màu tóc	Chiều cao	Cân nặng	Dùng thuốc	Kết quả
1	1	Đen	TB	Nhẹ	Không	Bị rám
	2	Đen	Thấp	Vừa	Không	Bị rám
	3	Bạc	TB	Nặng	Không	Bị rám

✓ j=1 (1 thuộc tính):

Có 4 tổ hợp {Màu tóc}, {Chiều cao}, {Cân nặng} và {Dùng thuốc}

{Màu tóc}	{Chiều cao}	{Cân nặng}	{Dùng thuốc}
Bạc (1)	∅	∅	∅

✓ Tổ hợp lớn nhất: {Màu tóc}= “Bạc”

R = R  $\cup$  {IF **Màu tóc = Bạc** THEN **Kết quả = Bị rám**}



## 6. Thuật giảiILA (tt)

- ✓ Đánh dấu dòng có {Màu tóc}= “Bạc”

Bảng	TT	Màu tóc	Chiều cao	Cân nặng	Dùng thuốc	Kết quả
1	1	Đen	TB	Nhẹ	Không	Bị rám
	2	Đen	Thấp	Vừa	Không	Bị rám
	3	Bạc	TB	Nặng	Không	Bị rám

- ✓  $R = R \cup \{\text{IF } \text{Màu tóc} = \text{Bạc } \text{THEN } \text{Kết quả} = \text{Bị rám}\}$

{Màu tóc}	{Chiều cao}	{Cân nặng}	{Dùng thuốc}
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

- ✓ Tổ hợp lớn nhất:  $\emptyset$ ;
- ✓ j=2 (2 thuộc tính):
  - {Màu tóc, Chiều cao}, {Màu tóc, Cân nặng}; {Màu tóc, Dùng thuốc}
  - {Chiều cao, Cân nặng}, {Chiều cao, <sub>25</sub>Dùng thuốc}; {Cân nặng, Dùng thuốc}



## 6. Thuật giảiILA (tt)

{Màu tóc, Chiều cao}	{Màu tóc, Cân nặng}	{Màu tóc, Dùng thuốc}
Đen, TB (1)	∅	Đen, Không (2)
{Chiều cao, Cân nặng}	{Chiều cao, Dùng thuốc}	{Cân nặng, Dùng thuốc}
TB, Nhẹ (1)	TB, Không (1)	Nhẹ, Không (1) Vừa, Không (1)

- ✓ Tổ hợp lớn nhất: {Màu tóc, Dùng thuốc}= “Đen, Không”
- ✓ Đánh dấu dòng có {Màu tóc, Dùng thuốc}= “Đen, Không”

Bảng	TT	Màu tóc	Chiều cao	Cân nặng	Dùng thuốc	Kết quả
1	1	Đen	TB	Nhẹ	Không	Bị rám
	2	Đen	Thấp	Vừa	Không	Bị rám
	3	Bạc	TB	Nặng	Không	Bị rám

- ✓  $R = R \cup \{IF \text{ } Màu \text{ tóc} = \text{Đen} \text{ AND } Dùng \text{ thuốc} = \text{Không}$   
THEN  $\text{Kết quả} = \text{Bị rám}\}$



## 6. Thuật giảiILA (tt)

➤ Xét bảng con i=2;

Bảng	TT	Màu tóc	Chiều cao	Cân nặng	Dùng thuốc	Kết quả
2	1	Đen	Cao	Vừa	Có	Không
	2	Râm	Thấp	Vừa	Có	Không
	3	Râm	Cao	Nặng	Không	Không
	4	Râm	TB	Nặng	Không	Không
	5	Đen	Thấp	Nhẹ	Có	Không

✓ j=1 (1 thuộc tính):

{Màu tóc}	{Chiều cao}	{Cân nặng}	{Dùng thuốc}
Râm (3)	Cao (2)	∅	Có (3)

✓ Tổ hợp lớn nhất: {Màu tóc} = “Râm” và {Dùng thuốc} = “Có”

Chọn tổ hợp đầu tiên: {Màu tóc} = “Râm”



## 6. Thuật giảiILA (tt)

- ✓ Đánh dấu các dòng có : {Màu tóc}= “Râm”

Bảng	TT	Màu tóc	Chiều cao	Cân nặng	Dùng thuốc	Kết quả
2	1	Đen	Cao	Vừa	Có	Không
	2	Râm	Thấp	Vừa	Có	Không
	3	Râm	Cao	Nặng	Không	Không
	4	Râm	TB	Nặng	Không	Không
	5	Đen	Thấp	Nhẹ	Có	Không

- ✓  $R = R \cup \{IF \text{ } \text{Màu tóc} = \text{Râm} \text{ THEN } \text{Kết quả} = \text{Không}\}$

{Màu tóc}	{Chiều cao}	{Cân nặng}	{Dùng thuốc}
Ø	Cao (1)	Ø	Có (2)

- ✓ Tổ hợp lớn nhất: {Dùng thuốc}="Có"

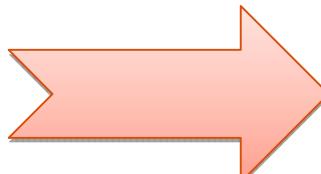


## 6. Thuật giảiILA (tt)

- ✓ Đánh dấu các dòng có : {Dùng thuốc}= “Có”

Bảng	TT	Màu tóc	Chiều cao	Cân nặng	Dùng thuốc	Kết quả
2	1	Đen	Cao	Vừa	Có	Không
	2	Râm	Thấp	Vừa	Có	Không
	3	Râm	Cao	Nặng	Không	Không
	4	Râm	TB	Nặng	Không	Không
	5	Đen	Thấp	Nhẹ	Có	Không

- ✓  $R = R \cup \{ \text{IF } \text{Dùng thuốc= Có THEN Kết quả =Không} \}$
- Đánh dấu hết dòng
- Đã xét hết các bảng con



Kết thúc

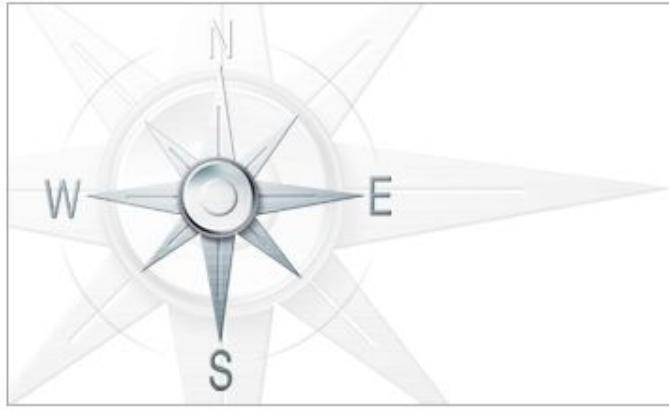


# Bài tập Chương

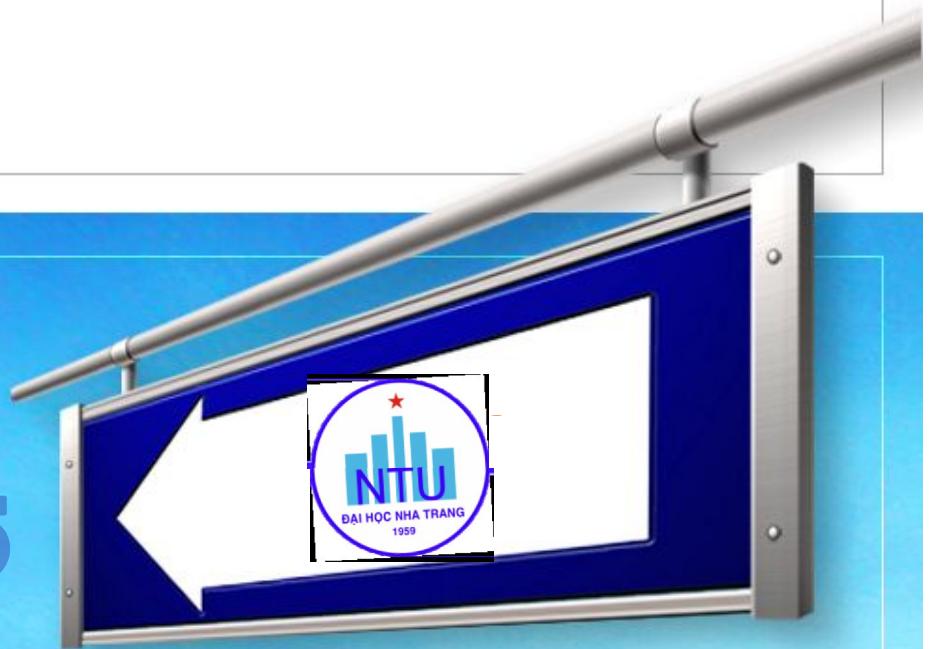
1. Đề xuất 3 bảng dữ liệu quan sát
2. Áp dụng các thuật toán để rút ra các luật từ các bảng dữ liệu đề xuất trong mục 1.

**\* Nghiên cứu chuyên sâu:**

1. Thuật giải di truyền
2. Mạng nơron nhân tạo
3. Hệ mờ và các ứng dụng



# END OF CHAPTER 5



**Bùi Đức Dương**  
*Khoa Công nghệ Thông tin*