

Đề Cương Nhập Môn Trí Tuệ Nhân Tạo PTIT

Cấu trúc đề thi trí tuệ nhân tạo PTIT(2020)

Câu 1: 2 điểm

- a. 1 trong 4 thuật toán tìm kiếm mù (DFS, BFS, UCS, IDS)
- b. 1 trong 3 thuật toán tìm kiếm có thông tin (A*, tham lam, IDA*)

Câu 2: 2 điểm

- a. Chuyển các câu sang logic vị từ + chuẩn hóa dạng CNF (Clause Form)
- b. Sử dụng 1 trong 4 thủ tục suy diễn để chứng minh câu truy vấn (suy diễn tiến , suy diễn lùi, phép giải , phép giải + phản chứng)

Câu 3: 3 điểm

- Xây dựng mạng bayes
- Tính xác xuất nhân quả (chuẩn đoán) (dạng $P(A|B)$)
- Dạng $P(A|B,C)$

Câu 4: 3 điểm

- Tìm nút gốc cho cây quyết định sử dụng thuật toán ID3
- Thuật toán k láng giềng hoặc phương pháp phân loại Bayes đơn giản.

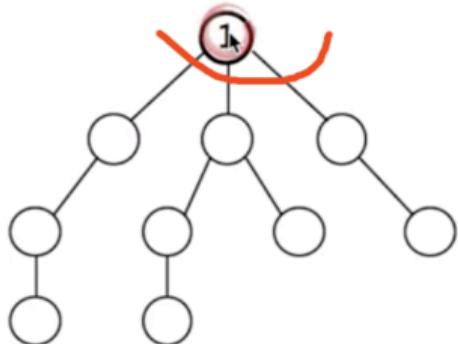
Phần 1: Các thuật toán tìm kiếm

I. Tìm kiếm mù

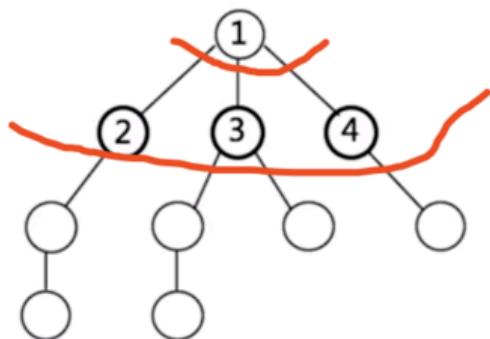
1. Tìm kiếm theo chiều rộng (BFS)

***Nguyên tắc:** Trong số những nút biên chọn những nút nông nhất (gần nút gốc nhất) để mở rộng

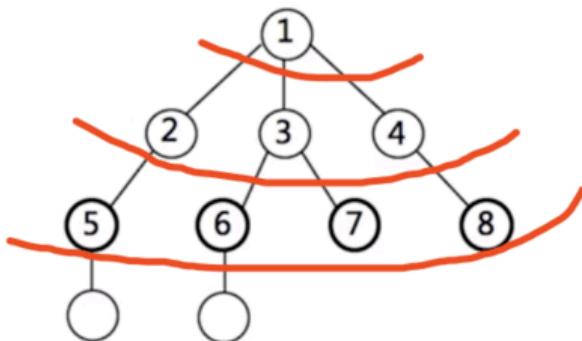
How the BFS would work



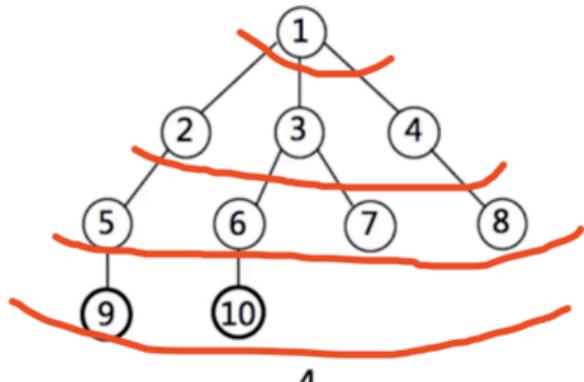
1



2



3



4

* Thuật toán tìm kiếm theo chiều rộng (BFS)

BFS(Q, S, G, P)

(Q: không gian trạng thái, S: trạng thái bắt đầu, G: đích, P: hành động)

Đầu vào: bài toán tìm kiếm

Đầu ra: trạng thái đích

Khởi tạo: tập các nút biên (nút mở) O = S

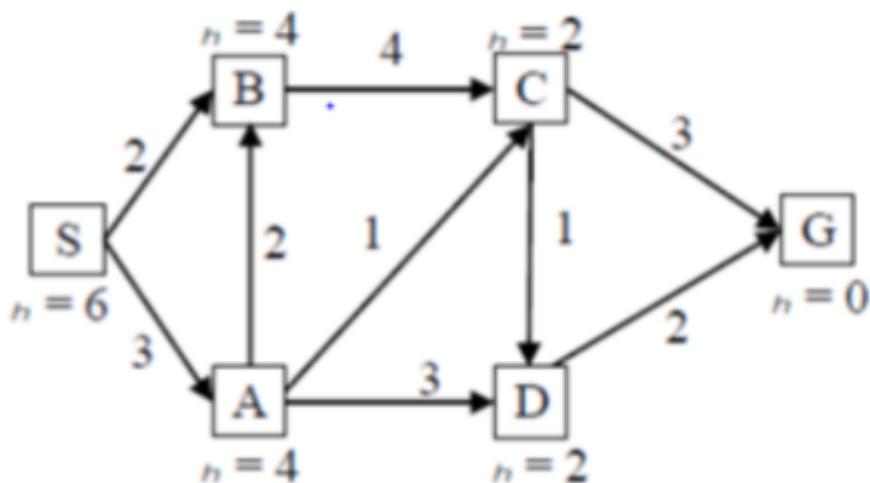
While(O ≠ ∅) do

1. lấy nút đầu tiên n khỏi O
2. if n ∈ G, return (đường đi tới n)
3. thêm P(n) vào đuôi O

Return: không tìm được lời giải

*Xử lý nút lặp: KHÔNG đưa nút đã được duyệt (mở rộng) hoặc đang thuộc tập biên O vào tập biên.

Ví dụ 1.1:



BFS

STT	Nút được mở rộng	Tập biên O
-----	------------------	------------

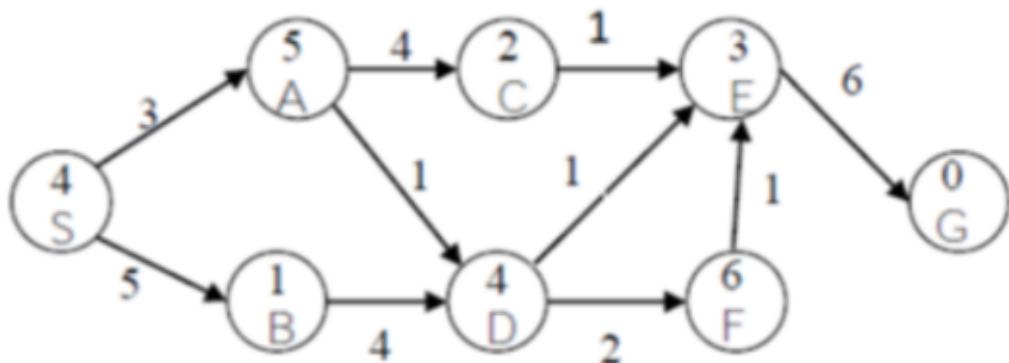
0		S
1	S	As, Bs
2	A	Bs, Ca, Da
3	B	Ca, Da
4	C	Da, Gc
5	D	Gc
6	G	Đích

Đường đi: G←C← A← S

Độ sâu: d = 3

Ví dụ 1.2:

Câu 1.3:



BFS

STT	Nút được mở rộng	Tập biên O
0		S
1	S	As, Bs
2	A	Bs, Ca, Da
3	B	Ca, Da
4	C	Da, Ec
5	D	Ec, Fd

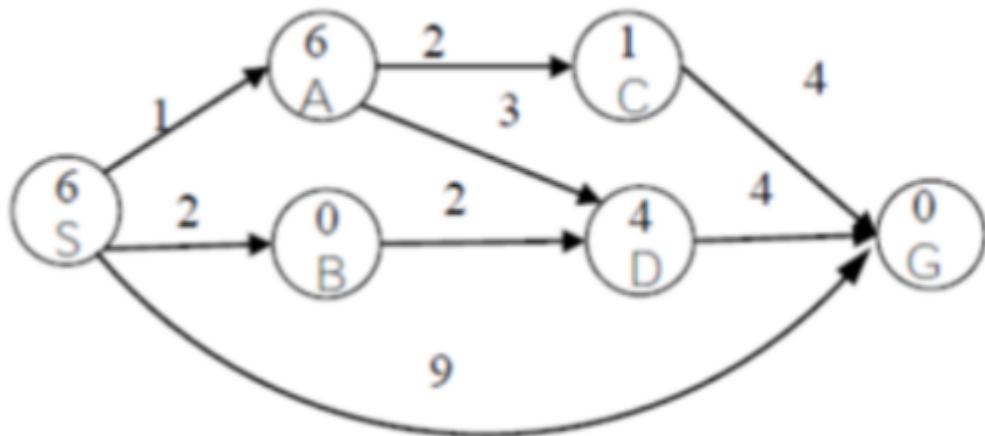
6	E	Fd, Ge
7	F	Ge
8	G	Đích

Đường đi: G← E← C← A← S

Độ sâu d = 4

Ví dụ: 1.3

Câu 1.4:



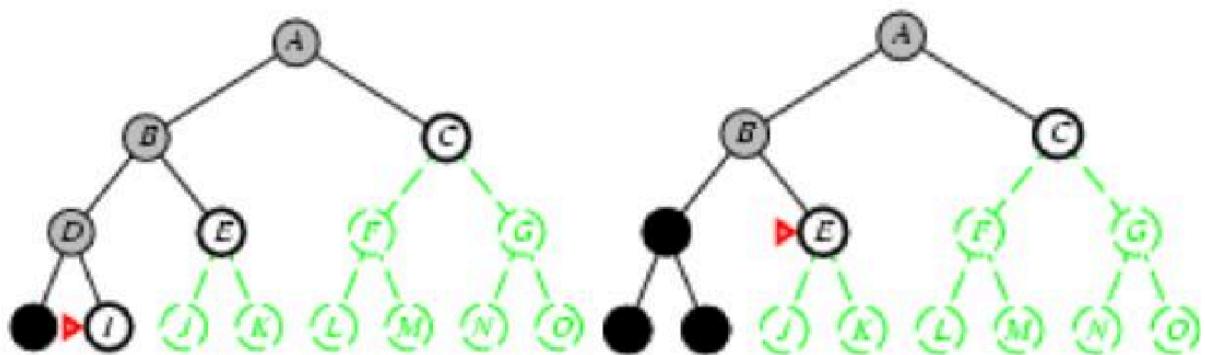
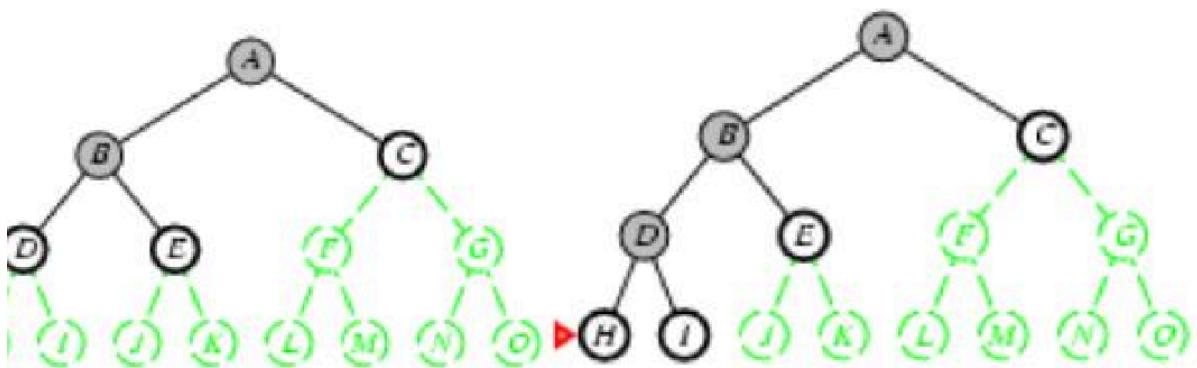
STT	Nút được mở rộng	Tập biên O
0		S
1	S	As,Bs,Gs
2	A	Bs,Gs,Ca,Da
3	B	Gs,Ca,Da
4	G	Đích

Đường đi: G← S

Độ sâu: d = 1

2. Tìm kiếm theo chiều sâu (DFS)

- **Nguyên tắc:** trong số những nút biên, lựa chọn nút sâu nhất (xa gốc nhất) để mở rộng



* Thuật toán tìm kiếm theo chiều sâu (DFS)

DFS(Q, S, G, P)

(Q: không gian trạng thái, S: trạng thái bắt đầu, G: đích, P: hành động)

Đầu vào: bài toán tìm kiếm

Đầu ra: trạng thái đích

Khởi tạo: tập các nút biên (nút mở) O = S

While(O ≠ ∅) do

1. lấy nút đầu tiên n khỏi O
2. if n ∈ G, return (đường đi tới n)
3. thêm P(n) vào đầu O

Return: không tìm được lời giải

*Xử lý nút lặp:

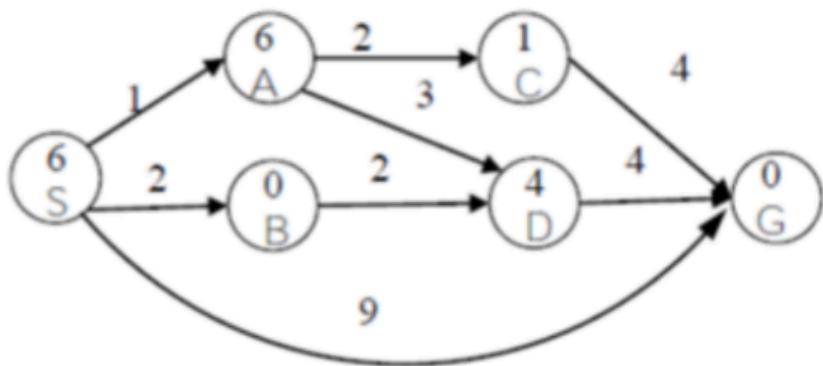
- Có lặp các nút ĐANG THUỘC TẬP BIÊN
- Không đưa các nút ĐÃ ĐƯỢC DUYỆT vào tập biên

▶ DFS

- **Có:** Việc đưa nút lặp vào ngăn xếp sẽ làm thay đổi thứ tự duyệt các nút trong ngăn xếp (thay đổi nhánh tìm kiếm), và thay đổi nghiệm của bài toán.
- Tuy nhiên nếu đây là một nút **đã được duyệt** rồi thì sẽ **không** đưa vào ngăn xếp nữa.

Ví dụ 2.1:

Câu 1.4:



DFS

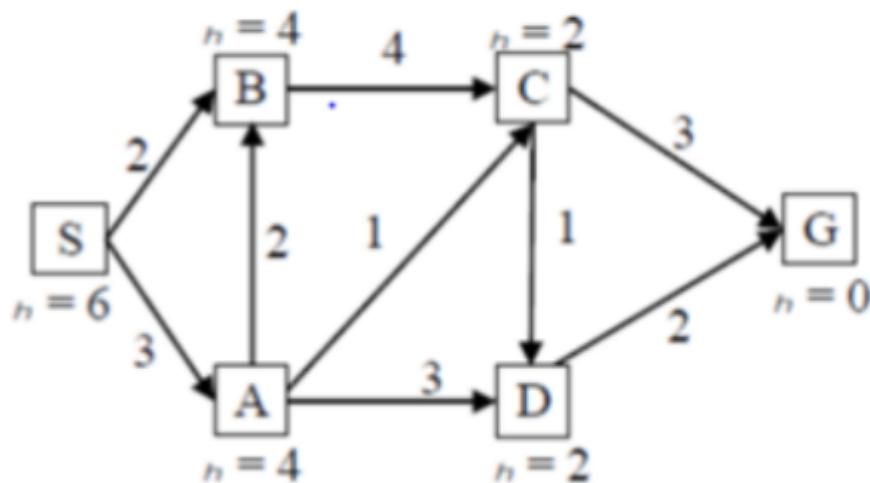
STT	Nút được mở rộng	Tập biên O
0		S
1	S	As,Bs,Gs
2	A	Ca,Da,Bs,Gs
3	C	Gc,Da,Bs,Gs
4	G	Đích

Đường đi: G←C←A←S

Độ sâu: d = 3

Ví dụ 2.2:
DFS

Câu 1.1:



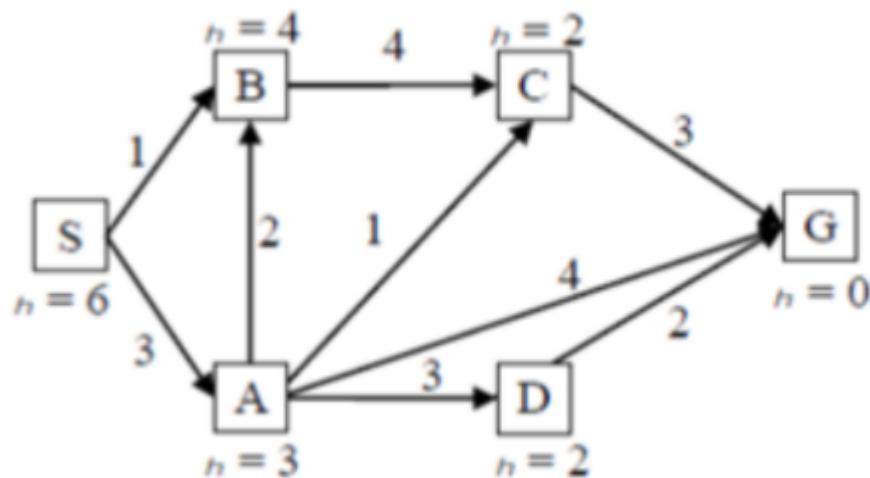
STT	Nút được mở rộng	Tập biên O
0		S
1	S	As,Bs
2	As	Ba,Ca,Da,Bs
3	Ba	Cb,Ca,Da,Bs
4	Cb	Dc,Gc,Ca,Da,Bs
5	Dc	Gd,Gc,Ca,Da,Bs
6	Gd	Đích

Đường đi: $G \leftarrow D \leftarrow C \leftarrow B \leftarrow A \leftarrow S$

Độ sâu d = 5

Ví dụ 2.3:

Câu 1.2:



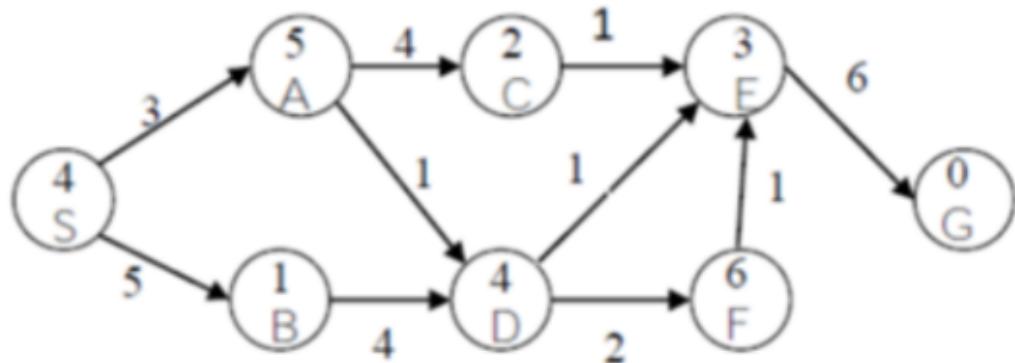
DFS

STT	Nút được mở rộng	Tập biên O
0		S
1	S	As, Bs
2	As	Ba,Ca,Da,Ga,Bs
3	Ba	Cb,Ca,Da,Ga,Bs
4	Cb	Gc,Ca,Da,Ga,Bs
5	Gc	Đích

Đường đi: $G \leftarrow C \leftarrow B \leftarrow A \leftarrow S$

Ví dụ 2.4:

Câu 1.3:



DFS

STT	Nút được mở	Tập biên O
0		S
1	S	As,Bs
2	A	Ca,Da,Bs
3	C	Ec,Da,Bs
4	E	Ge,Da,Bs
5	G	Đích

Đường đi: G← E← C← A← S

Độ sâu d = 4

3. Tìm kiếm theo giá thành thông nhất (UCS)

- **Phương pháp:** chọn nút có **giá thành nhỏ nhất** để mở rộng trước thay vì chọn nút nông nhất như trong BFS

****g(m) min**

***Xử lí nút lặp:** Nếu nút có chi phí tốt hơn:

+**ĐƯA LẠI** vào danh sách nếu **ĐÃ MỞ RỘNG**

+**CẬP NHẬT** thay nút cũ nếu **ĐANG TRONG DANH SÁCH**

► UCS

- Trong trường hợp nút lặp có giá thành (chi phí) tốt hơn, nó sẽ được **đưa lại danh sách** (nếu đã phát triển rồi) hoặc **cập nhật thay nút cũ** có giá thành kém hơn (nếu đang trong danh sách)

Ví dụ 3.1:

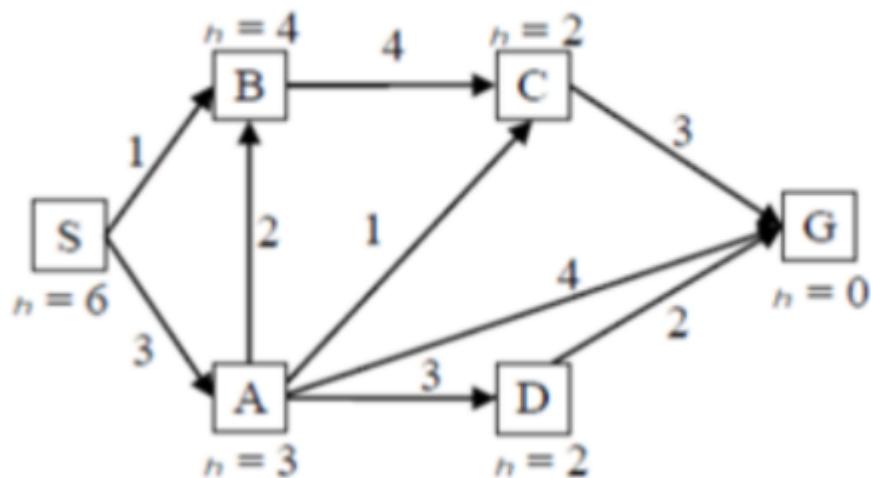
STT	Nút được mở rộng	Tập biên O
0		S(0)
1	S	As(3),Bs(2)
2	Bs	As(3),Cb(6)
3	As	Ca(4),Da(6)
4	Ca	Dc(5),Gc(7)
5	Dc	Gc(7)
6	Gc	Đích

Đường đi: G← C← A← S

chi phí c = 7

Ví dụ 3.2:

Câu 1.2:



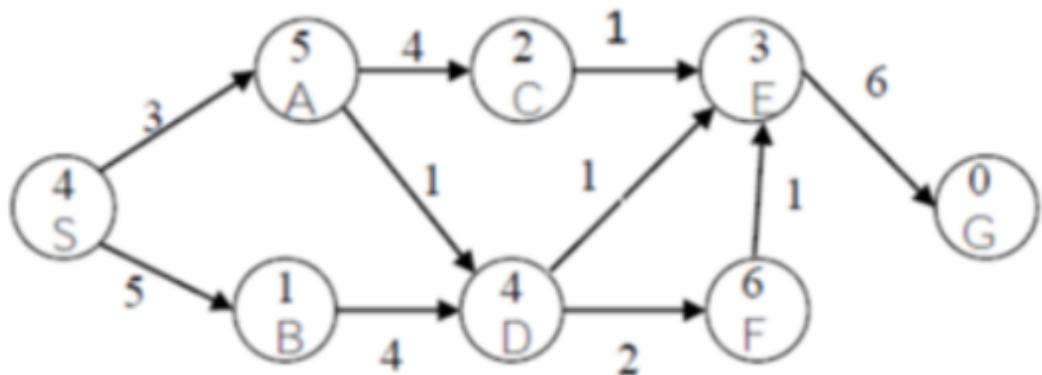
STT	Nút được mở rộng	Tập biên O
0		S
1	S	As(3),Bs(1)
2	Bs	As(3),Cb(5)
3	As	Ca(4),Da(6),Ga(7)
4	Ca	Da(6),Ga(7)
5	Da	Ga(7)
6	G	Đích

Đường đi: $G \leftarrow A \leftarrow S$

chi phí : $c = 7$

Ví dụ 3.3:

Câu 1.3:



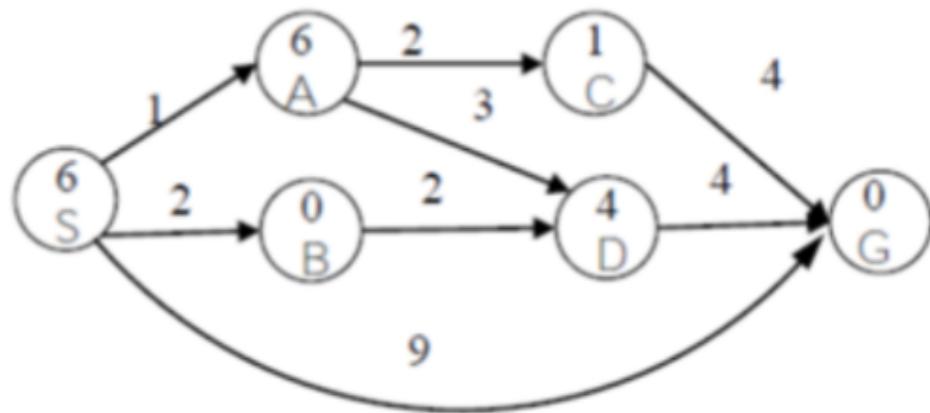
STT	Nút được mở rộng	Tập biên O
0		S
1	S	As(3),Bs(5)
2	As	Bs(5),Ca(7),Da(4)
3	Da	Bs(5),Ca(7),Ed(5),Fd(6)
4	Bs	Ca(7),Ed(5),Fd(6)
5	Ed	Ca(7),Fd(6),Ge(11)
6	Fd	Ca(7),Ge(11)
7	Ca	Ge(11)
8	Ge	Đích

Đường đi: $G \leftarrow E \leftarrow D \leftarrow A \leftarrow S$

chi phí c = 11

Ví dụ 3.4:

Câu 1.4:



STT	Nút được mở	Tập biên O
0		S
1	S	As(1),Bs(2),Gs(9)
2	As	Bs(2),Gs(9),Ca(3),Da(4)
3	Bs	Gs(9),Ca(3),Da(4)
4	Ca	Gc(7),Da(4)
5	Da	Gc(7)
6	Gc	Đích

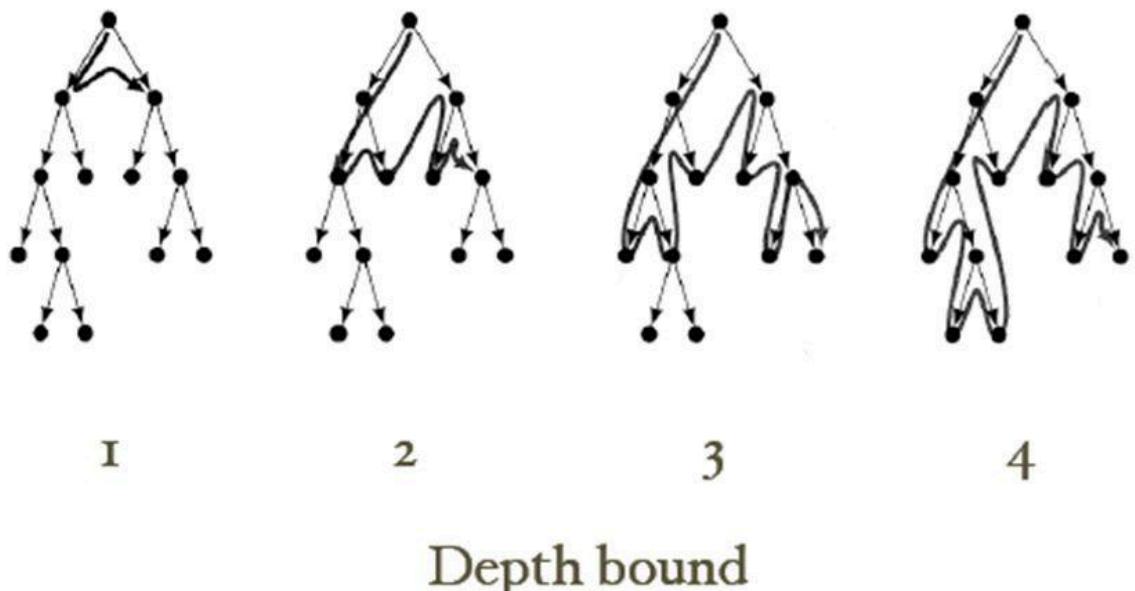
Đường đi: $G \leftarrow C \leftarrow A \leftarrow S$, chi phí c = 7

4. Tìm kiếm sâu dần (IDS)

****IDS từng độ sâu**

- ▶ **Phương pháp:** tìm theo DFS nhưng không bao giờ mở rộng các nút có độ sâu quá một giới hạn nào đó. Giới hạn độ sâu sẽ được tăng dần cho đến khi tìm được lời giải.

Iterative Deepening Search (IDS)



* Thuật toán tìm kiếm sâu dần (IDS)

IDS(Q, S, G, P)

(Q: không gian trạng thái, S: trạng thái bắt đầu, G: đích, P: hành động)

Đầu vào: bài toán tìm kiếm

Đầu ra: trạng thái đích

Khởi tạo: tập các nút biên (nút mở) $O = S$

$c = 0$ là độ sâu hiện thời

While(1) do

1. While($O \neq \emptyset$) do

a. lấy nút đầu tiên n khỏi O

b. if $n \in G$, return (đường đi tới n)

c. if $\text{depth}(n) < c$ then

thêm $P(n)$ vào đầu O

2. $c++$; $O = S$

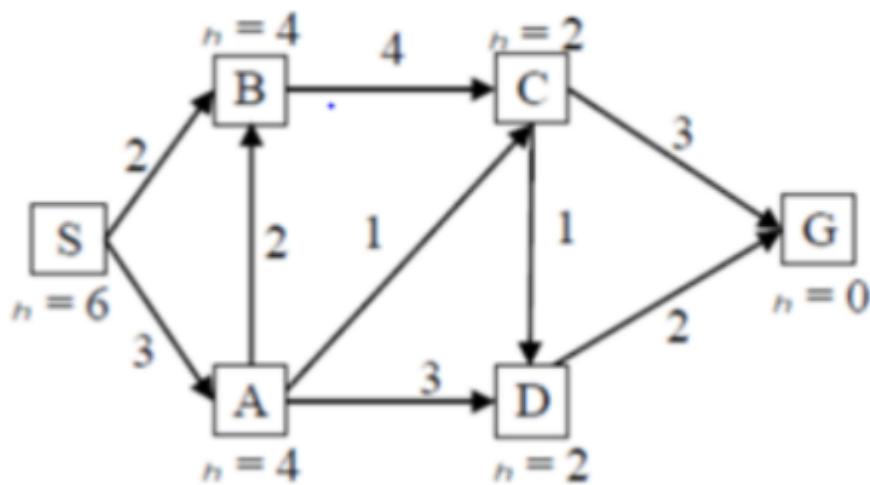
***Xử lý nút lặp:** Vì bản chất là DFS nên ở mỗi độ sâu c :

+**KHÔNG** đưa các nút **ĐÃ DUYỆT** vào tập biên

+**LẶP** các nút **ĐANG THUỘC TẬP BIÊN**

Ví dụ 4.1:

Câu 1.1:



STT	Nút được mở rộng	Tập biên O
$C = 0$		
0		S
	S	
$C = 1$		
0		S
1	S	As,Bs
2	As	Bs
3	Bs	
$C = 2$		
0		S
1	S	As,Bs
2	As	Ba,Ca,Da,Bs
3	Ba	Ca,Da,Bs

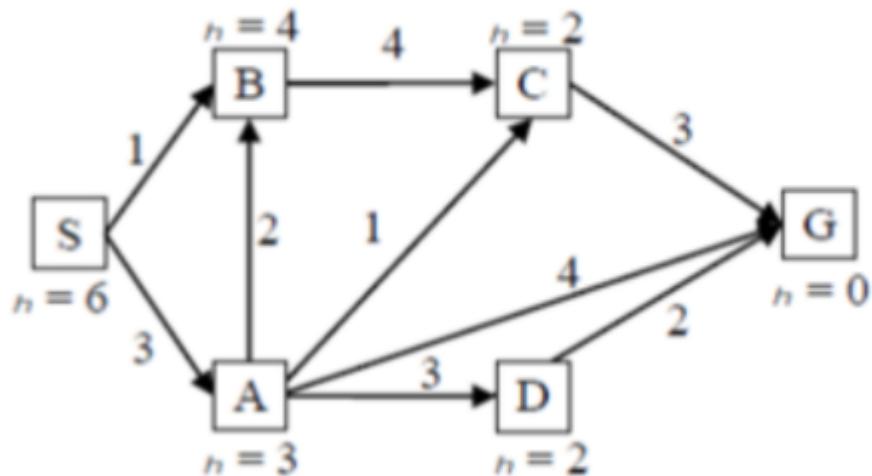
4	Ca	Da,Bs
5	Da	Bs
6	Bs	
$c = 3$		
0		S
1	S	As,Bs
2	As	Ba,Ca,Da,Bs
3	Ba	Cb,Ca,Da,Bs
4	Cb	Ca,Da,Bs
5	Ca	Dc,Gc,Da,Bs
6	Dc	Gc,Da,Bs
7	Gc	Đích

Đường đi: G ← C ← A ← S

Độ sâu d = 3

Ví dụ 4.2:

Câu 1.2:



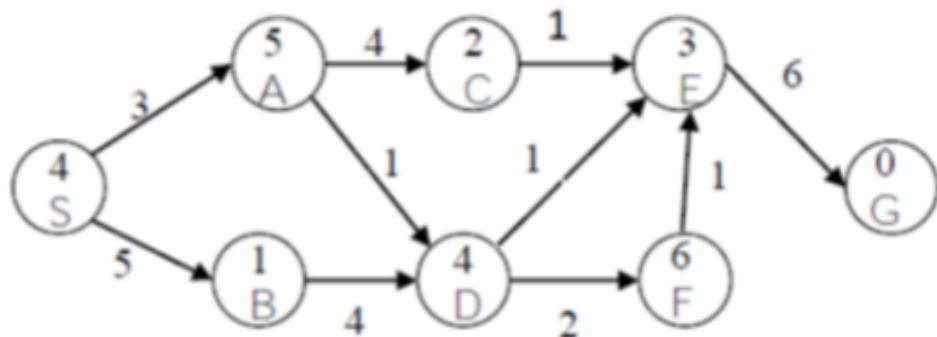
STT	Nút được mở rộng	Tập biên O
$C = 0$		
0		S
1	S	
$C = 1$		
0		S
1	S	As,Bs
2	As	Bs
3	Bs	
$C = 2$		
0		S
1	S	As,Bs
2	As	Ba,Ca,Da,Ga,Bs
3	Ba	Ca,Da,Ga,Bs
4	Ca	Da,Ga,Bs
5	Da	Ga,Bs
6	Ga	Đích

Đường đi : G ← A ← S

Độ sâu d = 2

Ví dụ 4.3:

Câu 1.3:



STT	Nút được mở rộng	Tập biên O
$C = 0$		
0		S
1	S	
$C = 1$		
0		S
1	S	As, Bs
2	As	Bs
3	Bs	
$C = 2$		
0		S
1	S	As, Bs
2	As	Ca, Da, Bs
3	Ca	Da, Bs
4	Da	Bs
5	Bs	
$C = 3$		

0		S
1	S	As,Bs
2	As	Ca,Da,Bs
3	Ca	Ec,Da,Bs
4	Ec	Da,Bs
5	Da	Fd,Bs
6	Fd	Bs
7	Bs	

C = 4

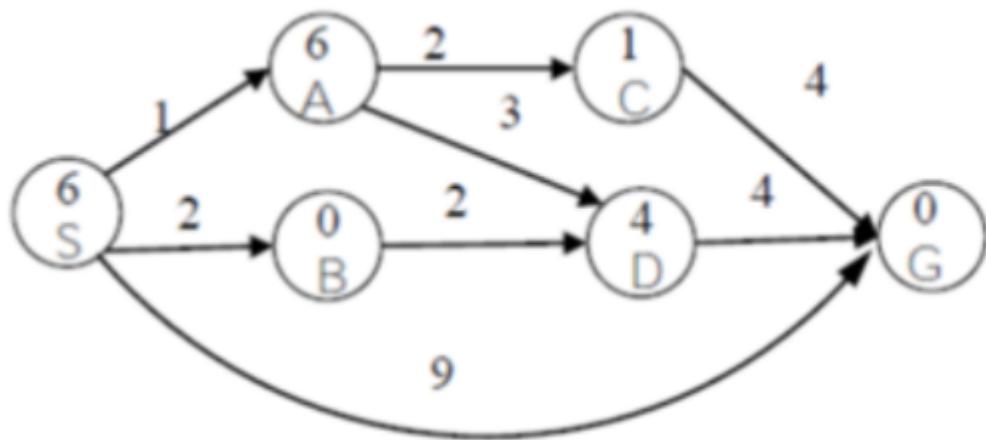
0		S
1	S	As,Bs
2	As	Ca,Da,Bs
3	Ca	Ec,Da,Bs
4	Ec	Ge,Da,Bs
5	Ge	Đích

Đường đi: G ← E ← C ← A ← S

Độ sâu d = 4

Ví dụ 4.4:

Câu 1.4:



STT	Nút được mở rộng	Tập biên O
$C = 0$		
0		S
	S	
$C = 1$		
0		S
1	S	As, Bs, Gs
2	As	Bs
3	Bs	Gs
4	Gs	Đích

Đường đi: $G \leftarrow S$

Độ sâu $d = 1$

5. Tìm kiếm tham lam

** $h(n)$ min

* Tìm kiếm tham lam

- Phương pháp: mở rộng nút có giá thành đường đi tới đích nhỏ nhất trước

$f(n) = h(n)$: hàm heuristic ước lượng giá thành đường đi từ n tới đích

- “Tham lam”: Chọn nút trong có vẻ tốt nhất để mở rộng, không quan tâm tới tương lai.

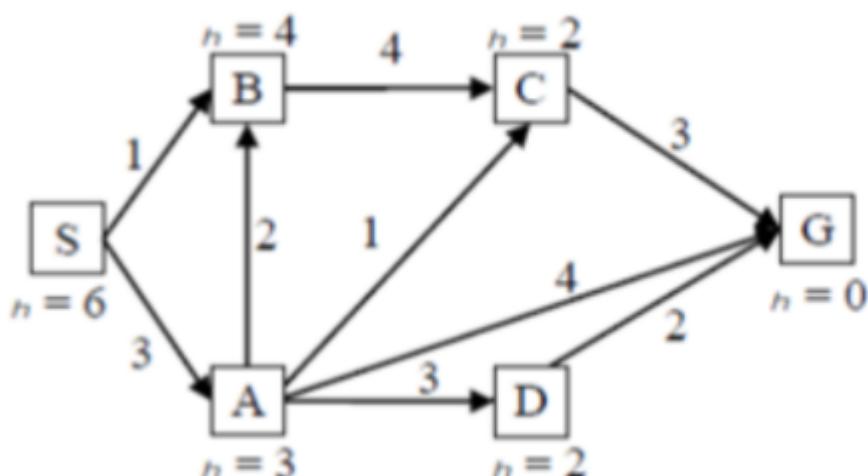
* Xử lý nút lặp:

▶ Tham lam

- **Không**: Việc đưa vào không làm thay đổi thuật toán (có thể dẫn đến vòng lặp)

Ví dụ 5.1:

Câu 1.2:



STT	Nút được mở rộng	Tập biên O
0		$S(6)$
1	S	$As(3), Bs(4)$
2	As	$Bs(4), Ca(2), Da(2), Ga(0)$

3	Ga	Đích
---	----	------

Đường đi: G←A←S

Chi phí c = 7

6. Tìm kiếm A*

$*f(m) = g(m) + h(m)$ min

* Tim kiếm A*

A^* (Q, S, G, P)

(Q: không gian trạng thái, S: trạng thái bắt đầu, G: đích, P: hành động, c: giá thành, h: heuristic)

Đầu vào: bài toán tìm kiếm

Đầu ra: đường đi tới nút đích

Khởi tạo: O ← S (O: danh sách các nút mở)

While(O ≠ ∅) do

1. lấy nút n có $f(n)$ nhỏ nhất khỏi O

2. if n ∈ G, return (đường đi tới n)

3. với mọi m ∈ P(n)

a) $g(m) = g(n) + c(n, m)$

b) $f(m) = g(m) + h(m)$

c) thêm m vào O cùng với giá trị $f(m)$

Return: không tìm được lời giải

*Xử lí nút lặp: Nếu nút có chi phí tốt hơn:

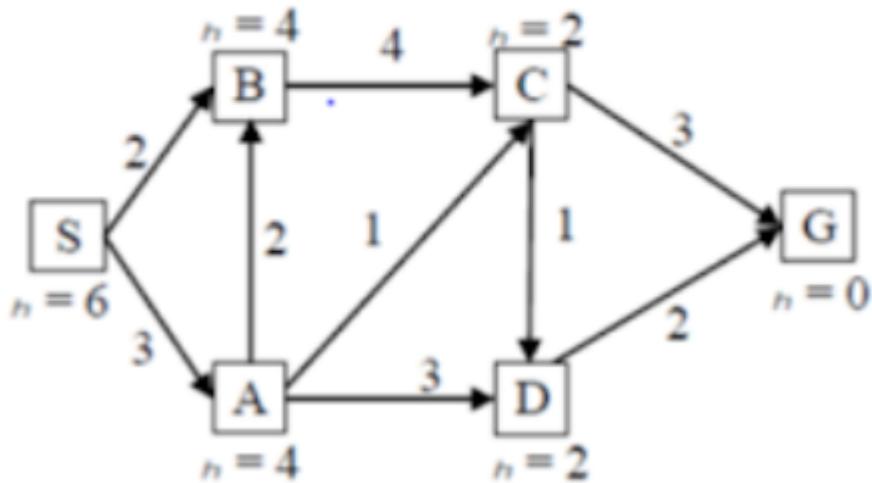
- **Đưa lại** vào danh sách nếu **đã mở rộng**
- **Cập nhật** nút cũ có giá thành kém hơn **nếu đang trong tập biên**

▶ A*

- Trong trường hợp nút lặp có giá thành (chi phí) tốt hơn, nó sẽ được **đưa lại danh sách** (nếu đã phát triển rồi) hoặc **cập nhật thay nút cũ** có giá thành kém hơn (nếu đang trong danh sách)

Ví dụ 6.1

Câu 1.1:



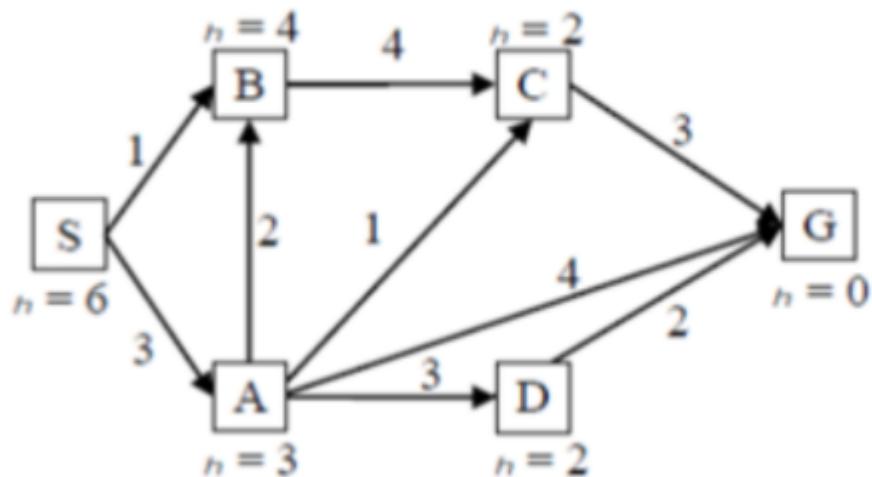
STT	Nút được mở rộng	Tập biên O
0		S(6)
1	S	As(7),Bs(6)
2	Bs	As(7),Cb(8)
3	As	Ca(6),Da(8)
4	Ca	Dc(7),Gc(7)
5	Dc	Gc(7)
6	Gc	Đích

Đường đi: G← C← A← S

Chi phí c = 7

Ví dụ 6.2:

Câu 1.2:

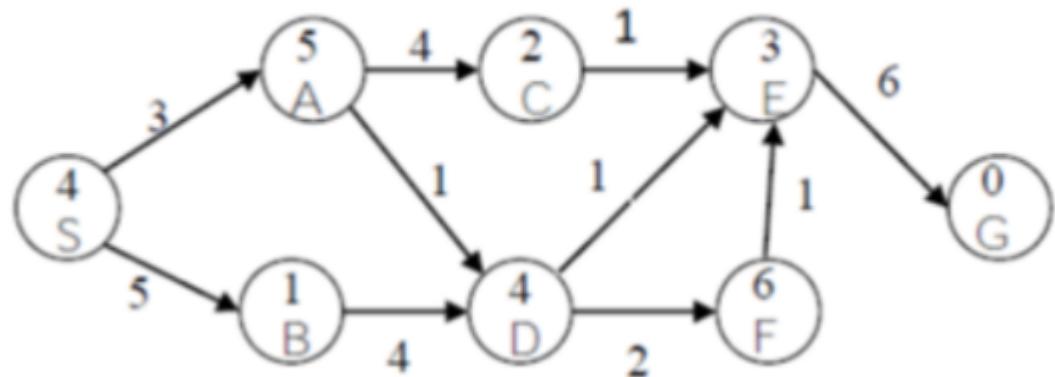


STT	Nút được mở rộng	Tập biên O
0		S(6)
1	S	As(6),Bs(5)
2	Bs	As(6),Cb(6)
3	As	Cb(6),Da(8),Ga(7)
4	Cb	Da(8),Ga(7)
5	Ga	Đích

Đường đi: $G \leftarrow A \leftarrow S$ chi phí c = 7

Ví dụ 6.3:

Câu 1.3:



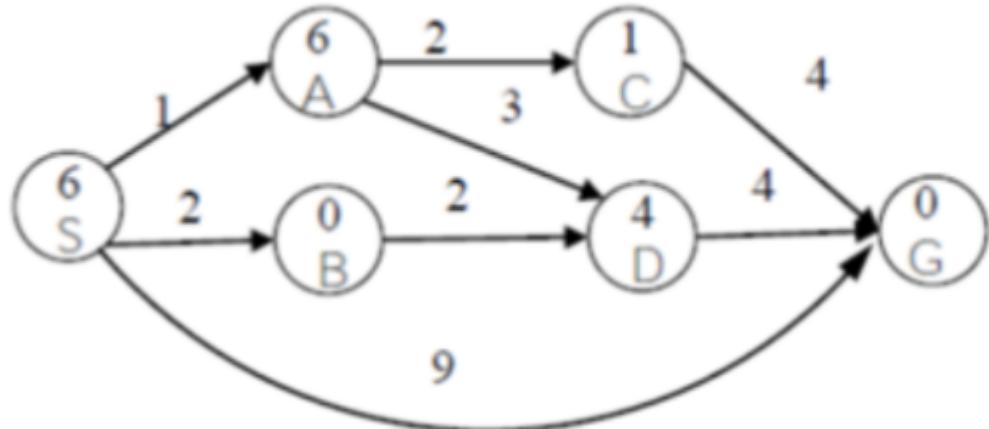
STT	Nút được mở rộng	Tập biên O
0		S(4)
1	S	As(8), Bs(6)
2	Bs	As(8), Db(13)
3	As	Da(8), Ca(9)
4	Da	Ca(9), Ed(8), Fd(12)
5	Ed	Ca(9), Fd(12), Ge(11)
6	Ca	Fd(12), Ge(11)
7	Ge	Đích

Đường đi:

G ← E ← D ← A ← S chi phí: 11

Ví dụ 6.4:

Câu 1.4:



STT	Nút được mở rộng	Tập biên O
0		S(6)
1	S	As(7), Bs(2), Gs(9)
2	Bs	As(7), Gs(9), Db(8)
3	As	Gs(9), Db(8), Ca(4)
4	Ca	Gc(7), Db(8),
5	Gc	Đích

Đường đi: $G \leftarrow C \leftarrow A \leftarrow S$

Chi phí c = 7

7.Tìm kiếm IDA*

IDB* = DFS + A*

-Thuật toán duyệt theo **DFS**

- $f(n)$ chỉ dùng để xét ngưỡng cho nút

IDA(Q, S, G, P, c, h)*

Đầu vào: bài toán tìm kiếm, hàm heuristic h

Đầu ra: đường đi ngắn nhất từ nút xuất phát đến nút đích

Khởi tạo: danh sách các nút biên (nút mở) $O \leftarrow S$

giá trị $i \leftarrow 0$ là ngưỡng cho hàm f

while (1) **do**

 1. **while** ($O \neq \emptyset$) **do**

 a) lấy nút n từ đầu O

 b) **if** n thuộc G , **return** (đường đi tới n)

 c) với mọi $m \in P(n)$

 i) $g(m) = g(n) + c(m, n)$

 ii) $f(m) = g(m) + h(m)$

 iii) **if** $f(m) \leq i$ **then** thêm m vào đầu O

 2. $i \leftarrow i + \beta, O \leftarrow S$

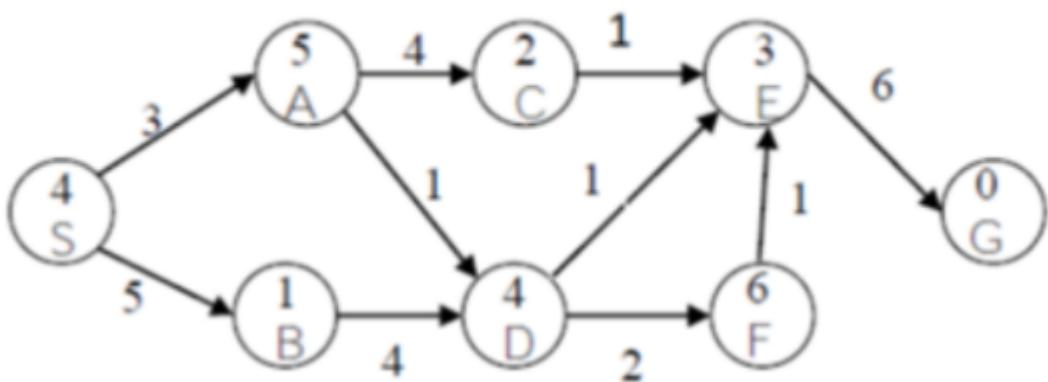
Xử lý nút lặp: Vì bản chất là DFS nên với mỗi ngưỡng i :

+KHÔNG lặp các nút ĐÃ ĐƯỢC MỞ RỘNG

+LẶP các nút ĐANG NẰM TRONG TẬP BIÊN

Ví dụ 7.1:

Câu 1.3:



$$\alpha = 8$$

STT	Nút được mở rộng	Tập biên O
$i=0$		
$i = 8$		
0		S(4)
1	S	As(8), Bs(6)
2	As	Da(8), Bs(6)
3	Da	Ed(8), Bs(6)
4	Ed	Bs(6)
5	Bs	
$i = 16$		
0		S(4)
1	S	As(8), Bs(6)
2	As	Ca(9), Da(8), Bs(6)
3	Ca	Ec(11), Da(8), Bs(6)
4	Ec	Ge(14), Da(8), Bs(6)
5	Ge	Đích

Đường đi: G← E← C← A← S

Chi phí c = 14

Phần 2: Logic Vị Tù

I. Logic mệnh đề

1.

► Một số công thức tương đương cơ bản

- $A \Rightarrow B \equiv \neg A \vee B$
- $A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$
- $\neg(\neg A) \equiv A$

► Luật De Morgan

- $\neg(A \vee B) \equiv \neg A \wedge \neg B$
- $\neg(A \wedge B) \equiv \neg A \vee \neg B$

▶ **Luật giao hoán**

- $A \vee B \equiv B \vee A$
- $A \wedge B \equiv B \wedge A$

▶ **Luật kết hợp**

- $(A \vee B) \vee C \equiv A \vee (B \vee C)$
- $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$

▶ **Luật phân phối**

- $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
- $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

2. Các quy tắc suy diễn

SỬ DỤNG CÁC QUY TẮC SUY DIỄN (1/2)

► Luật Modus Ponens

$$\frac{\alpha \Rightarrow \beta, \alpha}{\beta}$$

► Luật Modus Tollens

$$\frac{\alpha \Rightarrow \beta, \neg \beta}{\neg \alpha}$$

► Luật loại trừ và

$$\frac{\alpha_1 \wedge \dots \wedge \alpha_i \wedge \dots \wedge \alpha_m}{\alpha_i}$$

► Luật nhập đề và

$$\frac{\alpha_1, \dots, \alpha_i, \dots, \alpha_m}{\alpha_1 \wedge \dots \wedge \alpha_i \wedge \dots \wedge \alpha_m}$$

α, β, α_i là các công thức

SỬ DỤNG CÁC QUY TẮC SUY DIỄN (2/2)

- ▶ Luật nhập đề hoặc

$$\frac{\alpha_i}{\alpha_1 \vee \dots \vee \alpha_i \vee \dots \vee \alpha_m}$$

- ▶ Luật loại trừ phủ định kép

$$\frac{\neg(\neg\alpha)}{\alpha}$$

- ▶ Luật bắc cầu

$$\frac{\alpha \Rightarrow \beta, \beta \Rightarrow \gamma}{\alpha \Rightarrow \gamma}$$

$\alpha, \beta, \gamma, \alpha_i$ là các công thức

- ▶ Phép giải đơn vị

$$\frac{\alpha \vee \beta, \neg\beta}{\alpha}$$

- ▶ Phép giải

$$\frac{\alpha \vee \beta, \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

II. Logic vị từ

1.



Các công thức tương đương

1. $\forall x G(x) \equiv \forall y G(y)$
2. $\exists x G(x) \equiv \exists y G(y)$
3. $\neg(\forall x G(x)) \equiv \exists x (\neg G(x))$
4. $\neg(\exists x G(x)) \equiv \forall x (\neg G(x))$
5. $\forall x (G(x) \wedge H(x)) \equiv \forall x G(x) \wedge \forall x H(x)$
6. $\exists x (G(x) \vee H(x)) \equiv \exists x G(x) \vee \exists x H(x)$

2. Các quy tắc duy diễn



Các quy tắc suy diễn (1/5)

- ▶ Suy diễn với logic vị từ khó hơn logic mệnh đề do các biến có thể nhận vô số giá trị
 - Không thể dùng bảng chân lý
- ▶ Các quy tắc suy diễn cho logic mệnh đề cũng đúng với logic vị từ
 - Modus ponens, modus tollens, phủ định của phủ định, nhập để và/hoặc, loại trừ và/hoặc, phép giải
- ▶ Ngoài ra:
 - Có thêm một số quy tắc suy diễn dùng cho các lượng tử



Các quy tắc suy diễn (2/5)

- ▶ Phép thế (substitution)
 - Trước khi xem xét các quy tắc suy diễn, ta định nghĩa khái niệm phép thế, cần thiết cho những câu có chứa biến
 - Ký hiệu: $SUBST(\theta, a)$
 - Ý nghĩa: thế giá trị θ vào câu a
 - Ví dụ
 - $SUBST (\{x/Nam, y/An\}, Like(x, y)) = Like(Nam, An)$
- ▶ Phép loại trừ với mọi (universal elimination)

$$\frac{\forall x \alpha}{SUBST(\{x/g\}, \alpha)}$$

Ví dụ:

$$\forall x Like(x, IceCream) \quad \xrightarrow{\{x/Nam\}} \quad Like(Nam, IceCream)$$



Các quy tắc suy diễn (3/5)

▶ Phép loại trừ tồn tại (existential elimination)

$$\frac{\exists x \alpha}{\text{SUBST}(\{x/k\}, \alpha)} \quad k \text{ chưa xuất hiện trong KB}$$

Ví dụ:

$$\exists x \text{GoodAtMath}(x) \xrightarrow{\{x/C\}} \text{GoodAtMath}(C)$$

k được gọi là hằng Skolem và có thể đặt tên cho hằng này

▶ Nhập để tồn tại (existential introduction)

$$\frac{\alpha}{\exists x \text{SUBST}(\{g/x\}, \alpha)}$$

Ví dụ:

$$\text{Like(Nam, IceCream)} \xrightarrow{\{Nam/x\}} \exists x \text{Like}(x, \text{IceCream})$$



Các quy tắc suy diễn (4/5)

▶ Phép hợp nhất (unification)

- Hợp nhất là thủ tục xác định phép thế cần thiết để làm cho 2 câu cơ sở giống nhau
- **Kí hiệu:** $\text{UNIFY}(p, q) = (\theta)$
- $\text{SUBST}(\theta, p) = \text{SUBST}(\theta, q)$
- θ được gọi là hợp tử (phản tử hợp nhất)
- Trong trường hợp có nhiều hợp tử thì ta sử dụng hợp tử tổng quát nhất, tức là hợp tử sử dụng ít phép thế cho biến nhất
- MGU: most general unifier
- Phép hợp nhất có thể thực hiện tự động bằng thuật toán có độ phức tạp tỉ lệ tuyến tính với số lượng biến



Các quy tắc suy diễn (5/5)

▶ Modus Ponens tổng quát (GMP)

- Giả sử ta có các câu cơ sở p_i , p'_i , q , và tồn tại phép thay thế θ sao cho $UNIFY(p_i, p'_i) = \theta$, với mọi i
- Khi đó ta có:

$$\frac{p'_1, p'_2, \dots, p'_n, (p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q)}{SUBST(\theta, q)}$$

▶ Sử dụng GMP cho phép xây dựng thuật toán suy diễn tự động, suy diễn tiến và suy diễn lùi

III. Chuẩn hóa dạng CNF và Caluse Form

**1 trong 2 dạng:

-Dạng 1: A1 v A2 v.. Am

-Dạng 2: A1 (các câu đơn)



Conjunctive Normal Form (CNF) và Clause Form

- ▶ Clause là tuyển của literal, có dạng
 $A_1 \vee A_2 \vee \dots \vee A_m$, trong đó các A_i là literal
- ▶ Conjunctive Normal Form (CNF - dạng chuẩn hội), là câu bao gồm hội của phép tuyển của các literal hoặc là hội của clause
 - $A \wedge (B \vee C) \wedge (D \vee E \vee F)$
- ▶ Có thể biến đổi một công thức bất kỳ về công thức ở dạng CNF bằng cách áp dụng một số bước thủ tục



Đưa về CNF và Clause Form (1/3)

- ▶ Bước 1: Khử tương đương
 - Thay $P \Leftrightarrow Q$ bằng $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$
- ▶ Bước 2: Loại bỏ kéo theo
 - Thay $P \Rightarrow Q$ bởi công thức tương đương $\neg P \vee Q$
- ▶ Bước 3: Đưa các phủ định vào gần vị từ
 - Chuyển các dấu phủ định (\neg) vào sát các vị từ bằng cách áp dụng luật De Morgan và thay $\neg(\neg A)$ bởi A :
 - $\neg(\neg P) \equiv P$
 - $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$
 - $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
 - $\neg(\forall x Q) \equiv \exists x (\neg Q)$
 - $\neg(\exists x Q) \equiv \forall x (\neg Q)$



Đưa về CNF và Clause Form (2/3)

- ▶ Bước 4: Chuẩn hóa tên biến sao cho mỗi lượng tử có biến riêng

- Ví dụ

$$\begin{array}{l} \forall x \neg P(x) \vee Q(x) \\ \forall x \neg R(x) \vee Q(x) \end{array}$$



$$\begin{array}{l} \forall x \neg P(x) \vee Q(x) \\ \forall y \neg R(y) \vee Q(y) \end{array}$$

- ▶ Bước 5: Loại bỏ các lượng tử tồn tại bằng cách sử dụng hằng Skolem và hàm Skolem

- Biến đổi $\exists x P(x)$ thành $P(C)$, trong đó C là hằng mới (Skolem)
- Nếu \exists nằm trong \forall thì thay bằng hàm có biến là biến của \forall , hàm phải chưa xuất hiện trong KB và được gọi là hàm Skolem
- Ví dụ:

$$\forall x \exists y P(x, y) \text{ thành } \forall x P(x, f(x)), \quad f(x) \text{ là hàm Skolem}$$



Đưa về CNF và Clause Form (2/3)

- ▶ Bước 6: Loại bỏ các lượng tử với mọi (\forall)

- Để loại bỏ lượng tử với mọi (\forall), ta đưa các lượng tử với mọi (\forall) sang trái sau đó bỏ lượng tử với mọi (\forall)
- Ví dụ: $\forall x (P(x, y) \vee Q(x))$ thành $P(x, y) \vee Q(x)$

- ▶ Bước 7: Sắp xếp “và” ra ngoài “hoặc”

- $(P \wedge Q) \vee R \equiv (P \vee R) \wedge (Q \vee R)$
- $(P \vee Q) \vee R \equiv (P \vee R) \vee Q$

- ▶ Bước 8: Loại bỏ các phép “và”

- Ta thực hiện loại bỏ các phép “và” để tạo thành các clause riêng
- Ví dụ: $(P \vee R \vee S) \wedge (Q \vee \neg R)$ thành 2 câu: 1) $P \vee R \vee S$ 2) $Q \vee \neg R$

- ▶ Bước 9: Chuẩn hóa tên biến sao cho mỗi câu có biến riêng của mình

****Tóm Tắt:**

Bước 1: Khử tương đương \Leftrightarrow , kéo theo \Rightarrow bên trong \exists, \forall

▶ Một số công thức tương đương cơ bản

- $A \Rightarrow B \equiv \neg A \vee B$
 - $A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$

Bước 2: Khử \exists, \forall

- +Khữ \exists : -TH1: $\exists x P(x) \equiv P(C)$ { x/C }
 - TH2: $\forall x \exists y P(x,y) \equiv \forall x P(x, f(x))$
- +Khữ \forall : $\forall x (P(x) \vee Q(x)) \equiv P(x) \vee Q(x)$

Bước 3: Đưa về 1 trong 2 dạng:

Dạng 1: A1 v A2 v A3 v.. An

▶ Luật kết hợp

- $(A \vee B) \vee C \equiv A \vee (B \vee C)$
 - $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$

▶ Luật phân phối

- $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
 - $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

Dạng 2: A1 (câu đơn)

▶ Luật loại trừ và

$$\frac{\alpha_1 \wedge \dots \wedge \alpha_i \wedge \dots \wedge \alpha_m}{\alpha_i}$$

Ví dụ:

Boy(Nam) ^ Handsome(Nam)

Áp dụng luật loại trừ và ta có 2 câu mới: Boy(Nam) (1)

Handsome(Nam) (2)

Bước 4: Chuẩn hóa tên sao cho mỗi câu có biến của riêng mình.

Ví dụ 8.1:

Biểu diễn các câu sau ở dạng logic vị từ và viết chúng ở dạng chuẩn CNF (clause Form):

► Cho các câu sau

1. Mọi bé trai đều thích chơi bóng đá
2. Ai thích chơi bóng đá đều có giày đá bóng
3. Nam là một bé trai

Giải:

$$1. \forall x (\text{Boy}(x) \Rightarrow \text{Like}(x, \text{Football}))$$

$$\equiv \forall x (\neg \text{Boy}(x) \vee \text{Like}(x, \text{Football}))$$

$$\equiv \neg \text{Boy}(x) \vee \text{Like}(x, \text{Football})$$

$$2.. \forall x (\text{Like}(x, \text{Football}) \Rightarrow \text{Have}(x, \text{Shoes}))$$

$$\equiv \forall x (\neg \text{Like}(x, \text{Football}) \vee \text{Have}(x, \text{Shoes}))$$

$$\equiv \neg \text{Like}(x, \text{Football}) \vee \text{Have}(x, \text{Shoes})$$

$$\equiv \neg \text{Like}(y, \text{Football}) \vee \text{Have}(y, \text{Shoes})$$

$$3. \text{Boy}(\text{Nam})$$

Ví dụ 8.2:

Câu 2 (2 điểm)

Cho các mệnh đề sau dưới dạng ngôn ngữ tự nhiên và logic vị từ

- Máy tính mới thì chạy nhanh. $\forall x(M(x) \Rightarrow N(x))$

- Máy tính phòng thực hành chạy chậm. $\forall x(T(x) \Rightarrow \neg N(x))$

- Một số máy phòng thực hành có bộ nhớ ram lớn. $\exists x(T(x) \wedge R(x))$

a) Chuẩn hóa các câu trên về dạng Chuẩn tắc hội (CNF).

$$1. \forall x (M(x) \Rightarrow N(x))$$

$$\equiv \forall x (\neg M(x) \vee N(x))$$

$$\equiv \neg M(x) \vee N(x)$$

$$2.. \forall x (T(x) \Rightarrow \neg N(x))$$

$$\equiv \forall x (\neg T(x) \vee \neg N(x))$$

$$\equiv \neg T(x) \vee \neg N(x)$$

$$\equiv \neg T(y) \vee \neg N(y)$$

3. $\exists x(T(x) \wedge R(x))$
 $\equiv \exists x(T(x) \wedge R(x))$
 $\equiv T(C) \wedge R(C) \{x/C\}$: Luật loại trừ tồn tại
Áp dụng luật loại trừ và
3.1. $T(C)$ 3.2. $R(C)$

IV. Suy diễn tiến

4.1. Định nghĩa

*Bắt đầu từ các câu trong KB, sử dụng **GMP** và các quy tắc suy diễn để sinh ra các câu mới cho đến khi sinh không thể sinh ra câu nào nữa.



Ví dụ hợp nhất

p	q	θ
$Know(Nam, x)$	$Know(Nam, Bắc)$	$\{x/Bắc\}$
$Know(Nam, x)$	$Know(y, MotherOf(y))$	$\{y/Nam, x/MotherOf(Nam)\}$
$Know(Nam, x)$	$Know(y, z)$	$\{y/Nam, x/z\}$ $\{y/Nam, x/Nam, z/Nam\}$

► Phép hợp nhất (unification)

- Hợp nhất là thủ tục xác định phép thế cần thiết để làm cho 2 câu cơ sở giống nhau
- **Kí hiệu:** $UNIFY(p, q) = (\theta)$
 $SUBST(\theta, p) = SUBST(\theta, q)$
 θ được gọi là hợp tử (phản tử hợp nhất)

▶ Modus Ponens tổng quát (GMP)

- Giả sử ta có các câu cơ sở p_i, p'_i, q , và tồn tại phép thay thế θ sao cho $UNIFY(p_i, p'_i) = \theta$, với mọi i
- Khi đó ta có:

$$\frac{p_1, p'_1, \dots, p_n, (p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q)}{SUBST(\theta, q)}$$

*Trong bài thường gặp 2 dạng **GMP**:

Dạng 1:

p_1 và p'_1 có thể hợp nhất tại phép thay thế Θ

$$\frac{p'_1, p_1 \Rightarrow q}{SUBT(\Theta, q)}$$

Ví dụ:

$$1. \forall x Cat(x) \Rightarrow Like(x, Fish)$$

$$3. Cat(Tom)$$

\Rightarrow

$$4. GMP(1)(3) \Rightarrow Like(Tom, Fish)$$

Dạng 2:

p_1 và p'_1 , p_2 và p'_2 có thể hợp nhất tại phép thay thế Θ

$$\frac{p'_1, p'_2, p_1 \wedge p_2 \Rightarrow q}{SUBT(\Theta, q)}$$

Ví dụ:

2. $\forall x, y \text{ Cat}(x) \wedge \text{Like}(x, y) \Rightarrow \text{Eat}(x, y)$

3. $\text{Cat}(\text{Tom})$

4. $\text{Like}(\text{Tom}, \text{Fish})$

\Rightarrow

5. GMP (2) (3) (4) $\Rightarrow \text{Eat}(\text{Tom}, \text{Fish})$

tại { x/Tom, y/Fish }

4.2. Các ví dụ suy diễn tiến

*Ví dụ 4.2.1

Câu 2.1:

Câu 2 (2 điểm)

Cho cơ sở tri thức KB sau dưới dạng ngôn ngữ tự nhiên và logic vị từ:

- Chó đốm là chó. $\forall x (\text{Dalmatian}(x) \Rightarrow \text{Dog}(x))$

- Bo là chó đốm. $\text{Dalmatian}(Bo)$

- Chó đốm thích uống sữa. $\forall x (\text{Dalmatian}(x) \Rightarrow \text{Drink}(x, \text{Milk}))$

- Bo biết làm xiếc. $\text{Circus}(Bo)$

a) Viết truy vấn câu sau “Có con chó thích uống sữa và biết làm xiếc” dưới dạng logic vị từ sử dụng các vị từ đã cho.

*Chứng minh câu truy vấn trong phần a sử dụng phương pháp suy diễn tiến.

$$\forall x (\text{Dalmatian}(x) \Rightarrow \text{Dog}(x)) \quad (1)$$

$$\text{Dalmatian}(Bo) \quad (2)$$

$$\forall x (\text{Dalmatian}(x) \Rightarrow \text{Like}(x, \text{Milk})) \quad (3)$$

$$\text{Circus}(Bo) \quad (4)$$

Q: $\exists x (\text{Dog}(x) \wedge \text{Like}(x, \text{Milk}) \wedge \text{Circus}(x))$

*Cách làm:

-Suy diễn tiến: Sử dụng KB và GMP để suy ra câu :

$\text{Dog}(Bo) \wedge \text{Like}(Bo, \text{Milk}) \wedge \text{Circus}(Bo)$

-Sử dụng nhập đề tồn tại

▶ Nhập đề tồn tại (existential introduction)

Ví dụ:

$$\frac{\alpha}{\exists x \text{ SUBST}(\{g/x\}, \alpha)}$$

$$\text{Like}(Nam, IceCream) \xrightarrow{\{Nam/x\}} \exists x \text{ Like}(x, IceCream)$$

Giải:

$$\text{GMP (1),(2)} \Rightarrow \text{Dog(Bo)} \quad \{x/\text{Bo}\} \quad (5)$$

$$\text{GMP (2),(3)} \Rightarrow \text{Like(Bo, Milk)} \quad \{x/\text{Bo}\} \quad (6)$$

Nhập đề và: (5),(6),(7)

$$\Rightarrow \text{Dog(Bo) } \wedge \text{ Like(Bo, Milk) } \wedge \text{ Circus(Bo)} \quad (7)$$

Nhập đề tồn tại $\Rightarrow \exists x (\text{Dog}(x) \wedge \text{Like}(x, Milk) \wedge \text{Circus}(x))$ (đpcm)

*Ví dụ 4.2.2

Câu 2.2:

Câu 2 (2 điểm)

Cho các mệnh đề sau dưới dạng ngôn ngữ tự nhiên và logic vị từ

- Máy tính mới thi chạy nhanh. $\forall x(M(x) \Rightarrow N(x))$

- Máy tính phòng thực hành chạy chậm. $\forall x(T(x) \Rightarrow \neg N(x))$

- Một số máy phòng thực hành có bộ nhớ ram lớn. $\exists x(T(x) \wedge R(x))$

a) Chuẩn hóa các câu trên về dạng Chuẩn tắc hội (CNF).

b) Viết câu truy vấn sau “Có những máy tính có bộ nhớ ram lớn nhưng chậm” dưới dạng logic vị từ sử dụng các vị từ đã cho ở trên; và chứng minh câu truy vấn đúng sử dụng phép giải và phản chứng.

$$\forall x(M(x) \Rightarrow N(x)) \quad (1)$$

$$\forall x(T(x) \Rightarrow \neg N(x)) \quad (2)$$

$$\exists x(T(x) \wedge R(x)) \quad (3)$$

$$Q: \exists x(R(x) \wedge \neg N(x))$$

$$-\text{Chuẩn hóa câu (3) ta được 2 câu: } T(C) \quad (4)$$

$$R(C) \quad (5)$$

$$\text{GMP (2),(4)} \Rightarrow \neg N(C) \quad (6)$$

$$\text{Nhập đề và (5),(6)} \Rightarrow R(C) \wedge \neg N(C) \quad (7)$$

Nhập đề tồn tại cho (7) $\Rightarrow \exists x(R(x) \wedge \neg N(x))$ (đpcm)

Ví dụ 4.2.3

Câu 2 (2 điểm)

Cho các mệnh đề sau dưới dạng ngôn ngữ tự nhiên và logic vị từ

- Trẻ em thích Ipad. $\forall x(Child(x) \Rightarrow Like(x, Ipad))$

- Trẻ em đòi mua những gì mình thích. $\forall x \forall y(Child(x) \wedge Like(x, y) \Rightarrow Buy(x, y))$

- Nam là một em bé. $Child(Nam)$

a) Chuẩn hóa các câu trên về dạng chuẩn tắc hội (CNF).

b) Viết câu truy vấn sau “Nam đòi mua Ipad” dưới dạng logic vị từ; và chứng minh câu truy vấn đúng sử dụng thủ tục suy diễn lùi.

*Chứng minh câu truy vấn trong phần a sử dụng phương pháp suy diễn tiên.

$$\forall x(Child(x) \Rightarrow Like(x, Ipad)) \quad (1)$$

$$\forall x \forall y(Child(x) \wedge Like(x, y) \Rightarrow Buy(x, y)) \quad (2)$$

$$Child(Nam) \quad (3)$$

$$Q: Buy(Nam, Ipad)$$

$$GMP (1) (2) \Rightarrow Like(Nam, Ipad) \quad (4)$$

$$GMP (2),(3),(4) \Rightarrow Buy(Nam, Ipad) \text{ (đpcm)}$$

Ví dụ 4.2.4:

Câu 2 (2 điểm)

Cho các mệnh đề sau dưới dạng ngôn ngữ tự nhiên và logic vị từ

- Những người biết lập trình và biết dùng máy tính đều hay lướt Web. $\forall x(P(x) \wedge C(x) \Rightarrow W(x))$

- Ai biết lập trình đều dùng được máy tính. $\forall x(P(x) \Rightarrow C(x))$

- Ai học công nghệ thông tin cũng biết lập trình. $\forall x(IT(x) \Rightarrow P(x))$

- Nam học công nghệ thông tin. $IT(Nam)$

a) Chuẩn hóa các câu trên về dạng chuẩn tắc hội (CNF).

b) Viết câu truy vấn sau “Nam hay lướt Web” dưới dạng logic vị từ; và chứng minh câu truy vấn đúng sử dụng thủ tục suy diễn lùi.

$$\forall x(P(x) \wedge C(x) \Rightarrow W(x)) \quad (1)$$

$$\forall x(P(x) \Rightarrow C(x)) \quad (2)$$

$$\forall x(IT(x) \Rightarrow P(x)) \quad (3)$$

$$IT(Nam) \quad (4)$$

$$Q: W(Nam)$$

$$GMP (3),(4) \Rightarrow P(Nam) \quad (5)$$

$$GMP (2),(5) \Rightarrow C(Nam) \quad (6)$$

$$GMP (1),(5),(6) \Rightarrow W(Nam) \quad (\text{đpcm})$$

Ví dụ 4.2.5:

Câu 2.5:

Cho các mệnh đề sau dưới dạng ngôn ngữ tự nhiên và logic vị từ:

- Tất cả những người đi học là người có văn hóa. $\forall x(Hoc(x) \Rightarrow VanHoa(x))$
 - Trộm không có văn hóa. $\forall x(Trom(x) \Rightarrow \neg VanHoa(x))$
 - Một số tên trộm thông minh. $\exists x(Trom(x) \wedge ThongMinh(x))$
- a) Viết các câu trên dưới dạng câu tuyên (clause form)
- b) Viết câu truy vấn sau “Có một số người thông minh không được đi học” dưới dạng logic vị từ sử dụng các vị từ đã cho ở trên và chứng minh câu truy vấn là đúng sử dụng suy diễn tiến.

$$\forall x(Hoc(x) \Rightarrow VanHoa(x)) \quad (1)$$

$$\forall x(Trom(x) \Rightarrow \neg VanHoa(x)) \quad (2)$$

$$\exists x(Trom(x) \wedge ThongMinh(x)) \quad (3)$$

$$Q: \exists x(ThongMinh(x) \wedge \neg Hoc(x))$$

Chuẩn hóa CNF (3) ta được 2 câu:

$$Trom(C) \quad (4)$$

$$ThongMinh(C) \quad (5)$$

*Ý tưởng: sử dụng GMP để suy ra câu $ThongMinh(C) \wedge \neg Hoc(C)$ sau đó dùng nhập đề tồn tại suy ra đpcm.

$$GMP(2),(4) \Rightarrow \neg VanHoa(C) \quad (6)$$

► Luật Modus Tollens

$$\frac{\alpha \Rightarrow \beta, \neg \beta}{\neg \alpha}$$

$$Modus\ Tollens\ (1), (6) \Rightarrow \neg Hoc(C) \quad (7)$$

$$Nhập\ đề\ và\ (5),(7) \Rightarrow ThongMinh(C) \wedge \neg Hoc(C)$$

$$Nhập\ đề\ tồn\ tại \Rightarrow \exists x(ThongMinh(x) \wedge \neg Hoc(x))\ (\text{đpcm})$$

V. Suy diễn lùi

5.1. Định nghĩa

- Ngược lại với suy diễn tiến, suy diễn lùi bắt đầu từ **câu truy vấn**, sau đó tìm các sự kiện và quy tắc trong KB cho phép chứng minh câu truy vấn là đúng.

Thủ tục suy diễn lùi:

-Với mỗi câu q , nếu tồn tại q' hợp nhất được với q thì trả về q' hợp từ q được chứng minh

Ví dụ:

KB: Cat(Tom)

Q: Cat(Tom) ?

câu truy vấn Q hợp nhất được với Cat(Tom) trong KB \rightarrow Q được chứng minh.

-Với mỗi quy tắc có **về phải** q' hợp nhất được với q cố gắng chứng minh các phần tử **về trái** bằng suy diễn lùi:

► Luật Modus Ponens

$$\frac{\alpha \Rightarrow \beta, \alpha}{\beta}$$

Ví dụ:

KB: $\forall x (Hoc(x) \wedge Gioi(x) \Rightarrow VanHoa(x))$

Q: VanHoa(Nam)

câu truy vấn Q hợp nhất được với **về phải** của KB

\Rightarrow cần chứng minh **về trái** $Hoc(Nam) \wedge Gioi(Nam)$ bằng suy diễn lùi.

*Chú ý:

Nếu câu truy vấn Q có dạng các câu đơn hội với nhau:

$A \wedge B \wedge C..$

ta chứng minh từng câu đơn bằng suy diễn lùi.

vì nếu các câu đơn đúng, áp dụng luật nhập đề và thì các câu đơn hội với nhau cũng tất nhiên đúng.

Luật nhập đề và

$$\frac{\alpha_1, \dots, \alpha_i, \dots, \alpha_m}{\alpha_1 \wedge \dots \wedge \alpha_i \wedge \dots \wedge \alpha_m}$$

5.2. Các ví dụ suy diễn lùi

Ví dụ 5.2.1:

Câu 2 (2 điểm)

Cho các mệnh đề sau dưới dạng ngôn ngữ tự nhiên và logic vị từ

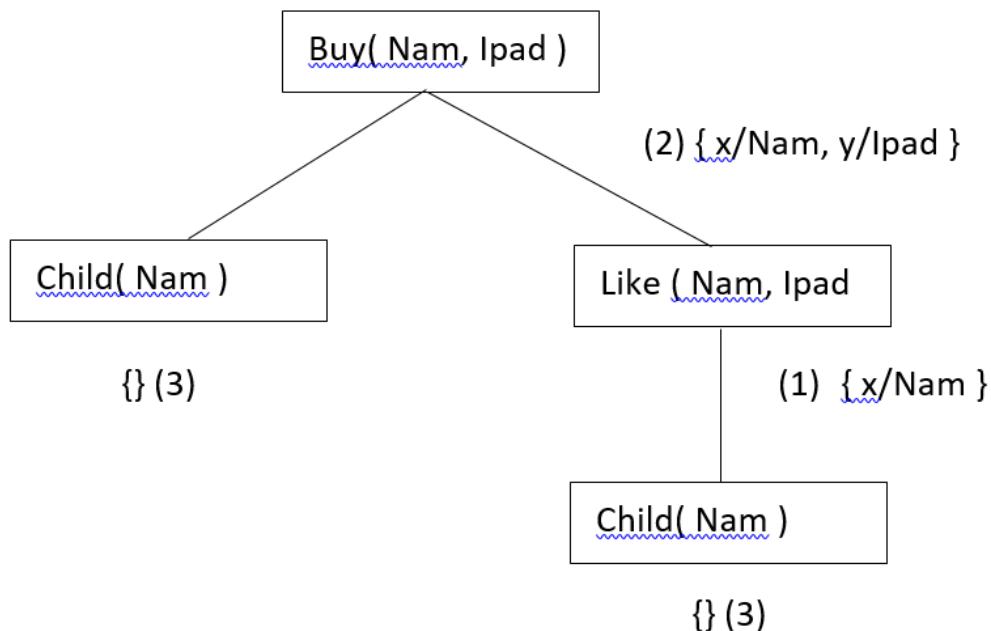
- Trẻ em thích Ipad. $\forall x(Child(x) \Rightarrow Like(x, Ipad))$
 - Trẻ em đòi mua những gì mình thích. $\forall x \forall y(Child(x) \wedge Like(x, y) \Rightarrow Buy(x, y))$
 - Nam là một em bé. $Child(Nam)$
- a) Chuẩn hóa các câu trên về dạng chuẩn tắc hội (CNF).
 - b) Viết câu truy vấn sau “Nam đòi mua Ipad” dưới dạng logic vị từ; và chứng minh câu truy vấn đúng sử dụng thủ tục suy diễn lùi.

$$\forall x(Child(x) \Rightarrow Like(x, Ipad)) \quad (1)$$

$$\forall x \forall y(Child(x) \wedge Like(x,y) \Rightarrow Buy(x,y)) \quad (2)$$

$$Child(Nam) \quad (3)$$

Q: $Buy(Nam, Ipad) ?$

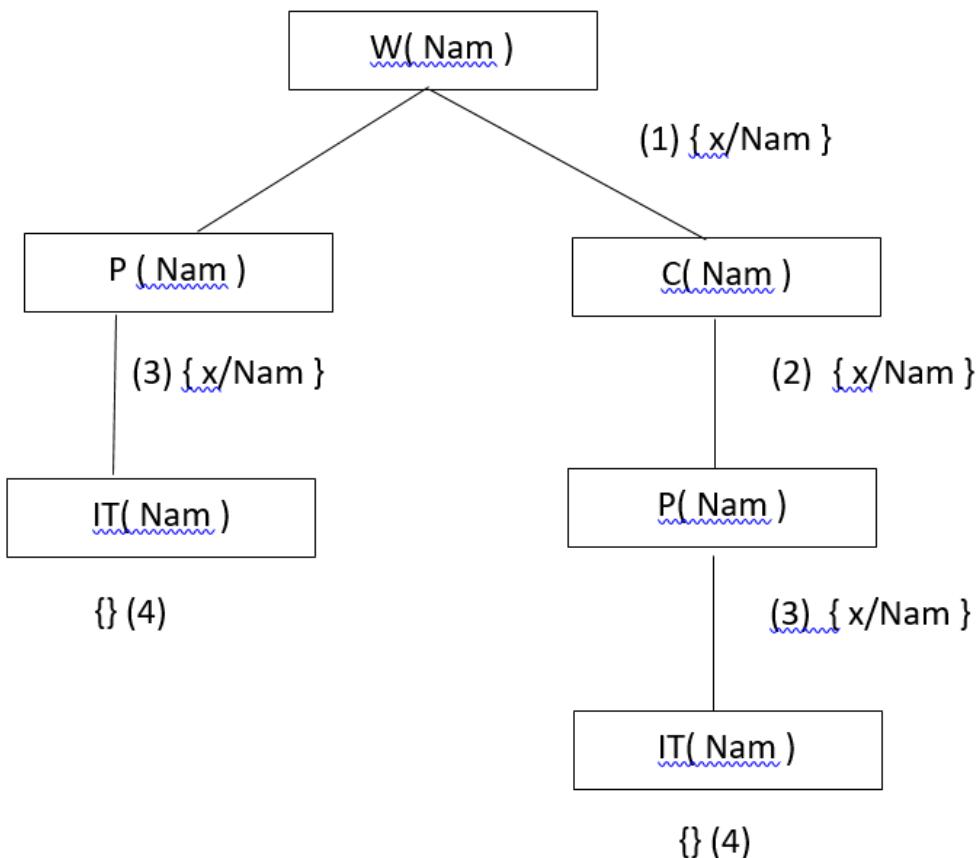


Ví dụ 5.2.2

Câu 2 (2 điểm)

Cho các mệnh đề sau dưới dạng ngôn ngữ tự nhiên và logic vị từ

- Những người biết lập trình và biết dùng máy tính đều hay lướt Web. $\forall x(P(x) \wedge C(x) \Rightarrow W(x))$
 - Ai biết lập trình đều dùng được máy tính. $\forall x(P(x) \Rightarrow C(x))$
 - Ai học công nghệ thông tin cũng biết lập trình. $\forall x(IIT(x) \Rightarrow P(x))$
 - Nam học công nghệ thông tin. $IIT(Nam)$
- a) Chuẩn hóa các câu trên về dạng chuẩn tắc hội (CNF).
 - b) Viết câu truy vấn sau “Nam hay lướt Web” dưới dạng logic vị từ; và chứng minh câu truy vấn đúng sử dụng thủ tục suy diễn lùi.



VI. Suy diễn sử dụng phép giải

6.1. Phương pháp: Sử dụng phép giải và các quy tắc suy diễn để chứng minh câu truy vấn cần tìm.

*Chú ý: khi sử dụng phép giải cần đưa hết các câu về dạng CNF
 $(A \vee B \vee C, \text{hoặc } A)$

-Trong logic mệnh đề:

- ▶ **Phép giải đơn vị**

$$\frac{\alpha \vee \beta, \neg \beta}{\alpha}$$

- ▶ **Phép giải**

$$\frac{\alpha \vee \beta, \neg \beta \vee \gamma}{\alpha \vee \gamma}$$

-Trong logic vị từ:

$p_2 \vee \neg p_2'$ có thể hợp nhất bởi phép thế θ

$$\frac{p_1 \vee p_2, \neg p_2' \vee p_3}{SUBT(\theta, p_1 \vee p_3)}$$

Ví dụ 1:

$$\text{Giàu}(x) \vee \text{Giỏi}(x) \quad (1)$$

$$\neg \text{Giỏi}(\text{Bắc}) \vee \text{Đẹp Trai}(\text{Bắc}) \quad (2)$$

$$\text{Áp dụng phép giải} \Rightarrow \text{Giàu}(\text{Bắc}) \vee \text{Đẹp Trai}(\text{Bắc})$$

Ví dụ 2:

$$\text{Cao}(x) \vee \text{DepTrai}(x) \quad (3)$$

$$\neg \text{Cao}(\text{Nam}) \quad (4)$$

$$\text{Áp dụng phép giải} \Rightarrow \text{DepTrai}(\text{Nam})$$

6.2. Các ví dụ suy diễn sử dụng phép giải

Ví dụ 6.2.1:

KB:

$$\neg \text{Dalmatian}(x) \vee \text{Dog}(x) \quad (1)$$

$$\text{Dalmatian}(\text{Bo}) \quad (2)$$

$$\neg \text{Dalmatian}(y) \vee \text{Like}(y, \text{Milk}) \quad (3)$$

$$\text{Circus}(\text{Bo}) \quad (4)$$

$$Q: \exists x (\text{Dog}(x) \wedge \text{Like}(x, \text{Milk}) \wedge \text{Circus}(x))$$

cần chứng minh câu $\text{Dog}(\text{Bo}) \wedge \text{Like}(\text{Bo}, \text{Milk}) \wedge \text{Circus}(\text{Bo})$

sau đó áp dụng nhập đề tồn tại

\Rightarrow chứng minh từng câu $\text{Dog}(\text{Bo})$, $\text{Like}(\text{Bo}, \text{Milk})$, $\text{Circus}(\text{Bo})$ sử dụng phép giải.

Giải:

$$\text{Áp dụng phép giải (1),(2)} \Rightarrow \text{Dog}(\text{Bo}) \quad (5)$$

$$\text{Áp dụng phép giải (2),(3)} \Rightarrow \text{Like}(\text{Bo}, \text{Milk}) \quad (6)$$

$$\text{Nhập đề và (4),(5),(6)} \Rightarrow \text{Dog}(\text{Bo}) \wedge \text{Like}(\text{Bo}, \text{Milk}) \wedge \text{Circus}(\text{Bo})$$

Nhập đề tồn tại \Rightarrow đpcm

Ví dụ 6.2.2

Câu 2 (2 điểm)

Cho cơ sở tri thức KB sau dưới dạng ngôn ngữ tự nhiên và logic vị từ:

- Chó đốm là chó. $\forall_x(\text{Dalmatian}(x) \Rightarrow \text{Dog}(x))$
- Bo là chó đốm. $\text{Dalmatian}(\text{Bo})$
- Chó đốm thích uống sữa. $\forall_x(\text{Dalmatian}(x) \Rightarrow \text{Drink}(x, \text{Milk}))$
- Bo biết làm xiếc. $\text{Circus}(\text{Bo})$

a) Viết truy vấn câu sau “Có con chó thích uống sữa và biết làm xiếc” dưới dạng logic vị từ sử dụng các vị từ đã cho.

b) Chứng minh câu truy vấn đúng sử dụng phép giải và phản chứng.

Câu 2.2:

Câu 2 (2 điểm)

Cho các mệnh đề sau dưới dạng ngôn ngữ tự nhiên và logic vị từ

- Máy tính mới thì chạy nhanh. $\forall_x(M(x) \Rightarrow N(x))$
- Máy tính phòng thực hành chạy chậm. $\forall_x(T(x) \Rightarrow \neg N(x))$
- Một số máy phòng thực hành có bộ nhớ ram lớn. $\exists_x(T(x) \wedge R(x))$

a) Chuẩn hóa các câu trên về dạng Chuẩn tắc hội (CNF).

b) Viết câu truy vấn sau “Có những máy tính có bộ nhớ ram lớn nhưng chậm” dưới dạng logic vị từ sử dụng các vị từ đã cho ở trên; và chứng minh câu truy vấn đúng sử dụng phép giải và phản chứng.

$$\neg M(x) \vee N(x) \quad (1)$$

$$\neg T(y) \vee \neg N(y) \quad (2)$$

$$T(C) \quad (3)$$

$$R(C) \quad (4)$$

$$Q: \exists x(R(x) \wedge \neg N(x))$$

$$\text{Áp dụng phép giải (2),(3)} \Rightarrow \neg N(C) \quad (5)$$

$$\text{Nhập đề và (5),(4)} \Rightarrow R(C) \wedge \neg N(C) \quad (6)$$

$$\text{Nhập đề tồn tại cho (6)} \Rightarrow \exists x(R(x) \wedge \neg N(x)) \quad (\text{đpcm})$$

Ví dụ 6.2.3

Câu 2 (2 điểm)

Cho các mệnh đề sau dưới dạng ngôn ngữ tự nhiên và logic vị từ

- Trẻ em thích Ipad. $\forall_x(\text{Child}(x) \Rightarrow \text{Like}(x, \text{Ipad}))$
- Trẻ em đòi mua những gì mình thích. $\forall_x \forall_y(\text{Child}(x) \wedge \text{Like}(x, y) \Rightarrow \text{Buy}(x, y))$
- Nam là một em bé. $\text{Child}(\text{Nam})$

a) Chuẩn hóa các câu trên về dạng chuẩn tắc hội (CNF).

b) Viết câu truy vấn sau “Nam đòi mua Ipad” dưới dạng logic vị từ; và chứng minh câu truy vấn đúng sử dụng thủ tục suy diễn lùi.

$$\neg \text{Child}(x) \vee \text{Like}(x, \text{Ipad}) \quad (1)$$

$$\neg \text{Child}(z) \vee \neg \text{Like}(z, y) \vee \text{Buy}(z, y) \quad (2)$$

$$\text{Child}(\text{Nam}) \quad (3)$$

Q: Buy(Nam, Ipad)

Áp dụng phép giải đơn vị (2),(3) $\Rightarrow \neg \text{Like}(\text{Nam}, y) \vee \text{Buy}(\text{Nam}, y)$ (4)

Áp dụng phép giải đơn vị (1),(3) $\Rightarrow \text{Like}(\text{Nam}, \text{Ipad})$ (5)

Áp dụng phép giải (4),(5) $\Rightarrow \text{Buy}(\text{Nam}, \text{Ipad})$ (đpcm)

Ví dụ 6.2.4:

Câu 2 (2 điểm)

Cho các mệnh đề sau dưới dạng ngôn ngữ tự nhiên và logic vị từ

- Những người biết lập trình và biết dùng máy tính đều hay lướt Web. $\forall x(P(x) \wedge C(x) \Rightarrow W(x))$

- Ai biết lập trình đều dùng được máy tính. $\forall x(P(x) \Rightarrow C(x))$

- Ai học công nghệ thông tin cũng biết lập trình. $\forall x(\text{IT}(x) \Rightarrow P(x))$

- Nam học công nghệ thông tin. $\text{IT}(\text{Nam})$

a) Chuẩn hóa các câu trên về dạng chuẩn tắc hội (CNF).

b) Viết câu truy vấn sau “Nam hay lướt Web” dưới dạng logic vị từ; và chứng minh câu truy vấn đúng sử dụng thủ tục suy diễn lùi.

$$\neg P(x) \vee \neg C(x) \vee W(x) \quad (1)$$

$$\neg P(y) \vee C(y) \quad (2)$$

$$\neg \text{IT}(z) \vee P(z) \quad (3)$$

$$\text{IT}(\text{Nam}) \quad (4)$$

Q: $W(\text{Nam})$

Áp dụng phép giải (3),(4) $\Rightarrow P(\text{Nam})$ (5)

Áp dụng phép giải (2),(5) $\Rightarrow C(\text{Nam})$ (6)

Áp dụng phép giải (1),(5),(6) $\Rightarrow W(\text{Nam})$ (đpcm)

Ví dụ 6.2.5:

Câu 2.5:

Cho các mệnh đề sau dưới dạng ngôn ngữ tự nhiên và logic vị từ:

- Tất cả những người đi học là người có văn hóa. $\forall x(\text{Hoc}(x) \Rightarrow \text{VanHoa}(x))$

- Trộm không có văn hóa. $\forall x(\text{Trom}(x) \Rightarrow \neg \text{VanHoa}(x))$

- Một số tên trộm thông minh. $\exists x(\text{Trom}(x) \wedge \text{ThongMinh}(x))$

a) Viết các câu trên dưới dạng câu tuyên (clause form)

b) Viết câu truy vấn sau “Có một số người thông minh không được đi học” dưới dạng logic vị từ sử dụng các vị từ đã cho ở trên và chứng minh câu truy vấn là đúng sử dụng suy diễn tiên.

$$\neg \text{Hoc}(x) \vee \text{VanHoa}(x) \quad (1)$$

-Trom(y) v -VanHoa(y) (2)

Trom(C) (3)

ThongMinh(C) (4)

Q: $\exists x (\text{ThongMinh}(x) \wedge \neg \text{Hoc}(x))$

Áp dụng phép giải (2),(3) $\Rightarrow \neg \text{VanHoa}(C)$ (5)

Áp dụng phép giải (1),(5) $\Rightarrow \neg \text{Hoc}(C)$ (6)

Nhập đề và (4),(6) $\Rightarrow \text{ThongMinh}(C) \wedge \neg \text{Hoc}(C)$

Nhập đề tồn tại \Rightarrow (đpcm)

VII. Suy diễn sử dụng phép giải và phản chứng

7.1. Phương pháp

▶ Cần chứng minh $KB \vdash Q$?

▶ Cách làm:

- Thêm $\neg Q$ vào KB , chứng minh tồn tại một tập con của KB mới có giá trị False
- $(KB \vdash Q) \Leftrightarrow (KB \wedge \neg Q \vdash \text{False})$

▶ Thuật toán

- $KB = UNION(KB, \neg Q)$
- **while** (KB không chứa False) **do**
 - 1. Chọn 2 câu S_1, S_2 từ KB sao cho có thể áp dụng phép giải cho 2 câu này
 - Thêm kết quả phép giải vào KB
 - 2. Nếu không có hai câu như vậy
 - **return** False
- **end while**
- **return** Success

*Tóm tắt:

+Thêm $\neg Q$ vào KB

+While(KB không chứa False)

{

-Chọn 2 câu trong KB có thể áp dụng được phép giải

-Thêm kết quả vào KB

}

***Lưu ý:**

-Do sử dụng phép giải nên các câu trong KB cần đưa về dạng CNF

7.2. Các ví dụ suy diễn sử dụng phép giải và phản chứng

Ví dụ 7.2.1:

Câu 2.1:

Câu 2 (2 điểm)

Cho cơ sở tri thức KB sau dưới dạng ngôn ngữ tự nhiên và logic vị từ:

- Chó đốm là chó. $\forall_x (\text{Dalmatian}(x) \Rightarrow \text{Dog}(x))$

- Bo là chó đốm. $\text{Dalmatian}(\text{Bo})$

- Chó đốm thích uống sữa. $\forall_x (\text{Dalmatian}(x) \Rightarrow \text{Drink}(x, \text{Milk}))$

- Bo biết làm xiếc. $\text{Circus}(\text{Bo})$

a) Viết truy vấn câu sau “Có con chó thích uống sữa và biết làm xiếc” dưới dạng logic vị từ sử dụng các vị từ đã cho.

b) Chứng minh câu truy vấn đúng sử dụng phép giải và phản chứng.

$$3. \quad \neg(\forall x G(x)) \equiv \exists x(\neg G(x))$$

$$4. \quad \neg(\exists x G(x)) \equiv \forall x(\neg G(x))$$

KB:

$$\neg \text{Dalmatian}(x) \vee \text{Dog}(x) \quad (1)$$

$$\text{Dalmatian}(\text{Bo}) \quad (2)$$

$$\neg \text{Dalmatian}(y) \vee \text{Like}(y, \text{Milk}) \quad (3)$$

$$\text{Circus}(\text{Bo}) \quad (4)$$

$$Q: \exists x(\text{Dog}(x) \wedge \text{Like}(x, \text{Milk}) \wedge \text{Circus}(x))$$

Thêm vào KB:

$$\neg(\exists x(\text{Dog}(x) \wedge \text{Like}(x, \text{Milk}) \wedge \text{Circus}(x)))$$

$$\equiv \forall x(\neg(\text{Dog}(x) \wedge \text{Like}(x, \text{Milk}) \wedge \text{Circus}(x)))$$

$$\equiv \forall x(\neg \text{Dog}(x) \vee \neg \text{Like}(x, \text{Milk}) \vee \neg \text{Circus}(x))$$

$$\equiv \neg \text{Dog}(x) \vee \neg \text{Like}(x, \text{Milk}) \vee \neg \text{Circus}(x)$$

$$\equiv \neg \text{Dog}(z) \vee \neg \text{Like}(z, \text{Milk}) \vee \neg \text{Circus}(z) \quad (5)$$

$$\text{Áp dụng phép giải (4),(5)} \Rightarrow \neg \text{Dog}(\text{Bo}) \vee \neg \text{Like}(\text{Bo}, \text{Milk}) \quad (6)$$

Áp dụng phép giải (1),(2) $\Rightarrow \text{Dog(Bo)}$ (7)

Áp dụng phép giải (2),(3) $\Rightarrow \text{Like(Bo,Milk)}$ (8)

Áp dụng phép giải (6),(7) $\Rightarrow \neg \text{Like(Bo, Milk)}$ (9)

Áp dụng phép giải (8),(9) $\Rightarrow \text{False(đpcm)}$

Ví dụ 7.2.2:

Câu 2.2:

Câu 2 (2 điểm)

Cho các mệnh đề sau dưới dạng ngôn ngữ tự nhiên và logic vị từ

- Máy tính mới thì chạy nhanh. $\forall_x(M(x) \Rightarrow N(x))$

- Máy tính phòng thực hành chạy chậm. $\forall_x(T(x) \Rightarrow \neg N(x))$

- Một số máy phòng thực hành có bộ nhớ ram lớn. $\exists_x(T(x) \wedge R(x))$

a) Chuẩn hóa các câu trên về dạng Chuẩn tắc hội (CNF).

b) Viết câu truy vấn sau “Có những máy tính có bộ nhớ ram lớn nhưng chậm” dưới dạng logic vị từ sử dụng các vị từ đã cho ở trên; và chứng minh câu truy vấn đúng sử dụng phép giải và phản chứng.

$$\neg M(x) \vee N(x) \quad (1)$$

$$\neg T(y) \vee \neg N(y) \quad (2)$$

$$T(C) \quad (3)$$

$$R(C) \quad (4)$$

$$Q: \exists x(R(x) \wedge \neg N(x))$$

Thêm vào KB:

$$\neg(\exists x(R(x) \wedge \neg N(x)))$$

$$\equiv \forall x(\neg R(x) \vee N(x))$$

$$\equiv \neg R(z) \vee N(z) \quad (5)$$

Áp dụng phép giải (4),(5) $\Rightarrow N(C)$ (6)

Áp dụng phép giải (2),(6) $\Rightarrow \neg T(C)$ (7)

Áp dụng phép giải (3),(7) $\Rightarrow \text{False}$

Phần 3: Suy diễn với Mạng Bayes

I. Các tiên đề xác xuất và một số tính chất cơ bản

► Các tính chất của xác suất điều kiện

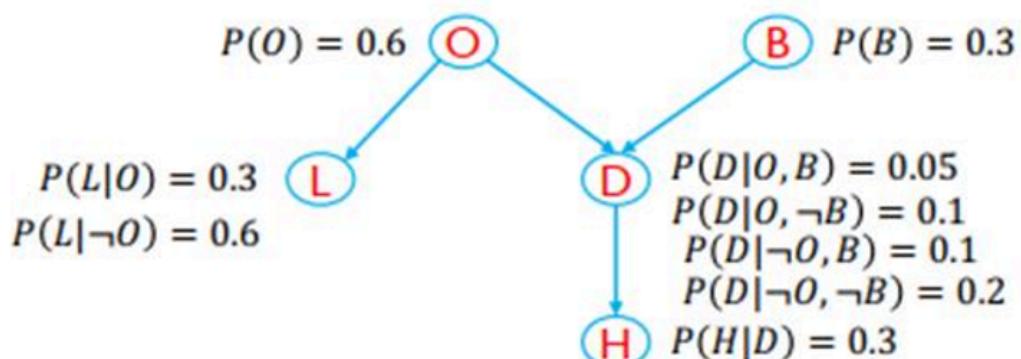
- $P(A, B) = P(A|B)P(B)$
- Quy tắc chuỗi: $P(A, B, C, D) = P(A|B, C, D) P(B|C, D) P(C|D) P(D)$
- Quy tắc chuỗi có điều kiện: $P(A, B|C) = P(A|B, C) P(B|C)$
- Quy tắc Bayes: $P(A|B) = \frac{P(A) P(B|A)}{P(B)}$
- Bayes có điều kiện: $P(A|B, C) = \frac{P(B, C|A) \cdot P(A)}{P(B, C)}$
- $P(A) = \sum_b \{P(A|B = b) P(B = b)\}$, tổng lấy theo tất cả giá trị b của B
- $P(\neg B|A) = 1 - P(B|A)$

II. Suy diễn với Mạng bayes

1. Tính chất đặc biệt của Mạng bayes

*Mỗi nút V **độc lập** có điều kiện với tất cả các nút không phải **hậu duệ** của V, nếu biết giá trị các nút **bố mẹ** của V.

Ví dụ 1.1:



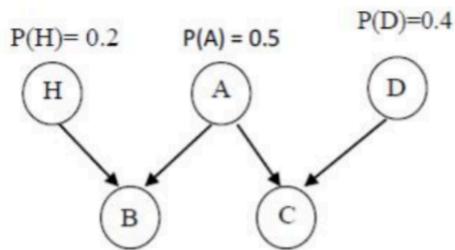
H độc lập có điều kiện với L,O,B nếu biết D

$$P(H | L, O, B, D) = P(H | D)$$

Ví dụ 1.2:

Câu 3 (3 điểm)

Cho mạng Bayes sau, các biến có thể nhận giá trị {T,F} ({true, false})



H	A	$P(B = T A, H)$
F	F	0.7
F	T	0.1
T	F	0.2
T	T	0.6

A	D	$P(C = T A, D)$
F	F	0.8
F	T	0.4
T	F	0.3
T	T	0.1

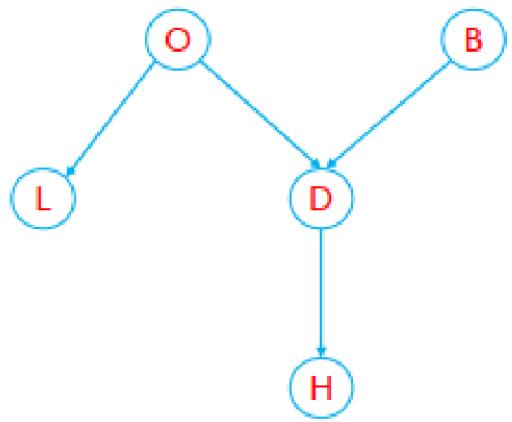
a) Tính xác suất cả năm biến cùng nhận giá trị F.

$$\begin{aligned}
 & P(\neg B, \neg C, \neg A, \neg D, \neg H) \\
 &= P(\neg B | \neg C, \neg A, \neg D, \neg H) \cdot P(\neg C | \neg A, \neg D, \neg H) \\
 &\quad \cdot P(\neg A | \neg D, \neg H) \cdot P(\neg D | \neg H) \cdot P(\neg H) \\
 &= P(\neg B | \neg A, \neg H) \cdot P(\neg C | \neg A, \neg D) \cdot P(\neg A) \cdot P(\neg H) \cdot P(\neg D) \\
 &= (1 - 0.7) \cdot (1 - 0.8) \cdot (1 - 0.5) \cdot (1 - 0.2) \cdot (1 - 0.4) \\
 &= ...
 \end{aligned}$$

2. Suy diễn chuẩn đoán và suy diễn nhân quả

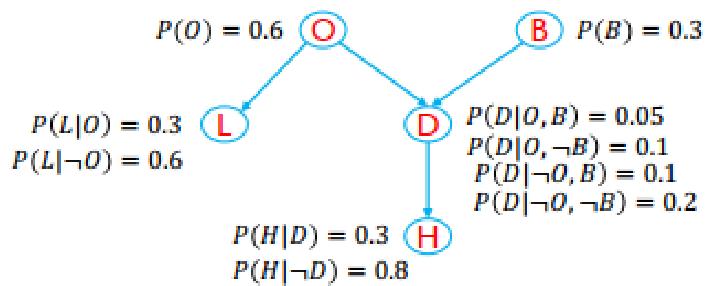
► Trường hợp đơn giản nhất

- Khi bằng chứng E và kết quả Q có duy nhất một liên kết trực tiếp với nhau
- Phân biệt 2 trường hợp:
 - **Suy diễn nhân quả** (trên xuống): cần tính $P(Q|E)$ khi E là nút cha của Q
 - **Suy diễn chẩn đoán** (dưới lên): cần tính $P(Q|E)$ khi E là nút con của Q



Suy diễn nhân quả (1/3)

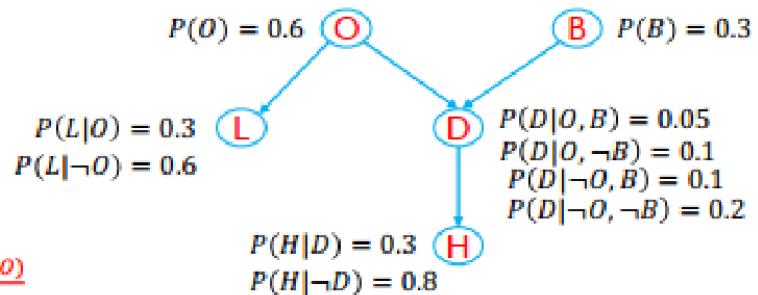
► Ví dụ: tính $P(D|B)$



Suy diễn nhân quả (2/3)

► Ví dụ: tính $P(D|B)$

$$\begin{aligned}
 P(D|B) &= \frac{P(D, B)}{P(B)} \\
 &= \frac{P(D, B, O) + P(D, B, \neg O)}{P(B)} \\
 &= \frac{P(D|B, O)P(B, O) + P(D|B, \neg O)P(B, \neg O)}{P(B)} \\
 &= \frac{P(D|B, O)P(B)P(O) + P(D|B, \neg O)P(B)P(\neg O)}{P(B)} \\
 &= P(D|B, O)P(O) + P(D|B, \neg O)P(\neg O) \\
 &= (0.05)(0.6) + (0.1)(1 - 0.6) \\
 &= 0.07
 \end{aligned}$$



▶ Áp dụng cho cả suy diễn nhân quả và suy diễn chuẩn đoán

- **Bước 1:** Đưa xác suất điều kiện về các xác suất đồng thời
- **Bước 2:** Sử dụng tính độc lập xác suất trên mạng Bayes, viết lại các xác suất đồng thời dưới dạng các xác suất điều kiện của nút con khi biết các giá trị bố mẹ
- **Bước 3:** Sử dụng các giá trị xác suất từ bảng xác suất điều kiện để tính

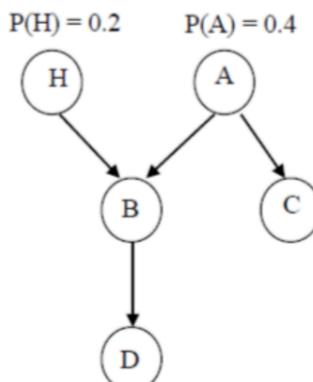
Ví dụ 2.1:

Câu 3.1:

Câu 3 (3 điểm)

Cho mạng Bayes sau, các biến có thể nhận giá trị {T, F} ({true, false})

- a) Tính xác suất cả năm biến cùng nhận giá trị F.
 b) Tính $P(A|B)$.
 c) Mạng đã cho có dạng Polytree hay không ?



H	A	$P(B A, H)$
F	F	0.6
F	T	0.2
T	F	0.1
T	T	0.5

A	$P(C A)$
T	0.6
F	0.4

B	$P(D B)$
T	0.4
F	0.6

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

*Tính $P(B|A)$:

$$P(B|A) = \frac{P(A,B)}{P(A)} = \frac{P(A,B,H) + P(A,B,\neg H)}{P(A)} = \frac{P(B|A,H).P(A,H) + P(B|A,\neg H).P(A,\neg H)}{P(A)}$$

*Tính $(B \setminus A)$:

$$P(B \setminus \neg A) = \frac{P(\neg A, B)}{P(\neg A)} = \frac{P(\neg A, B, H) + P(\neg A, B, \neg H)}{P(\neg A)} = \frac{P(B \setminus \neg A, H) \cdot P(\neg A, H) + P(B \setminus \neg A, \neg H) \cdot P(\neg A, \neg H)}{P(\neg A)}$$

$$P(B \setminus \neg A) = \frac{P(B \setminus \neg A, H) \cdot P(\neg A) \cdot P(H) + P(B \setminus \neg A, \neg H) \cdot P(\neg A) \cdot P(\neg H)}{P(\neg A)}$$

$$P(B \setminus \neg A) = P(B \setminus \neg A, H) \cdot P(H) + P(B \setminus \neg A, \neg H) \cdot P(\neg H)$$

$$P(B \setminus \neg A) = 0.1 \cdot 0.2 + 0.6 \cdot (1 - 0.2) = 0.5$$

Theo quy tắc Bayes:

$$P(A \setminus B) = \frac{P(B \setminus A) \cdot P(A)}{P(B)} = \frac{0.26 \cdot 0.4}{P(B)}$$

$$P(\neg A \setminus B) = \frac{P(B \setminus \neg A) \cdot P(\neg A)}{P(B)} = \frac{0.5 \cdot (1 - 0.4)}{P(B)}$$

$$\text{Mà } P(A \setminus B) + P(\neg A \setminus B) = 1 \Rightarrow P(B) = 0.404$$

$$\Rightarrow P(A \setminus B) = \frac{P(B \setminus A) \cdot P(A)}{P(B)} = \frac{0.26 \cdot 0.4}{0.404} = 0.26$$

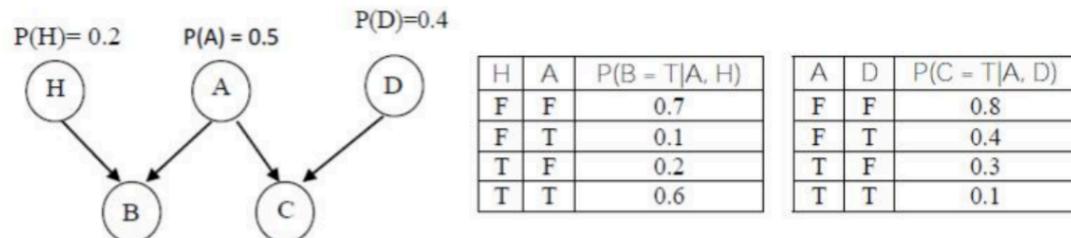
*Mạng đã cho có dạng Polytree do giữa 2 nút bất kì chỉ có 1 đường đi.

Ví dụ 2.2:

Câu 3.2:

Câu 3 (3 điểm)

Cho mạng Bayes sau, các biến có thể nhận giá trị {T,F} ({true, false})



- a) Tính xác suất cả năm biến cùng nhận giá trị F.
- b) Tính $P(A|C)$.
- c) Tính $P(A|B,C)$.

*Tính $P(A|B,C)$:

$$P(A|B,C) = \frac{P(B,C|A) \cdot P(A)}{P(B,C)} = \frac{P(B|A) \cdot P(C|A) \cdot P(A)}{P(B,C)} \quad (*)$$

$$P(B,C|A) = \frac{P(B,C,A)}{P(A)} = \frac{P(B|C,A) \cdot P(C|A) \cdot P(A)}{P(A)} = P(B|A) \cdot P(C|A)$$

$$P(A|B,C) + P(\neg A|B,C) = 1 \quad (**)$$

$$P(\neg A|B,C) = \frac{P(B,C|\neg A).P(\neg A)}{P(B,C)} = \frac{P(B|\neg A).P(C|\neg A).P(A)}{P(B,C)}$$

⇒ Tính được $P(B,C)$ từ (**)

thay vào (*) ⇒ $P(A|B,C)$

*Lời giải chi tiết:

b)

$$P(C|A) = \frac{P(A,C)}{P(A)} = \frac{P(A,C,D) + P(A,C,\neg D)}{P(A)} = \frac{P(C|A,D).P(A,D) + P(C|A,\neg D).P(A,\neg D)}{P(A)}$$

$$P(C|A) = \frac{P(C|A,D).P(A).P(D) + P(C|A,\neg D).P(A).P(\neg D)}{P(A)}$$

$$P(C|A) = P(C|A,D).P(D) + P(C|A,\neg D).P(\neg D)$$

$$P(C|A) = 0.1 \cdot 0.4 + 0.3 \cdot (1 - 0.4) = 0.22$$

$$P(C|\neg A) = \frac{P(\neg A,C)}{P(\neg A)} = \frac{P(\neg A,C,D) + P(\neg A,C,\neg D)}{P(\neg A)} = \frac{P(C|\neg A,D).P(\neg A,D) + P(C|\neg A,\neg D).P(\neg A,\neg D)}{P(\neg A)}$$

$$P(C|\neg A) = \frac{P(C|\neg A,D).P(\neg A).P(D) + P(C|\neg A,\neg D).P(\neg A).P(\neg D)}{P(\neg A)}$$

$$P(C|\neg A) = P(C|\neg A,D).P(D) + P(C|\neg A,\neg D).P(\neg D)$$

$$P(C|\neg A) = 0.4 \cdot 0.4 + 0.8 \cdot (1 - 0.4) = 0.64$$

$$P(B|\neg A) = \frac{P(\neg A,B)}{P(\neg A)} = \frac{P(\neg A,B,H) + P(\neg A,B,\neg H)}{P(\neg A)} = \frac{P(B|\neg A,H).P(\neg A,H) + P(B|\neg A,\neg H).P(\neg A,\neg H)}{P(\neg A)}$$

$$P(B|\neg A) = \frac{P(B|\neg A,H).P(\neg A).P(H) + P(B|\neg A,\neg H).P(\neg A).P(\neg H)}{P(\neg A)}$$

$$P(B|\neg A) = P(B|\neg A,H).P(H) + P(B|\neg A,\neg H).P(\neg H)$$

$$P(B|\neg A) = 0.2 \cdot 0.2 + 0.7 \cdot (1 - 0.2) = 0.6$$

Theo quy tắc Bayes:

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} = \frac{0.2 \cdot 0.5}{P(B)} ; P(\neg A|B) = \frac{P(B|\neg A).P(\neg A)}{P(B)} = \frac{0.6 \cdot (1-0.5)}{P(B)}$$

Mà $P(A|B) + P(\neg A|B) = 1 \Rightarrow P(B) = 0.4$

Theo quy tắc Bayes: $P(A|B,C) = \frac{P(B,C|A).P(A)}{P(B,C)} ; P(\neg A|B,C) = \frac{P(B,C|\neg A).P(\neg A)}{P(B,C)}$

Theo tính độc lập xác suất: $P(B,C|A) = P(B|A).P(C|A) ; P(B,C|\neg A) = P(B|\neg A).P(C|\neg A)$

$$\Rightarrow P(A|B,C) = \frac{P(B|A).P(C|A).P(A)}{P(B,C)} = \frac{0.2 \cdot 0.22 \cdot 0.5}{P(B,C)}$$

$$P(\neg A|B,C) = \frac{P(B|\neg A).P(C|\neg A).P(\neg A)}{P(B,C)} = \frac{0.6 \cdot 0.64 \cdot 0.5}{P(B,C)}$$

Mà $P(A|B,C) + P(\neg A|B,C) = 1 \Rightarrow P(B, C) = 0.214$

$$P(A|B,C) = \frac{P(B|A).P(C|A).P(A)}{P(B,C)} = \frac{0.2 \cdot 0.22 \cdot 0.5}{0.214} = 0.103$$

Phần 4: Học Máy

1. Thuật toán ID3



Thuật toán ID3

- ▶ Xây dựng lần lượt các nút của cây bắt đầu từ gốc
- ▶ Thuật toán
 - **Khởi đầu:** nút hiện thời là nút gốc chứa toàn bộ tập dữ liệu huấn luyện
 - Tại nút hiện thời *n*, lựa chọn thuộc tính
 - Chưa được sử dụng ở nút tổ tiên
 - Cho phép phân chia tập dữ liệu hiện thời thành các tập con **một cách tốt nhất**
 - Với mỗi giá trị thuộc tính được chọn thêm một nút con bên dưới
 - Chia các ví dụ ở nút hiện thời về các nút con theo giá trị thuộc tính được chọn
 - **Lặp** (đệ quy) cho tới khi
 - Tất cả các thuộc tính đã được sử dụng ở các nút phía trên, hoặc
 - Tất cả ví dụ tại nút hiện thời có cùng nhãn phân loại
 - Nhãn của nút được lấy theo đa số nhãn của ví dụ tại nút hiện thời

Lựa chọn thuộc tính tại mỗi nút thế nào?



Entropy

- Trường hợp tập dữ liệu S có 2 loại nhãn: đúng (+) hoặc sai (-)

$$\text{Entropy}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

p_+ : % số mẫu đúng, p_- : % số mẫu sai

- Trường hợp tổng quát: có C loại nhãn

$$\text{Entropy}(S) = \sum_{i=1}^C p_i \log_2 p_i$$

p_i : % ví dụ của S thuộc loại i

- Ví dụ

$$\begin{aligned}\text{Entropy}([9^+, 5^-]) &= -(9/14) \log_2(9/14) - (5/14) \log_2(5/14) \\ &= 0.94\end{aligned}$$



Độ tăng thông tin IG

Với tập (con) mẫu S và thuộc tính A

$$IG(S, A) = \text{Entropy}(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

Trong đó:

$\text{values}(A)$: tập các giá trị của A

S_v là tập con của S bao gồm các mẫu có giá trị của A bằng v

$|S|$ số phần tử của S

Ví dụ 4.1:

Câu 4.1:

Câu 4 (3 điểm)

Cho dữ liệu huấn luyện như trong bảng, trong đó Loại, Độ ồn, KL là thuộc tính, f là nhãn phân loại.

a) Hãy xác định nhãn cho ví dụ:

Loại = Piano, **Độ ồn** = To, **KL** = Nặng

bảng phương pháp phân lớp Bayes đơn giản (chi rõ các xác suất điều kiện thành phần).

b) Hãy xác định nút gốc cho cây quyết định sử dụng thuật toán ID3.

Chú ý: Trong trường hợp có các thuộc tính với cùng độ ưu tiên thì chọn thuộc tính theo thứ tự từ trái sang phải, tức là: Loại, Độ ồn, KL.

Loại	Độ ồn	KL	f
Trống	To	Nặng	-
Ghita	To	Nhỏ	+
Trống	Nhỏ	Nhỏ	-
Piano	Nhỏ	Nặng	-
Ghita	Nhỏ	Nặng	+
Piano	To	Nhỏ	+
Piano	Nhỏ	Nhỏ	-
Trống	Nhỏ	Nặng	-