Bài 3: Tô màu đồ thị

Tô màu đỉnh, với số màu ít nhất, sau cho 2 đỉnh nối với nhau được tô khác màu

Thuật toán tô màu tuần tự như sau (thuật toán tham lam)
Lặp lại cho đến khi tô hết các đỉnh
[
- Giả sử bạn đã dùng k màu
- Chọn đỉnh có bậc cao nhất
- k > 0
{ Chọn màu từ 1 đến k để tô cho đỉnh đó.
- Nếu tồn tại màu i khác với màu các đỉnh kề với nó thì chọn màu i.
- Nếu k tồn tại màu i thì đánh dấu đỉnh đó chưa tô màu.
}
Nếu đỉnh chưa tô màu. Tô màu cho đỉnh đó màu mới k +1.
]

Tô màu	1	3	2	3	2	1	2	3
	A	B	C	D	E	F	G	H
A		1	1	1				
B	1		1		1			
C	1	1		1		1		
D	1		1			1	1	
E		1				1		1
F			1	1	1		1	1
G				1		1		1
H					1	1	1	
Bậc	3	3	4	4	3	5	3	3

Khởi đầu: Đỉnh(Bậc): A(3), B(3), C(4), D(4), E(3), F(5), G(3), H(3)
Tô màu lần 1: chọn đỉnh có bậc lớn nhất F(5)
Hạ bậc lần 1: A(3), B(3), C^1(3), D^1(3), E^1(2), F(0), G^1(2), H^1(2) (Đỉnh mới tô màu bậc = 0, các đỉnh nối trực tiếp với đỉnh vừa tô giảm 1 bậc và các đỉnh này được đánh dấu để k tô lại màu 1)
Tô màu lần 2: chọn đỉnh có bậc lớn nhất khác F: A(3), B(3), C^1(3), D^1(3), E^1(2), G^1(2), H^1(2) ---> Do nhiều đỉnh có bậc là 3, là bậc lớn nhất. Nên chọn theo thứ tự là A(3). Vì A k bị đánh dấu, nên ưu tiên tô màu 1 cho A
Hạ bậc lần 2: A(0), B^1(2), C^1(2), D^1(2), E^1(2), G^1(2), H^1(2) các đỉnh có cung nối trực tiếp đến A giảm đi 1 bậc và được đánh dấu k tô màu 1
Tô màu lần 3: chọn đỉnh có bậc lớn nhất khác F, A: B^1(2), C^1(2), D^1(2), E^1(2), G^1(2), H^1(2) ---> Các đỉnh có bậc bằng nhau, nên chọn đỉnh nào cũng được ----> Chọn đỉnh B, do đỉnh B k được tô màu 1, nên tô màu 2 cho B
Hạ bậc lần 3: B(0), $C^{1,2}$(2), D^1(2), $E^{1,2}$(1), G^1(2), H^1(2) các đỉnh có cung nối trực tiếp đến B giảm đi 1 bậc và được đánh dấu k tô màu 1 và màu 2
Tô màu lần 4 : chọn đỉnh có bậc lớn nhất khác F, A, B --->$C^{1,2}$(2), D^1(2), $E^{1,2}$(1), G^1(2), H^1(2)

Bài 6: Bài toán đèn giao thông (Thuật toán tô màu)

Hãy xây dựng các cột đèn sao cho việc lưu thông không bị giao nhau (số màu đèn là bao nhiêu).

- Lưu ý: Đường nối EC chỉ có tuyến một chiều EC.
- Gợi ý:
  - Xác định tại giao lộ có bao nhiêu tuyến đường:
    - Từ A: AB AC AD
    - Từ B: BA BC BD
    - Từ D: DA DB DC
    - Từ E: EA EB EC ED
  - Lấy 13 tuyến đường làm đỉnh đồ thị.
  - Cung nối những tuyến đường không thể cùng đi một lúc
    - Các tuyến đường giao nhau: EC, AD, DA; EB, AC, AD, DA; AC, EB, BD,DB
    - Các tuyến đường ngược nhau: AB,BC; ED,DC; EA, AB;
  - BA, DC, ED: không giao nhau với các tuyến khác (được phép rẻ phải).
  - Các tuyến cùng đỉnh xuất phát hay cùng đích thì không giao nhau: ED,EA; BC, BA; BC,DC; BA, EA;
  - Các tuyến song song thì không giao nhau (AB và BA, AC và CA,...)
- Xây dựng ma trận M các tuyến đường với M[i][j] = 1, nếu 2 tuyến không thể cùng đi một lúc

Tô màu	1	1	1	1	2	2	3	3	1	2	4	4	1
	AB	AC	AD	BA	BC	BD	DA	DB	DC	EA	EB	EC	ED
AB					1	1	1			1			
AC						1	1	1		1	1		
AD										1	1	1	
BA													
BC	1							1			1		
BD	1	1					1				1	1	
DA	1	1				1					1	1	
DB		1			1							1	
DC													
EA	1	1	1										
EB		1	1		1	1	1						
EC			1			1	1	1					
ED													
Bậc	4	5*	3	0	3	5**	5***	3	0	3	5****	4	
Hạ bậc	3	0	2	0					0				0
Hạ bậc					2	0				2			
Hạ bậc							0	2					
Hạ bậc											0	3	

Kết quả dùng 4 màu cho đèn giao thông:

- Màu 1: AB, AC, AD, BA, DC, ED
- Màu 2: BC, BD, EA
- Màu 3: DA, DB
- Màu 4: EB, EC.

Bài 5: (Thuật toán Robinson) (Mệnh đề đối ngẫu: P và ¬P)

1. CMR: ¬p ∨ q, ¬q ∨ r, ¬r ∨ s, ¬u ∨ ¬s → ¬p, ¬ u
2. Cho {p → q, q → r, r → s, p} Hỏi p ∧ s ?
3. Cho{a ∧ b → c, b ∧ c → d, a ∧ b}. Hỏi d ?

Giải a: CMR: ¬p ∨ q, ¬q ∨ r, ¬r ∨ s, ¬u ∨ ¬s → ¬p, ¬ u

B3: { ¬p ∨ q, ¬q ∨ r, ¬r ∨ s, ¬u ∨ ¬s , **p** , **u** }
B4 : Có tất cả 6 mệnh đề nhưng chưa có mệnh đề nào đối ngẫu nhau.
B5 : tuyển một cặp mệnh đề (chọn hai mệnh đề có biến đối ngẫu). Chọn 2 mệnh đề đầu : ¬p ∨ q , ¬q ∨ r ¬p ∨ r
Danh sách mệnh đề thành : { ¬p ∨ r , ¬r ∨ s , ¬u ∨ ¬s , p , u } Chưa có mệnh đề đối ngẫu.

Tuyển tiếp hai cặp mệnh đề đầu tiên ¬p ∨ r , ¬r ∨ s ¬p ∨ s
Danh sách mệnh đề thành { ¬p ∨ s , ¬u ∨ ¬s , p , u } Vẫn chưa có 2 mệnh đề đối ngẫu

Tiếp tục hai cặp mệnh đề đầu tiên ¬p ∨ s , ¬u ∨ ¬s ¬p ∨¬u
Danh sách mệnh đề thành : {¬p ∨ ¬u , p , u } Vẫn chưa có hai mệnh đề đối ngẫu

Tiếp tục với hai cặp mệnh đề : ¬p ∨ ¬u , u ¬p
Danh sách mệnh đề trở thành : { ¬p, p } Có hai mệnh đề đối ngẫu nên biểu thức ban đầu đã được chứng minh.

Giải b: Cho {p → q, q → r, r → s, p} Hỏi p ∧ s ?

Biến đổi: p → q = ¬p ∨ q
q → r = ¬q ∨ r
r → s = ¬r ∨ s

B1: Phát biểu có dạng chuẩn: ¬p ∨ q, ¬q ∨ r, ¬r ∨ s, p → p ∧ s
B2: Chuyển vế kết luận: { ¬p ∨ q, ¬q ∨ r, ¬r ∨ s, p ∨ p, **¬p ∨ ¬s** }
B3: Tuyển từng cặp mệnh đề, xét tính đối ngẫu:

- ¬p ∨ q , ¬q ∨ r , ¬r ∨ s , p ∨ p , ¬p ∨ ¬s
- ¬p ∨ r
- ¬p ∨ r , ¬r ∨ s , p ∨ p , ¬p ∨ ¬s
- ¬p ∨ s
- ¬p ∨ s , p ∨ p , ¬p ∨ ¬s
- s ∨ p
- s ∨ p , ¬p ∨ ¬s
- s ∨ ¬s Được chứng minh.

Giải c: Cho {a ∧ b → c, b ∧ c → d, a ∧ b}. Hỏi d ?

Biến đổi: a ∧ b → c = ¬(a ∧ b) ∨ c = ¬a ∨ ¬b ∨ c
b ∧ c → d = ¬(b ∧ c) ∨ d = ¬b ∨ ¬c ∨ d

B1: Phát biểu có dạng chuẩn: ¬a ∨ ¬b ∨ c, ¬b ∨ ¬c ∨ d, a ∧ b → d
B2: Chuyển vế kết luận: { ¬a ∨ ¬b ∨ c, ¬b ∨ ¬c ∨ d, a ∧ b, **¬d** }
B3: Tuyển từng cặp mệnh đề, tính đối ngẫu:

- ¬a ∨ ¬b ∨ c , ¬b ∨ ¬c ∨ d , a ∧ b, ¬d
- ¬a ∨ ¬b ∨ d
- ¬a ∨ ¬b ∨ d, a ∧ b, ¬d
- ¬(a ∧ b) ∨ d , (a ∧ b), ¬d
- d, ¬d Được chứng minh.

Thuật giải Robinson

B1: Phát biểu lại giả thiết và kết luận của vấn đề theo dạng chuẩn sau : GT_1, GT_2, ..., GT_n KL_1, KL_2, ..., KL_m
Trong đó các GT_i và KL_i là các mệnh đề được xây dựng từ các biến mệnh đề và 3 phép nối cơ bản : ∧ (dấu tuyển), ∨ (dấu hội) , ¬ (dấu bù)

B2 : Nếu GT_i có phép ∧, KL_i có phép ∨ thì thay thế bằng dấu ","

B3:(Khử dấu →) Biến đổi dòng chuẩn ở B1 về thành danh sách mệnh đề như sau:
{ GT_1, GT_2, ..., GT_n , ¬KL_1, ¬KL_2, ..., ¬KL_m }

B4 : Nếu trong danh sách mệnh đề ở bước 2 có 2 mệnh đề đối ngẫu nhau thì bài toán được chứng minh. Ngược lại thì chuyển sang B4. (a và ¬a gọi là hai mệnh đề đối ngẫu nhau)

B5 : Xây dựng một mệnh đề mới bằng cách tuyển một cặp mệnh đề trong danh sách mệnh đề ở bước 2. Nếu mệnh đề mới có các biến mệnh đề đối ngẫu nhau thì các biến đó được loại bỏ.
Ví dụ : p ∨ **¬q** ∨ ¬r ∨ s ∨ **q**
Hai mệnh đề **q**, **¬q** là đối ngẫu nên sẽ được loại bỏ
p ∨ ¬r ∨ s

B6 : Thay thế hai mệnh đề vừa tuyển trong danh sách mệnh đề bằng mệnh đề mới.
Ví dụ : { p ∨ **¬q** , ¬r ∨ s ∨ **q** , w ∨ r , s ∨ q }
{p ∨ ¬r ∨ s , w ∨ r , s ∨ q}

B7 : Nếu không xây dựng được thêm một mệnh đề mới nào và trong danh sách mệnh đề không có 2 mệnh đề nào đối ngẫu nhau thì vấn đề không được chứng minh.

Bài 4: Chứng minh

1. Cho {p → q, q → r}. Kết luận: {p → r}
2. Cho {(a ∧ b) → c, (b ∧ c) → d, ¬d. CM: a → b

Giải a:
Ta có: p → q = ¬p ∨ q
q → r = ¬q ∨ r
p → r = ¬p ∨ r

B1: Dạng chuẩn: ¬p ∨ q, ¬q ∨ r → ¬p ∨ r
B3: ¬p ∨ q, ¬q ∨ r → ¬p, r
B4: Phân thành 2 dòng: (1) ¬p, ¬q ∨ r → ¬p, r (CM)
(2) q, ¬q ∨ r → ¬p, r

B2: Chuyển vế: (2) p, q, ¬q ∨ r → r
B4: Phân thành 2 dòng: (1') p, q, r → r (CM)
(2") p, q, ¬q → r
B2 : Chuyển vế (2') : p, q → , r, q (CM)
KL : Tất cả các nhánh con đều được chứng minh Bài toán đã được chứng minh

---

Giải b: Ta có : (a ∧ b) → c = ¬(a ∧ b) ∨ c = ¬a ∨ ¬b ∨ c
(b ∧ c) → d = ¬(b ∧ c) ∨ d = ¬b ∨ ¬c ∨ d
a → b = ¬a ∨ b

B1 : Dạng chuẩn : ¬a ∨ ¬b ∨ c, ¬b ∨ ¬c ∨ d, ¬d → ¬a ∨ b
B2: Chuyển vế: ¬a ∨ ¬b ∨ c, ¬b ∨ ¬c ∨ d, → ¬a ∨ b, d
B3: ¬a ∨ ¬b ∨ c, ¬b ∨ ¬c ∨ d, → ¬a , b, d
B4: Phân dòng:

- (1) ¬a , ¬b ∨ ¬c ∨ d, → ¬a , b, d (CM)
- (2) ¬b ∨ c, ¬b ∨ ¬c ∨ d, → ¬a , b, d

B2: Chuyển vế (2): a, ¬b ∨ c, ¬b ∨ ¬c ∨ d → b, d
B4: Phân dòng:
(1') a, ¬b ∨ c, ¬b ∨ ¬c → b, d
(2') a, ¬b ∨ c, d, → b, d (CM)
B2: (1') a, ¬b ∨ c, ¬(b ∧ c) → b, d
Chuyển vế: a, ¬b ∨ c → b, d, b ∧ c
B4: Phân dòng:
(1") a, ¬b → b, d, b ∧ c
(2") a, c → b, d, b ∧ c
B2: Chuyển vế (1") a → (b),b, d, b ∧ c
(2") a, c → b, d, b ∧ c
B4: Phân dòng:
(1"') a → b, d (không CM)
(2"') a → b, d, c

Kết luận: Bài toán không được chứng minh.

Thuật giải Vương Hạo

B1: Phát biểu lại giả thiết và kết luận của vấn đề theo dạng chuẩn sau : GT_1, GT_2, ..., GT_n KL_1, KL_2, ..., KL_m
Trong đó các GT_i và KL_i là các mệnh đề được xây dựng từ các biến mệnh đề và 3 phép nối cơ bản : ∧ (dấu tuyển), ∨ (dấu hội) , ¬ (dấu bù)

Phủ định của phủ định	¬ (¬ p) ≡ p
	(p ∨ q) ≡ (¬ p → q)
Tương phản	(p → q) ≡ (¬ p → ¬ q)
De Morgan	¬ (p ∨ q) ≡ (¬ p ∧ ¬ q)
	¬ (p ∧ q) ≡ (¬ p ∨ ¬ q)
Giao hoán	(p ∧ q) ≡ (q ∧ p)
	(p ∨ q) ≡ (q ∨ p)
Kết hợp	(p ∧ q) ∧ r ≡ (p ∧ (q ∧ r))
	(p ∨ q) ∨ r ≡ (p ∨ (q ∨ r))
Phân phối	p ∨ (q ∧ r) ≡ (p ∨ q) ∧ (p ∨ r)
	p ∧ (q ∨ r) ≡ (p ∧ q) ∨ (p ∧ r)

B2 (Khử dấu ¬) Chuyển về các GT_i và KL_i có dạng phủ định.
Ví dụ : p ∨ q, ¬(r ∧ s), ¬g, p ∨ r → s, ¬p
p ∨ q, p ∨ r, p → (r ∧ s), g, s

B3 (Khử dấu ∧, ∨): Nếu GT_i có phép ∧, KL_i có phép ∨ thì thay thế bằng dấu ","
Ví dụ : p ∧ q, r ∧ (¬p ∨ s) → ¬q, ¬s
p, q, r, ¬p ∨ s → ¬q, ¬s

B4 : Nếu GT_i có phép ∨ hay ở KL_i có phép ∧ thì tách thành hai dòng con.

Ví dụ : p, ¬p ∨ q → q p, ¬p →q
p, q →q

B5 : Một dòng được chứng minh nếu tồn tại chung một mệnh đề ở ở cả hai phía.
Ví dụ : p, q → q được chứng minh
p, ¬p → q p → p, q

B6 :
a) Nếu một dòng không còn phép nối ∨ hoặc ∧ ở cả hai vế và ở 2 vế không có chung một biến mệnh đề thì dòng đó không được chứng minh.
b) Một vấn đề được chứng minh nếu tất cả dòng dẫn xuất từ dạng chuẩn ban đầu đều được chứng minh.

Bài 1: Tính chi phí hành trình tốt nhất (tiết kiệm nhất): (Thuật toán GTS_2- Greedy)

	A	B	C	D	E	F
A	∞	20	42	30	6	25
B	12	∞	16	7	33	19
C	23	5	∞	28	14	9
D	12	9	24	∞	31	15
E	14	7	21	15	∞	45
F	36	15	16	5	205	∞

- Với số thành phố xuất phát p = 4:
  - 一. tp 1 xuất phát từ A,
  - 二. tp2 từ B,
  - 八. tp 4 từ D
  - 三. tp 6 từ F
- tương ứng với 4 hàng v_1=A, v_2=B, v_3=D, v_4=F.

GIẢI:

- Bước 1: cost = ∞; // Tổng trọng số của cung (chi phí đi đường)
  - Best = {}; // lộ trình tiết kiệm nhất
  - k = 0; // duyệt lần lượt các điểm xuất phát
- Bước 2: Do k=0 <p → Bước 3
  Bước 3: k = 1
  - Gọi GTS_1(1)
    - T_1 = A → E → B → D → F → C → A (lộ trình v_1: bắt đầu từ tp 1)
    - C_1 = 6 + 7 + 7 + 15+ 16+ 23 = 74 (chi phí cho lộ trình v_1)
  - Bước 4: do C_1< cost cost=74; best=T_1;
- Bước 2: Do k = 1 <p → Bước 3
  - Bước 3: k = 2
    - Gọi GTS_1(2)
      - T_2 = B → D → A → E → C → F → B (lộ trình v_2: bắt đầu từ tp 2)
      - C_2 = 7 + 12 + 6 + 21 + 9 + 15 = 70 (chi phí cho lộ trình v_2)
  - Bước 4: do C_2 < cost (C_1=74) cost =70; best=T_2
- Bước 2: Do k=2 <p → Bước 3
  - Bước 3: k=3
    - Gọi GTS_1(4)
      - T_3 = D → B → A → E → C → F → D (lộ trình v_3: bắt đầu từ tp 4)
      - C_3 = 9 + 12 + 6 + 21 + 9 + 15 = 72 (chi phí cho lộ trình v_3)
  - Bước 4: do C_3 > cost (C_2=70) cost=70; best=T_2;
- Bước 2: Do k=3 <p → Bước 3
  - Bước 3: k=4
    - Gọi GTS_1(6)
      - T_4 = F → D → B → A → E → C → F (lộ trình v_4: bắt đầu từ tp 6)
      - C_4 = 5 + 9 + 12 + 6 + 21 + 9 = 62 (chi phí cho lộ trình v_4)
  - Bước 4: do C_4 < cost (C_2=70) cost=62; best=T_4;
- Bước 2: do k= 4 = p → dừng
- Kết luận: Hành trình tốt nhất T_4: F → D → B → A → E → C → F với chi phí là 62.

Phát biểu GTS_2

- Bước 1: cost=∞; (giá trị rất lớn)
  - Best={};
  - k=0;
- Bước 2: Nếu k<p thì qua Bước 3, Ngược lại thì dừng;
- Bước 3: Tăng k=k+1;
  - Gọi GTS_1 với thành phố xuất phát là v_k
  - Tính T_k
  - Chi phí C_k
- Bước 4: Cập nhật lại hành trình với chi phí thấp nhất;
  - Nếu C_k< C thì cost=C_k; Best=T_k
- Bước 5: Quay lại Bước 2
  -

BÀI 2: Sắp xếp hội thảo (Thuật toán tô màu)

Giả sử có 9 cuộc mitting a,b,c,d,e,f,g,h,i được tổ chức. Mỗi cuộc mitting được tổ chức trong một buổi. Các cuộc mitting sau không được xảy ra đồng thời: ae, bc, cd, ed, abd, ahi, bhi, dfi, dhi, fgh. Hãy sử dụng thuật toán tô màu tối ưu để bố trí các cuộc mitting vào các buổi sao cho số buổi diễn ra là ít nhất.

Giải:
Xây dựng ma trận M các cuộc mitting diễn ra với:
M[i][j] = 1 , nếu các buổi mitting không được diễn ra đồng thời;
Xác định bậc của các buổi mitting, mitting có bậc cao nhất là mitting đã ghép nhiều buổi nhất
Ưu tiên chọn cuộc mitting có số bậc cao nhất. và hạ bậc các cuộc liên quan..Ta có
Chọn d=7, hạ bậc lần 1: a(4) , b(4), c(1), **d(0)**, e(1), f(3), **g(2)**, h(5), i(4) → tô màu 1: d,g
Chọn h=5, hạ bậc lần 2: a(3) , b(3), **c(1)**, d(0), **e(1)**, f(2), g(2), **h(0)**, i(3) → tô màu 2: h, c, e
Chọn a=3, hạ bậc lần 3: **a(0)** , b(2), c(1), d(0), e(1), **f(2)**, g(2), h(0), i(2) → tô màu 3: a, f
Chọn b=2, hạ bậc lần 4: a(0) , **b(0)**, c(1), d(0), e(1), f(2),g(2), h(0), i(1) → tô màu 4: b
Chọn i=1, Hạ bậc lần 5: a(0) , b(0), c(1), d(0), e(1), f(2),g(2), h(0), **i(0)** → tô màu 5: i

Kết quả tổ chức các buổi mitting (số màu bằng số buổi)
Buổi 1: d, g; Buổi 2: c, e, h; Buổi 3: a, f; Buổi 4: b và Buổi 5: i

Tô màu	3	4	2	1	2	3	1	2	5
	a	b	c	d	e	f	g	h	i
a		1		1	1			1	1
b	1		1	1				1	1
c		1		1					
d	1	1	1		1	1		1	1
e	1			1					
f				1			1	1	1
g						1		1	
h	1	1		1		1	1		1
i	1	1		1		1		1	
Bậc	**5**	**5**	**2**	**7**	**2**	**4**	**2**	**6**	**5**
Hạ bậc (d)	4	4	1	0	1	3	2	5	4
Hạ bậc (h)	3	3	1		1	2	1	0	3
Hạ bậc (a)	0	2	1		0	2	1		2
Hạ bậc (b)		0	0			2	1		1
Hạ bậc (i)									

Câu 1: Cho CSDL như sau:

Ngày	Quang cảnh	Nhiệt độ	Độ ẩm	Gió	Chơi Tennis?
1	Nắng	Nóng	Cao	Thấp	Không đi
2	Âm u	Nóng	Cao	Thấp	Đi
3	Mưa	Lạnh	TB	Cao	Không đi
4	Âm u	TB	Cao	Thấp	Đi
5	Mưa	TB	Cao	Thấp	Đi
6	Mưa	Lạnh	TB	Thấp	Đi
7	Nắng	TB	Cao	Thấp	Không đi
8	Nắng	Lạnh	TB	Thấp	Đi
9	Âm u	Lạnh	TB	Cao	Đi
10	Mưa	TB	TB	Thấp	Đi
11	Nắng	Nóng	Cao	Cao	Không đi
12	Nắng	TB	TB	Cao	Đi
13	Âm u	TB	Cao	Cao	Đi
14	Âm u	Nóng	TB	Thấp	Đi
15	Mưa	TB	Cao	Cao	?

a. Tìm các luật phân lớp dựa trên cây quyết định với độ đo **Information Gain**:

$$Gain(S,A) = Entropy(S) - \sum_{x \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Trong đó:

- Value(A) là tập tất cả các giá trị có thể có đối với thuộc tính A và S_v là tập con của S mà A có giá trị là v

- Với S bao gồm c lớp, thì Entropy của S được tính bằng công thức sau:

$$Entropy(S) = \sum_{i=1}^{c} -P_i \log_2 P_i$$

Ở đây p_i là tỉ lệ của các mẫu thuộc lớp i trong tập S.

b. Cho biết lớp của mẫu #15?

a)

Entropy(S) = -4/14* log2(4/14) - 10/14*log2(10/14) = 0.86

Với từng thuộc tính, ta có độ đo Information Gain tương ứng là:

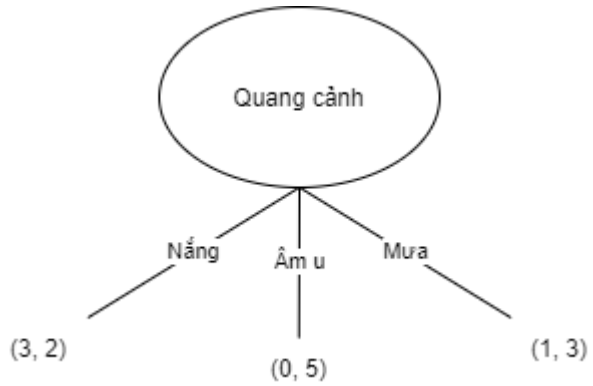
Gain(S, Quang cảnh) = 0.86 - (5/14*(-⅖*log2(⅖) - ⅗*log2(⅗)) + 5/14*0 + 4/14*(-¾*log2(¾)-¼*log2(¼))) = 0.28

Gain(S, Nhiệt độ) = 0.86 - (4/14*(-2/4*log2(2/4) - 2/4*log2(2/4)) + 4/14*(-¾ *log2(¾) - ¼*log2(¼))) = 0.06

Gain(S, Độ ẩm) = 0.86 - (7/14*(-4/7*log2(4/7) - 3/7*log2(3/7)) + 7/14*(-6/7*log2(6/7) - 1/7*log2(1/7))) = 0.07

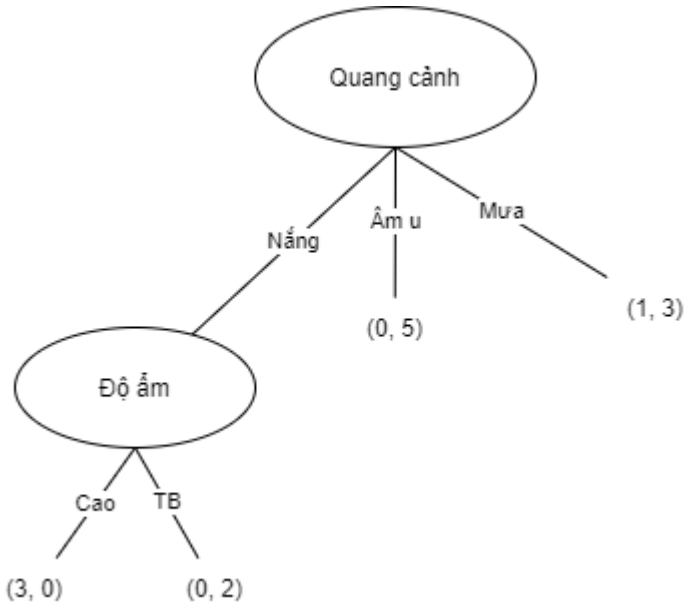
Gain(S, Gió) = 0.86 - (9/14*(-7/9*log2(7/9) - 2/9*log2(2/9)) + 5/14*(-3/5*log2(3/5) - 2/5*log2(2/5))) = 0.02

Vì thuộc tính Quang cảnh có Information Gain lớn nhất nên được chọn để làm nút gốc cho cây quyết định.
Với bộ số (số lượng không đi, số lượng đi), ta có cây quyết định như sau:

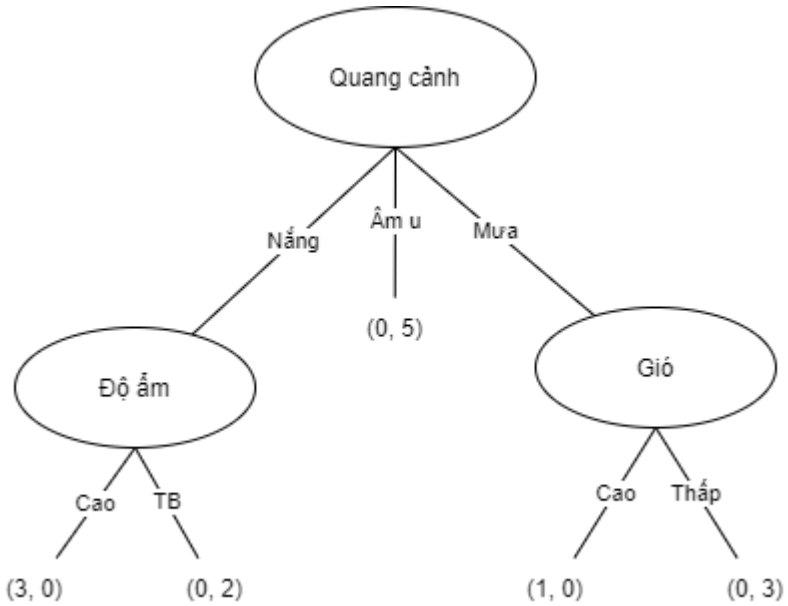


Ta tiếp tục xét với hai nhánh có nút lá chưa đồng nhất:

- Nhánh nắng:
  Entropy(S_nắng) = -⅗ * log2(⅗) - ⅖ * log2(⅖) = 0.97
  Gain(S_nắng, Nhiệt độ) = 0.97 - (⅖*0 + ⅖*(-½*log2(½) - ½*log2 (½)) + ⅕*0) = 0.57
  Gain(S_nắng, Độ ẩm) = 0.97 - (⅗*0 + ⅖*0) = 0.97
  Gain(S_nắng, Gió) = 0.97 - (⅗*(-⅔*log2(⅔)-⅓*log2(⅓)) + ⅔*(-½*log2(½)-½*log2(½))) = 0.02
  Vì thuộc tính Độ ẩm có Information Gain lớn nhất nên được chọn để làm nút tiếp theo nhánh nắng:



- Nhánh mưa:
  Entropy(S_mưa) = -¼*log2(¼) - ¾*log2(¾) = 0.81
  Gain(S_mưa, Nhiệt độ) = 0.81 - (2/4*(-½*log2(½) - ½*log2(½)) + 2/4*0) = 0.31
  Gain(S_mưa, Độ ẩm) = 0.81 - (¾*(-⅔*log2(⅔) - ⅓*log2(⅓)) + ¼*0) = 0.12
  Gain(S_mưa, Gió) = 0.81 - (¾*0 + ⅓*0) = 0.81
  Vì thuộc tính Gió có Information Gain lớn nhất nên được chọn để làm nút tiếp theo nhánh mưa:



Vì tất cả các nốt lá đều đồng nhất nên ta rút ra luật:

- Nếu quang cảnh nắng, độ ẩm cao thì không đi chơi tennis.
- Nếu quang cảnh nắng, độ ẩm trung bình thì đi chơi tennis.
- Nếu quang cảnh âm u thì đi chơi tennis.
- Nếu quang cảnh mưa, gió cao thì không đi chơi tennis.
- Nếu quang cảnh mưa, gió thấp thì đi chơi tennis.

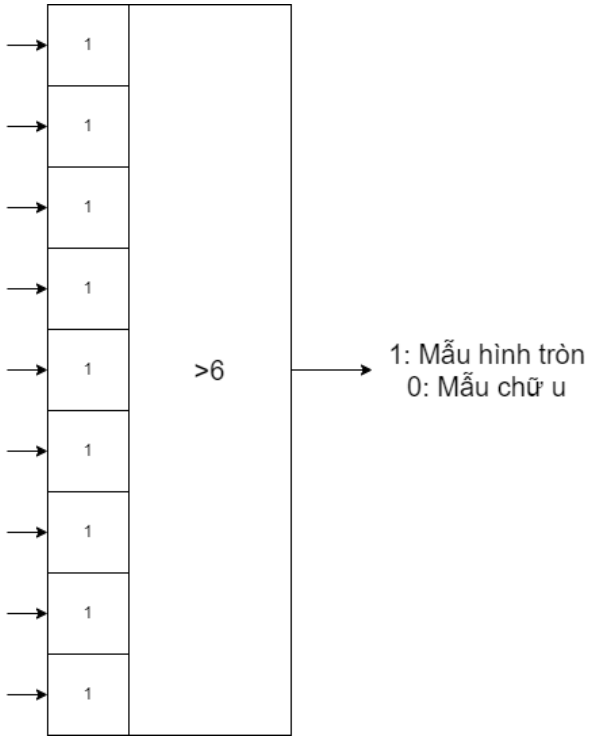
b)

Dựa trên luật vừa rút ra, ta có:

- Mẫu 15 có quang cảnh mưa, gió cao nên sẽ thuộc lớp không đi chơi tennis.

Câu 2. Thiết kế một mạng nơ-ron nhân tạo có thể phân biệt mẫu nào trong số hai mẫu sau đây nằm trong khung nhìn của chúng.

Gọi hình thứ nhất là mẫu chữ U, hình thứ hai là mẫu chứ O, khi đó mạng nơ ron phân biệt với 9 đầu vào tương ứng với 9 ô vuông của mẫu là:



Câu 3: Cho một cơ sở tri thức được biểu diễn dưới dạng logic mệnh đề như sau:

$$KB = \{A, B, A \vee C, K \wedge E \leftrightarrow A \wedge B, \neg C \rightarrow D, E \vee F \rightarrow \neg D\}$$

Kiểm tra các câu sau có rút ra được từ tập cơ sở trên hay không

a) $B \wedge C$?

b) $C \vee E \rightarrow F \wedge B$?

a)

Bước	Công thức	Nguồn gốc
1	A	Cho trước
2	B	Cho trước
3	A v C	Cho trước
4	K ^ E <-> A ^ B	Cho trước
5	~C -> D	Cho trước
6	E v F -> ~D	Cho trước
7	A ^ B	1,2 And-Introduction
8	A ^ B -> K ^ E	4 Biconditional Elimination

9	K ^ E	7, 8 Modus Pones
10	E	9 And-Elimination
11	E v F	10 Or-Elimination
12	~D	11, 6 Modus Pones
13	C	12, 5 Modus Tolens
14	B ^ C	13, 2 And-Introduction

b)
Xét trường hợp A = B = C = E = True, D = F = False, khi đó ta có:

Công thức	Giá trị	Nguồn gốc
A	True	KB
B	True	KB
A v C	True	KB
K ^ E <-> A ^ B	True	KB
~C -> D	True	KB
E v F -> ~D	True	KB
C v E -> F ^ B	False	Kết luận

Vì xảy ra trường hợp KB = True và Kết luận là False nên KB -> Kết luận là không hợp lệ, do đó không thể C v E -> F ^ B không thể rút ra từ tập cơ sở KB.

Câu 1:

Cho các câu sau:

1. Jack sở hữu một con chó.
2. Ai sở hữu một con chó là người yêu động vật.
3. Người nào yêu động vật thì không giết động vật.
4. Jack giết Tuna hoặc Curiosity giết Tuna
5. Tuna là một con mèo.
6. Mọi con mèo đều là động vật.

a) Hãy sử dụng các vị từ sau đây biểu diễn các câu trên về dạng logic bậc nhất.

D(x): "x là con chó" — O (x, y): "x sở hữu y"

L(x): "x là người yêu động vật" — A(x): "x là động vật"

K (x, y): "x giết y" — C(x): "x là con mèo"

b) Từ các câu trên, hãy chứng minh xem Curiosity có giết Tuna hay không?

a)

1. ∃x.D(x) ^ O(Jack, x)
2. ∀x.[∃y.D(y) ^ O(x, y)] => L(x)
3. ∀x.L(x) => (∀y.A(y) => ~K(x, y))
4. K(Jack, Tuna) v K(Curiosity, Tuna)
5. C(Tuna)
6. ∀x.C(x)=>A(x)

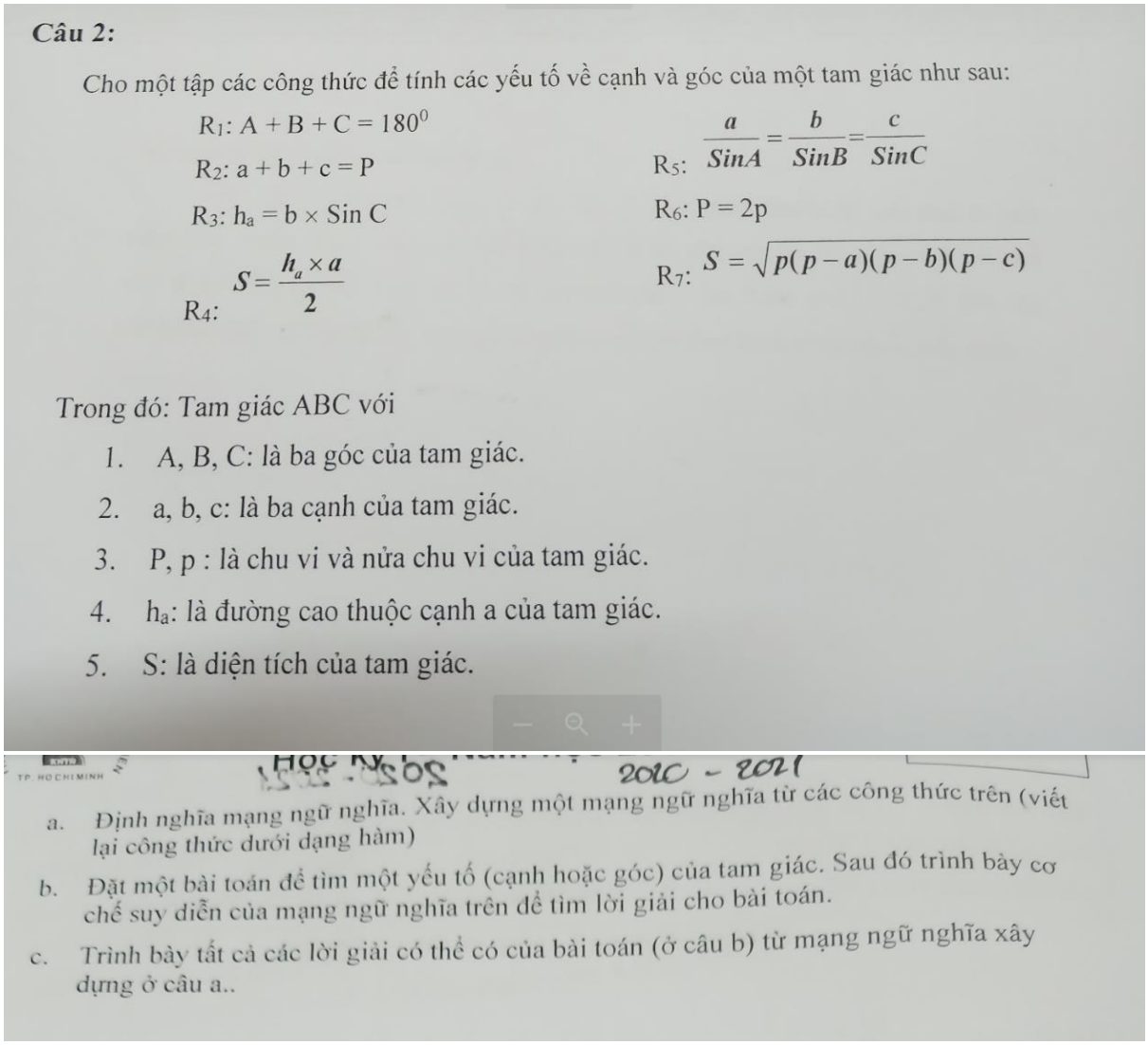
b)

Giả sử Curiosity giết Tuna hay ta cần kết luận K(Curiosity, Tuna), ta có

Số thứ tự	Mệnh đề	Nguồn gốc
1	D(x) ^ O(Jack, x)	Tiền đề
2	~D(y) v ~O(x,y) v L(x)	Tiền đề
3	~L(x) v ~A(y) v ~K(x, y)	Tiền đề
4	K(Jack, Tuna) v K(Curiosity, Tuna)	Tiền đề
5	C(Tuna)	Tiền đề
6	~C(x) v A(x)	Tiền đề
7	~K(Curiosity, Tuna)	Phủ định của kết luận
8	K(Jack, Tuna)	7, 4
9	A(Tuna)	5, 6 theta = {x/Tuna}
10	~L(x) v ~K(x, Tuna)	9, 2 theta = {y/ Tuna}
11	~(D(y) ^ O(x,y)) v L(x)	2

12	L(Jack)	11, 1 theta = {x/Jack, y/x}
13	~K(Jack, Tuna)	12, 10 theta = {Jack/x}
14	False	13, 8

Do đó Curiosity giết Tuna <- Hồi nãy mình kết luận bị nhầm là Curiosity không giết Tuna, rất xin lỗi mọi người :((



Câu 2:

Cho một tập các công thức để tính các yếu tố về cạnh và góc của một tam giác như sau:

R_1: $A + B + C = 180^0$

R_2: $a + b + c = P$

R_3: $h_a = b \times \text{Sin } C$

R_4: $S = \frac{h_a \times a}{2}$

R_5: $\frac{a}{SinA} = \frac{b}{SinB} = \frac{c}{SinC}$

R_6: $P = 2p$

R_7: $S = \sqrt{p(p-a)(p-b)(p-c)}$

Trong đó: Tam giác ABC với

1. A, B, C: là ba góc của tam giác.
2. a, b, c: là ba cạnh của tam giác.
3. P, p : là chu vi và nửa chu vi của tam giác.
4. h_a: là đường cao thuộc cạnh a của tam giác.
5. S: là diện tích của tam giác.

TP. HỒ CHÍ MINH

2020 - 2021

a. Định nghĩa mạng ngữ nghĩa. Xây dựng một mạng ngữ nghĩa từ các công thức trên (viết lại công thức dưới dạng hàm)

b. Đặt một bài toán để tìm một yếu tố (cạnh hoặc góc) của tam giác. Sau đó trình bày cơ chế suy diễn của mạng ngữ nghĩa trên để tìm lời giải cho bài toán.

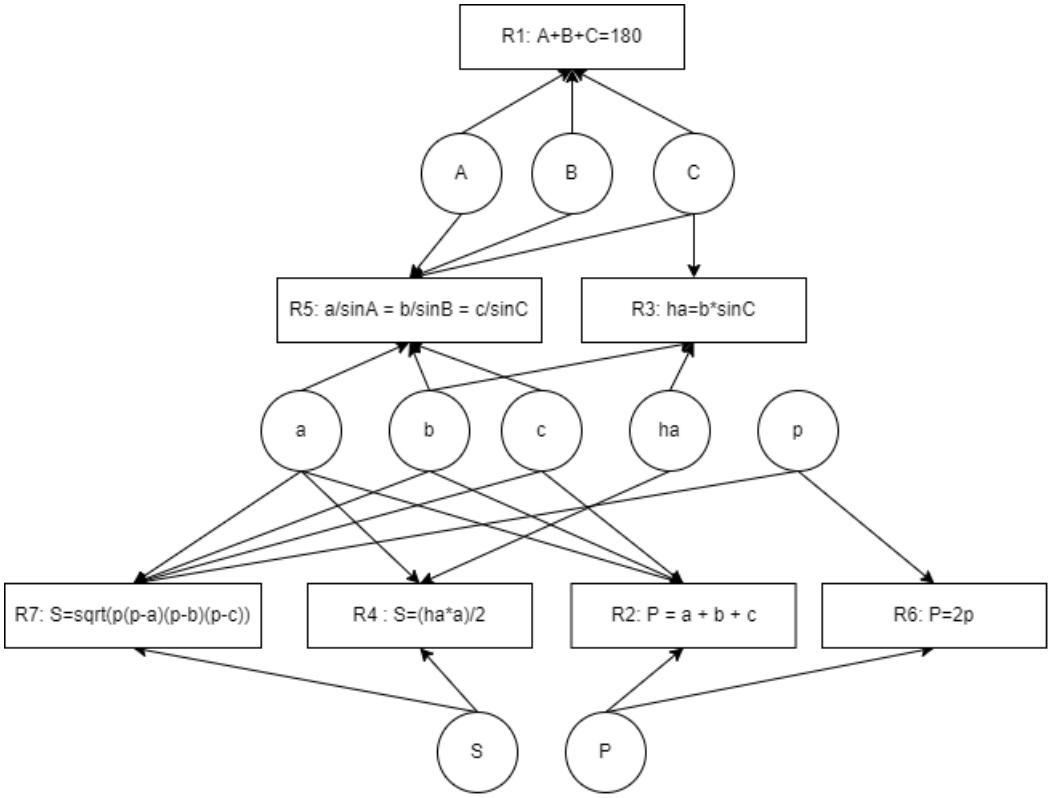
c. Trình bày tất cả các lời giải có thể có của bài toán (ở câu b) từ mạng ngữ nghĩa xây dựng ở câu a..

a)

Mạng ngữ nghĩa là sự thay thế logic vị từ để biểu diễn tri thức dưới dạng mạng đồ thị với:

- Các nút đại diện cho các đối tượng.
- Các cung mô tả các mối quan hệ giữa các đối tượng đó.

Mạng ngữ nghĩa công thức trên:



b)

Cho độ dài b, ha. Tìm góc C:

Cơ chế suy diễn, kích hoạt b, ha, C

- Suy luận:

1. b, ha -> R3
2. R3 -> C

c)

Lời giải của câu b) là lời giải duy nhất của bài toán.

Câu 3. *Áp dụng thuật giải A*cho bài toán "Taci" với cấu hình như sau:*

Start:

2	8	3
1	6	4
7		5

(a)

Goal:

1	2	3
8		4
7	6	5

(b)

Với hàm Heuristic là: Seq(n) = 1 nếu ô ở giữa khác 0 và 2 nếu ô ở biên không tuân theo *thứ tự tăng (theo chiều kim đồng hồ của trạng thái đích).*

Gọi Sx là trạng thái x của bài toán.

hàm g(Sx) mang giá trị là lần đẩy thứ n của trò chơi

Khi đó theo thuật toán A*, ta có hàm f(x) = g(Sx) + Seq(n). Xét:

S0

2 8 3

1 6 4

7 5

g(S0) = 0

Seq(S0) = 4*2 + 1 = 9

f(S0) = 9
Các trường hợp cho đẩy lần 1 của S0:
S1
2 8 3
1 6 4
7 5
g(S1) = 1
Seq(S1) = 5*2 + 1 = 11
f(S1) = 12

S2
2 8 3
1 4
7 6 5
g(S2) = 1
Seq(S2) = 3*2 = 6
f(S2) = 7

S3
2 8 3
1 6 4
  7 5
g(S3) = 1
Seq(S3) = 5*2 + 1 = 11
f(S3) = 12

Vì S2 có giá trị f(S2) = 7 nhỏ nhất nên S2 sẽ là lần đẩy thứ nhất
Các trường hợp đẩy lần hai của S2:

S4
2 3
1 8 4
7 6 5
g(S4) = 2
Seq(S4) = 3*2 + 1 = 7
f(S4) = 9

S5
2 8 3
1 4
7 6 5
g(S5) = 2
Seq(S5) = 4*2 + 1 = 9
f(S5) = 11

S6
2 8 3
1 6 4

7 5
g(S6) = 2
Seq(S6) = 4*2 + 1 = 9
f(S6) = 11

S7
2 8 3
  1 4
7 6 5
g(S7) = 2
Seq(S7) = 3*2 + 1 = 7
f(S7) = 9

Vì S4 có giá trị f(S4) = 9 nhỏ nhất nên S4 sẽ là lần đẩy thứ 2
Các trường hợp đẩy lần 3 của S4:

S8
  2 3
1 8 4
7 6 5
g(S8) = 3
Seq(S8) = 2*2 + 1 = 5
f(S8) = 8

S9
2 8 3
1 4
7 6 5
g(S9) = 3
Seq(S9) = 3*2 = 6
f(S9) = 9

S10
2 3
1 8 4
7 6 5
g(S10) = 3
Seq(S10) = 4*2 + 1 = 9
f(S10) = 12

Vì S8 có giá trị f(S8) = 8 nhỏ nhất nên S8 sẽ là lần đẩy thứ 3
Các trường hợp đẩy lần 4 của S8:

S11
1 2 3
  8 4
7 6 5
g(S11) = 4

Seq(S11) = 2 + 1 = 3
f(S11) = 7

S12
2 3
1 8 4
7 6 5
g(S12) = 4
Seq(S12) = 2*3 + 1 = 7
f(S12) = 11

Vì S11 có giá trị f(S11) = 7 nhỏ nhất nên S11 sẽ là lần đẩy thứ 4
Các trường hợp đẩy lần 5 của S11:

S13
1 2 3
8 4
7 6 5
g(S12) = 5
Seq(S12) = 0
f(S12) = 5

S14
  2 3
1 8 4
7 6 5
g(S13) = 5
Seq(S13) = 2*2 + 1 = 5
f(S13) = 10

S15
1 2 3
7 8 4
  6 5
g(S14) = 5
Seq(S14) = 2*2 + 1 = 5
f(S14) = 10

Vì S13 có giá trị f(S13) = 5 nhỏ nhất nên S13 sẽ là lần đẩy thứ 5
Vì S13 là trường hợp đích nên thuật toán kết thúc.

Câu 2.

Sử dụng thuật giải Heuristic (Greedy) tô màu tối ưu trên đồ thị, để giải bài toán sau:

"Giả sử có một hội thảo khoa học, có 9 chủ đề: a, b, c,..... được tổ chức. Mỗi chủ đề diễn ra trong một buổi. Trong đó các chủ đề sau không được diễn ra một cách đồng thời: ae, bc, cd, ed, abd, ahi, bhi, dfi, dhi, fhg.

Hãy bố trí các chủ đề trên vào các buổi, để số buổi diễn ra hội thảo là ít nhất

Gọi V là tập các biến biểu diễn các chủ đề:

V = { a, b, c, d, e, f, g, h, i}

Gọi D_n là tập miền giá trị ứng với biến n:

D_n ∈ N*

Gọi C là ma trận biểu diễn ràng buộc giữa các biến, khi đó mỗi cạnh sẽ nối hai đỉnh không được diễn ra đồng thời, ta có ma trận kề của C là:

	A	B	C	D	E	F	G	H	I
A		1		1	1			1	1
B	1		1	1				1	1
C		1		1					
D	1	1	1		1	1		1	1
E	1			1					
F				1			1	1	1
G						1		1	
H	1	1		1		1	1		1
I	1	1		1		1		1	

Ta chia lịch bằng thuật giải Heuristic (Greedy) tô màu tối ưu bằng cách tìm chủ đề có bậc lớn nhất và xếp buổi gần nhất có thể. Khi đó ta lần lượt thực hiện các bước:

Lần		A	B	C	D	E	F	G	H	I
1	Bậc	5	5	2	7	2	4	2	6	5
	Buổi				1					
2	Bậc	4	4	1		1	3	2	5	4
	Buổi								2	
3	Bậc	3	3	1		1	2	1		3
	Buổi	3								
4	Bậc		2	1		0	2	1		2
	Buổi		4							

5	Bậc			0		0	2	1		1
	Buổi						3			
6	Bậc			0		0		0		0
	Buổi			2						
7	Bậc					0		0		0
	Buổi					2				
8	Bậc							0		0
	Buổi							1		
9	Bậc									0
	Buổi									5

Do đó ta có kết quả bố trí như sau:
Buổi 1: G, D
Buổi 2: C, E, H
Buổi 3: A, F
Buổi 4: B
Buổi 5: I

VỀ CÁC THUẬT TOÁN XÂY DỰNG CÂY QUYẾT ĐỊNH VÀ RÚT GỌN TẬP LUẬT

1. CÂY QUYẾT ĐỊNH

Việc xây dựng các cây quyết định chính là quá trình phát hiện ra các luật phân chia tập dữ liệu đã cho thành các lớp đã được định nghĩa trước. Trong thực tế, tập các cây quyết định có thể có đối với bài toán này rất lớn và rất khó có thể duyệt hết được một cách tường tận. Độ phức tạp tính toán của việc tìm một cây phân lớp quyết định tối ưu là NP([2]).

- Một cây quyết định là một cấu trúc hình cây, trong đó:
- Mỗi đỉnh trong (đỉnh có thể khai triển được) biểu thị cho một phép thử đối với một thuộc tính;
- Mỗi nhánh biểu thị cho một kết quả của một phép thử;
- Các đỉnh lá (các đỉnh không khai triển được) biểu thị các lớp hoặc các phân bố lớp;
- Đỉnh trên cùng trong một cây được gọi là gốc.

Việc sinh cây quyết định bao gồm hai giai đoạn:

(i) Xây dựng cây:

- Tại thời điểm khởi đầu, tất cả các ca (case) dữ liệu học đều nằm tại gốc;
- Các ca dữ liệu được phân chia đệ qui trên cơ sở các thuộc tính được chọn.

(ii) Rút gọn cây:

- Phát hiện và bỏ đi các nhánh chứa các điểm dị thường và nhiễu trong dữ liệu.

Hầu hết các thuật toán dựa vào qui nạp hiện có đều sử dụng phương pháp của Hunt [1] làm thuật toán cơ sở. Dưới đây là mô tả qui nạp phương pháp của Hunt dùng để xây dựng một cây quyết định từ một tập *T* các ca học với các lớp được ký hiệu là $\{C_1,C_2,...,C_k\}$.

Trường hợp 1: *T* chứa một hoặc nhiều ca, tất cả đều thuộc về một lớp đơn C_j: Cây quyết định cho *T* là một lá định dạng lớp C_j.

Trường hợp 2: *T* không chứa ca nào: Cây quyết định cho *T* là một lá, nhưng lớp được gắn với lá này phải được xác định từ các thuộc tính không thuộc *T*.

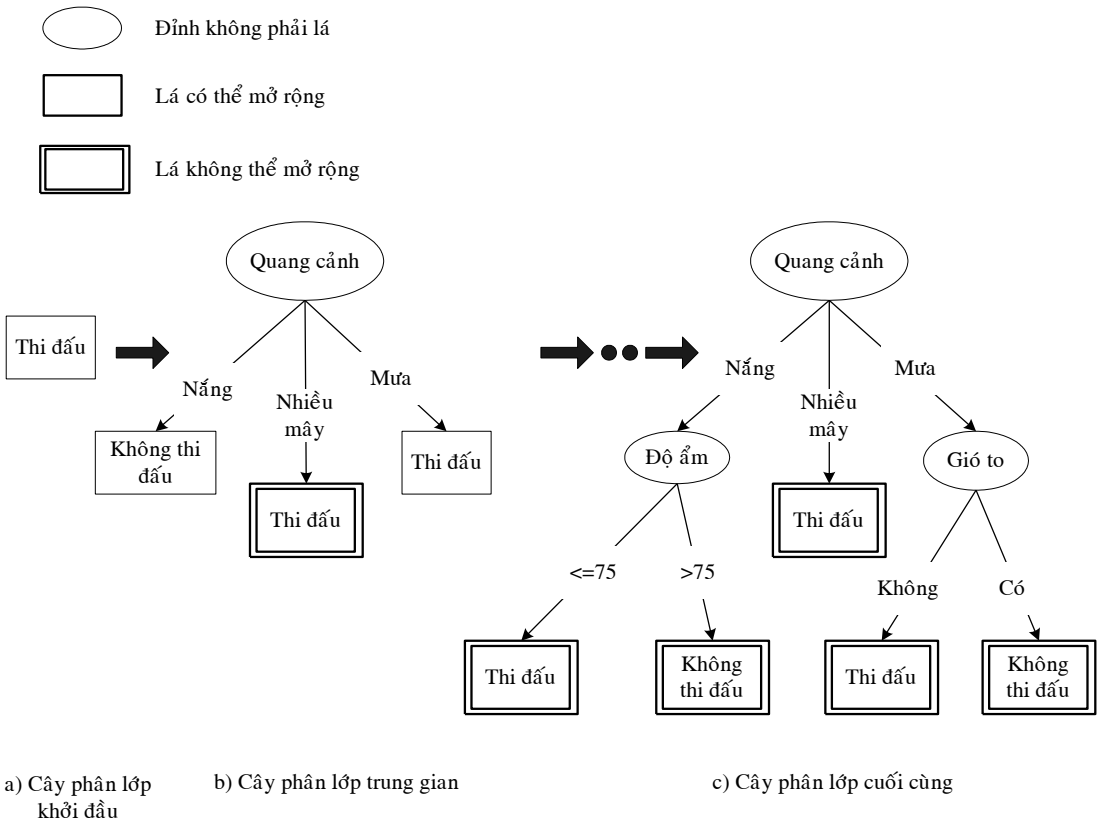
Trường hợp 3: T chứa các ca thuộc về một hỗn hợp các lớp: Một phép thử được lựa chọn dựa vào một thuộc tính đơn có một hoặc nhiều kết quả (giá trị) loại trừ lẫn nhau $\{O_1,O_2,...,O_n\}$. *T* được phân chia thành các tập con $T_1,T_2,...,T_n$, trong đó T_i chứa tất cả các ca trong *T* có kết quả O_i của phép thử đã chọn. Cây quyết định cho *T* gồm một đỉnh quyết định định danh cho phép thử, và một nhánh cho mỗi kết quả có thể có. Cơ chế xây dựng cây này được áp dụng đệ qui cho từng tập con của các ca học.

Bảng 1 là một tập dữ liệu học của một ví dụ về thi đấu tennis với năm thuộc tính và hai lớp (thuộc tính **Ngày** được sử dụng làm định danh cho các ca). Hình 1 chỉ

ra cách làm việc của thuật toán Hunt với tập dữ liệu học này. Trong trường hợp 3 của phương pháp Hunt, một phép thử dựa trên thuộc tính đơn được chọn để khai triển đỉnh hiện hành.

Ngày	Quang cảnh	Nhiệt độ	Độ ẩm (%)	Gió to	Kết quả
N1	Nắng	24	70	Không	Thi đấu
N2	Nắng	27	90	Có	Không thi đấu
N3	Nắng	30	85	Không	Không thi đấu
N4	Nắng	22	95	Không	Không thi đấu
N5	Nắng	20	70	Không	Thi đấu
N6	Nhiều mây	22	90	Có	Thi đấu
N7	Nhiều mây	28	75	Không	Thi đấu
N8	Nhiều mây	18	65	Có	Thi đấu
N9	Nhiều mây	28	75	Không	Thi đấu
N10	Mưa	21	80	Có	Không thi đấu
N11	Mưa	18	70	Có	Không thi đấu
N12	Mưa	24	80	Không	Thi đấu
N13	Mưa	20	80	Không	Thi đấu
N14	Mưa	21	96	Không	Thi đấu

Bảng 1. Một tập dữ liệu học ([1])



a) Cây phân lớp khởi đầu

b) Cây phân lớp trung gian

c) Cây phân lớp cuối cùng

Hình 1. Minh hoạ phương pháp của Hunt

2. THUẬT TOÁN ID3

Thuật toán ID3 (**Quinlan86**) là một trong những thuật toán xây dựng cây quyết định sử dụng information gain để lựa chọn thuộc tính phân lớp các đối tượng. Nó xây dựng cây theo cách từ trên xuống, bắt đầu từ một tập các đối tượng và một đặc tả của các thuộc tính. Tại mỗi đỉnh của cây, một thuộc tính có *information gain* lớn nhất sẽ được chọn để phân chia tập đối tượng. Quá trình này được thực hiện một cách đệ qui cho đến khi một tập đối tượng tại một cây con đã cho trở nên thuần nhất, tức là nó chỉ chứa các đối tượng thuộc về cùng một lớp. Lớp này sẽ trở thành một lá của cây.

Việc lựa chọn một thuộc tính nào cho phép thử là rất quan trọng. Nếu chọn không thích hợp, chúng ta có thể có một cây rất phức tạp. Ví dụ, nếu ta chọn thuộc tính **Nhiệt độ** làm gốc cây thì cây quyết định sẽ có hình dạng như trong Hình 2. Nhưng nếu chọn thuộc tính **Quang cảnh** làm gốc thì ta lại có một cây quyết định rất đơn giản như đã chỉ trong Hình 1. Vậy nên chọn thuộc tính nào là tốt nhất?

Thông thường việc chọn thuộc tính đều dựa vào một độ đo gọi là **Entropy Gains** hay còn gọi là **Information Gains** của các thuộc tính. Entropy của một thuộc tính được tính toán từ các thuộc tính phân lớp. Đối với các thuộc tính rời rạc, cần phải có các thông tin phân lớp của từng giá trị thuộc tính.

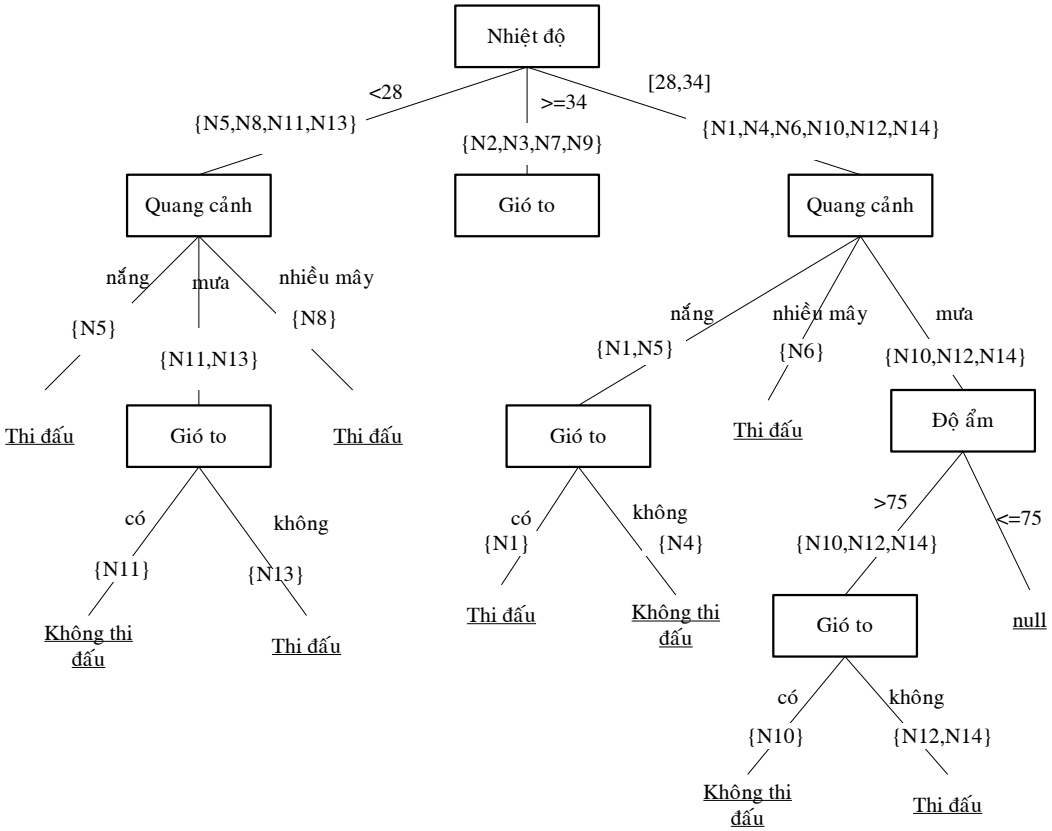
Giá trị thuộc tính	Lớp	
	Thi đấu	Không thi đấu
Nắng	2	3
Nhiều mây	4	0
Mưa	3	2

Bảng 2. Thông tin phân bố lớp của thuộc tính Quang cảnh

Giá trị thuộc tính	Phép thử nhị phân	Lớp	
		Thi đấu	Không thi đấu
65		1	0
	>	8	5
70		3	1
	>	6	4
75		5	1
	>	4	4
78		5	1
	>	4	4
80		7	2
	>	2	3
85		7	3
	>	2	2
90		8	4
	>	1	1
95		8	5
	>	1	0
96		9	5
	>	0	0

Bảng 3. Thông tin phân bố lớp của thuộc tính Độ ẩm

Bảng 2 cho thấy thông tin phân lớp của thuộc tính **Quang cảnh**. Đối với một thuộc tính liên tục, chúng ta phải xét phép thử nhị phân đối với tất cả các giá trị khác nhau của thuộc tính. Bảng 3 chỉ ra thông tin phân lớp của thuộc tính **Độ ẩm**.



Hình 2. Một cây quyết định chọn Nhiệt độ làm gốc

Một khi đã thu nhận được các thông tin phân lớp của tất cả các thuộc tính, chúng ta sẽ tính được Entropy. Một thuộc tính với Entropy lớn nhất sẽ được chọn làm một phép thử để khai triển cây.

2.1. Hàm Entropy

Hàm Entropy xác định tính không thuần khiết của một tập các ca dữ liệu bất kỳ. Chúng ta gọi *S* là tập các ca dương tính (ví dụ Thi đấu) và âm tính (ví dụ Không thi đấu), $P_{\oplus}$ là tỉ lệ các ca dương tính trong S, $P_{\ominus}$ là tỉ lệ các ca âm tính trong *S*.

$$Entropy(S) = -P_{\oplus}\log_2 P_{\oplus} - P_{\ominus}\log_2 P_{\ominus}$$

Ví dụ 1. Trong Bảng 1 của ví dụ thi đấu tennis, tập S có 9 ca dương và 5 ca âm (ký hiệu là [9+,5-]).

$$Entropy(S) = Entropy([9+,5\text{-}]) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940$$

Nhận xét. Entropy bằng 0 nếu tất cả các ca trong S đều thuộc về cùng một lớp. Chẳng hạn như, nếu tất cả các ca đều dương thì $P_{\oplus} = 1$ và $P_{\ominus} = 0$, do vậy:

$$Entropy(S) = -1\log_2(1) - 0\log_2(0) = 0$$

Entropy bằng 1 nếu tập *S* chứa số ca dương và âm bằng nhau. Nếu số các ca này khác nhau thì Entropy nằm giữa 0 và 1.

Trường hợp tổng quát, nếu *S* bao gồm *c* lớp, thì Entropy của *S* được tính bằng

công thức sau:

$$Entropy(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

trong đó P_i là tỉ lệ của các ca thuộc lớp i trong tập S.

2.2. Độ đo (Informatic Gain):

Độ đo, đo mức độ hiệu quả của một thuộc tính trong bài toán phân lớp dữ liệu. Đó chính là sữ rút gọn mà ta mong đợi khi phân chia các ca dữ liệu theo thuộc tính này. Nó được tính theo công thức sau đây:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

trong đó Value(A) là tập tất cả các giá trị có thể có đối với thuộc tính A, và S_v là tập con của S mà A có giá trị là v.

Ví dụ 2.

Value(Gió to)={true,false}, S=[9+,5-]

S_{true}, là đỉnh con với giá trị là "true", bằng [2+,3-]

S_{false}, là đỉnh con với giá trị là "false", bằng [7+,2-]

$$\text{Gain(S,Gió to)} = Entropy(S) - \sum_{v \in \{true,\, false\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$= Entropy(S) - \frac{5}{14} * Entropy(S_{true}) - \frac{9}{14} * Entropy(S_{false})$$

$$= 0.940 - \frac{5}{14} * 0.97 - \frac{9}{14} * 0.764$$

$$= 0.1024$$

Tương tự như vậy, ta có thể tính được độ đo cho các thuộctính còn lại của ví dụ trong Bảng 1. Đối với thuộc tính **Độ ẩm**, ta lấy độ ẩm 75% để chia các ca thành hai phần, một phần ứng với các ca có độ ẩm $\leq 75\%$ được gọi là có độ ẩm Bình thường ([5+,1-]), phần còn lại được gọi là có độ ẩm Cao([4+,4-]). Còn đối với thuộc tính Nhiệt độ, ta sẽ chia thành ba mức, các ngày có nhiệt độ nhỏ hơn 21^0 được gọi là Lạnh (4 ngày), các ngày có nhiệt độ lớn hơn hay bằng 21^0 đến nhỏ hơn 27^0 được gọi là Ấm (6 ngày), và còn lại là những ngày có nhiệt độ lớn hơn hoặc bằng 27^0 được gọi là Nóng (4 ngày).

Gain(S,Quang cảnh)=0.246

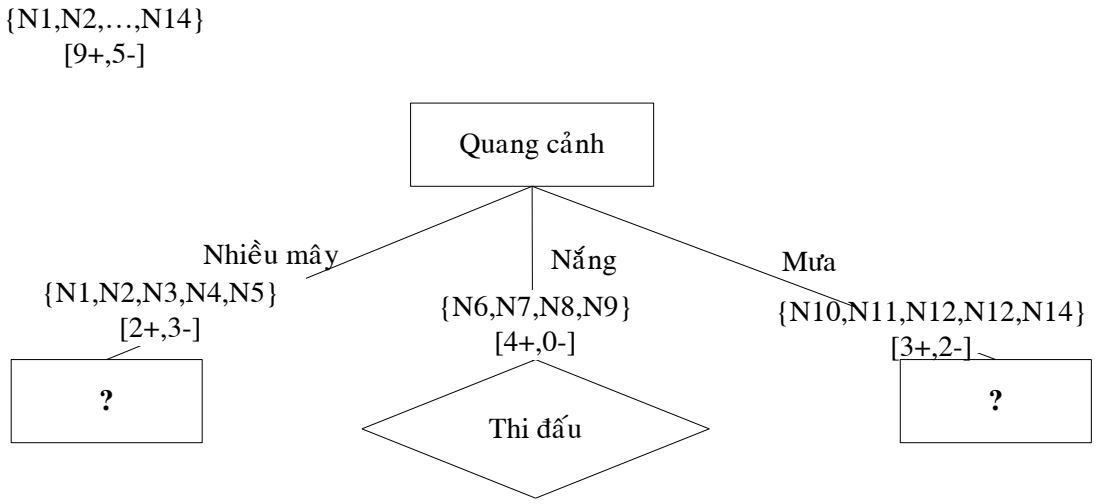
Gain(S,Gió to) =0.1024.

Gain(S,Nhiệt độ)=0.029

Gain(S,Độ ẩm)=0.045

Từ đây ta thấy rằng **độ đo** của S đối với thuộc tính **Quang cảnh** là lớn nhất

trong số 4 thuộc tính. Như vậy, có thể quyết định chọn **Quang cảnh** làm thuộc tính đầu tiên để khai triển cây. Hình 3 là khai triển của cây quyết định theo thuộc tính **Quang cảnh**.



Hình 3. Khai triển cây theo thuộc tính đã chọn

Tương tự như vậy, ta có thể tiến hành triển khai các nút ở mức tiếp theo.

$S_{nắng}$=*{N1,N2,N3,N4,N5}*

Entropy($S_{nắng}$)= $-\frac{2}{5}*\log_2\frac{2}{5}-\frac{3}{5}*\log_2\frac{3}{5}=0.097$

Gain($S_{nắng}$, *Độ ẩm*)= $0.970-\frac{3}{5}*0.0-\frac{2}{5}*0.0=0.970$

Gain($S_{nắng}$, *Nhiệt độ*)= $0.970-\frac{2}{5}*0.0-\frac{2}{5}*1.0-\frac{1}{5}*0.0=0.570$

Gain($S_{nắng}$, *Gió to*)= $0.970-\frac{2}{5}*1.0-\frac{3}{5}*0.918=0.019$

Từ các giá trị của Entropy Gain, ta thấy Độ ẩm là thuộc tính tốt nhất cho đỉnh nằm dưới nhánh Nắng của thuộc tính Quang cảnh.

Tiếp tục quá trình trên cho tất cả các đỉnh và sẽ dừng khi không còn đỉnh nào có thể khai triển được nữa. Cây kết quả sẽ có dạng như phần c) của Hình 1.

2.3. Thuật toán C4.5

Thuật toán này do Quinlan đưa ra năm 1993. Thuật toán C4.5 sinh ra một cây quyết định phân lớp đối với một tập dữ liệu đã được cho bằng cách phân chia đệ qui dữ liệu. Cây quyết định được triển khai theo chiến lược *chiều sâu trước* (Depth-first). Thuật toán này xét tất cả các phép thử có thể phân chia tập dữ liệu đã cho và chọn ra một phép thử cho GainRatio tốt nhất. GainRatio cũng là một độ đo sự hiệu quả của một thuộc tính trong thuật toán triển khai cây quyết định. Nó được tính trên cơ sở của **độ đo** như sau:

$$GainRatio(S,A) \equiv \frac{Gain(S,A)}{SplitInformation(S,A)}$$

$$SplitInformation(S,A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

trong đó S_i là tập con của S với A có giá trị là v_i.

Đối với mỗi thuộc tính rời rạc, chúng ta phải xét một phép thử với tất cả các giá trị khác nhau của nó. Còn đối với mỗi thuộc tính liên tục, ta phải xét các phép thử nhị phân cho mọi giá trị phân biệt của thuộc tính này. Để thu thập Entropy Gain của tất cả các phép thử nhị phân này một cách hữu hiệu thì tập dữ liệu thuộc về đỉnh đang xét phải được phân loại theo các giá trị của thuộc tính liên tục và Entrropy Gains của phép cắt nhị phân dựa trên mỗi giá trị phân biệt của thuộc tính này được tính toán bằng một lần duyệt các dữ liệu đã được phân loại. Quá trình này được thực hiện đối với mọi thuộc tính liên tục.

3. THUẬT TOÁN RÚT GỌN CÁC LUẬT QUYẾT ĐỊNH

3.1. Sinh các luật phân lớp từ cây quyết định

Luật	Nếu	Thì
1	Trời nắng Độ ẩm bình thường	Thi đấu
2	Trời nắng Độ ẩm cao	Không thi đấu
3	Trời nhiều mây	Thi đấu
4	Trời mưa Có gió to	Không thi đấu
5	Trời mưa Không có gió to	Thi đấu

Bảng 4. Tập luật phân lớp chưa rút gọn

Một khi đã xây dựng được cây quyết định, ta có thể chuyển cây quyết định này thành một tập các luật phân lớp tương đương. Ví dụ, từ cây quyết định của tập dữ liệu học của Bảng 1, ta rút ra các luật phân lớp như chỉ ra trong Bảng 4.

3.2. Rút gọn các luật phân lớp

Sau khi thu được một tập các luật phân lớp từ cây quyết định, cần phải tiến hành rút gọn các luật dư thừa nếu có. Dưới đây, chúng tôi đề xuất một phương pháp đơn giản sử dụng các phép thử thống kê để loại bỏ các luật không cần thiết. Phương pháp này bao gồm các bước sau đây:

1) Loại bỏ các tiền đề không cần thiết để đơn giản hoá các luật.

- Xây dựng các bảng ngẫu nhiên (contingency table) cho mỗi luật có chứa nhiều hơn một tiền đề.
- Kiểm chứng sự độc lập của kết quả đối với một tiền đề bằng một trong các phép thử sau:
    - Sử dụng phép thử Khi bình phương nếu các tần xuất mong đợi lớn hơn 10.
    - Sử dụng phép thử Yates nếu các tần xuất mong đợi nằm tròg khoảng [5,10].
    - Sử dụng phép thử Fisher nếu các tần xuất này nhỏ hơn 5.

2) Loại bỏ các luật không cần thiết để rút gọn tập luật

- Một khi đã đơn giản hoá các luật bằng cách loại bỏ các tiền đề dư thừa thì có thể rút gọn toàn bộ tập luật bằng cách bỏ đi các luật không cần thiết.
- Thử thay thế các luật có chung kết quả chung nhất bằng một luật mặc nhiên được tự động áp dụng khi không có luật nào khác thích hợp.

Các bảng ngẫu nhiên

Sau đây là một bảng ngẫu nhiên dùng để biểu diễn một luật dưới dạng bảng:

	C_1	C_2	Các tổng biên
R_1	x_1	x_2	$R_{1T}=x_1+x_2$
R_2	x_3	x_4	$R_{2T}=x_3+x_4$
Các tổng biên	$C_{1T}=x_1+x_3$	$C_{2T}=x_2+x_4$	$T=x_1+x_2+x_3+x_4$

Trong đó:

- R_1 và R_2 biểu diễn các trạng thái Boolean của một tiền đề đối với các kết luận C_1 và C_2 (C_2 là phủ định của C_1).
- x_1 cho đến x_4 biểu diễn tần xuất của từng cặp tiền đề - kết luận.
- R_{1T}, R_2, C_{1T}, C_{2T} là tổng biên của các dòng và các cột tương ứng.
- Các tổng biên và T (tổng tất cả các tần xuất của bảng) được sử dụng để tính các giá trị mong đợi tại các ô dùng trong phép thử độc lập.

Phép thử sự độc lập

Cho một bảng ngẫu nhiên gồm r dòng và c cột:

1) Tính các tổng biên.

2) Tính tổng tần xuất T của bảng.

3) Tính các tần xuất mong đợi cho mỗi ô theo công thức:

$$e_i = \frac{R_{iT} * C_{iT}}{T}$$

Trong đó, R_{iT} và C_{iT} là các tổng dòng và tổng cột tương ứng của ô i trong bảng ngẫu nhiên.

4) Chọn phép thử cần sử dụng để tính χ^2 dựa vào tần xuất mong đợi cao

nhất *m*:

Nếu	Thì sử dụng
$m > 10$	Phép thử Khi bình thường
$5 \le m \le 10$	Phép thử Yates
$m < 5$	Phép thử Fisher

5) Tính χ^2 theo phép thử đã chọn

6) Tính bậc tự do $df = (r-1)*(c-1)$

7) Sử dụng một bảng Khi bình phương với χ^2 và df đã tính được để xác định xem liệu các kết quả có sự độc lập với tiền đề tại mức ý nghĩa đã chọn không.

- Giả sử $\alpha = 0.05$.
- Nếu $\chi^2 > \chi^2_\alpha$ thì loại bỏ giả thiết không về tính độc lập và chấp nhận giả thiết thay thế về tính phụ thuộc. Tức là giữ lại các tiền đề này vì các kết luận phụ thuộc vào chúng.
- Nếu $\chi^2 \le \chi^2_\alpha$ thì chấp nhận giả thuyết không về tính độc lập. Chúng ta sẽ loại bỏ các tiền đề này vì các kết luận độc lập với chúng.

Sau đây là các công thức để tính χ^2 cho các phép thử:

Phép thử Khi bình phương:

$$\chi^2 = \sum_i \frac{(0_i - e_i)^2}{e_i}$$

Phép thử Yates:

$$\chi^2 = \sum_i \frac{(|\, 0_i - e_i\,| - 0.5)^2}{e_i}$$

Chúng ta sẽ thử rút gọn tập luật của ví dụ thi đấu tennis theo phương pháp trên đây. Tập luật chưa rút gọn được liệt kê trong Bảng 4.

Giả sử mức ý nghĩa được lấy là $\alpha = 0.05$. Các dữ liệu học được nhân lên bốn lần để có thể sử dụng theo phép thử Khi bình phương.

3.2.1. Loại bỏ các tiền đề không cần thiết:

a) Xét hai tiền đề trong luật 1: Trời nắng và Độ ẩm bình thường

- Trời nắng

Thực tế:

	Thi đấu	Không thi đấu	Tổng biên
Trời nắng	8	12	20
Không nắng	28	8	36
Tổng biên	36	20	56

Mong đợi:

	Thi đấu	Không thi đấu
Trời nắng	12.9	7.1
Không nắng	23.1	12.9

Do tần xuất mong đợi lớn nhất *m=23.1* nên ta chọn phép thử độc lập Khi bình phương.

Từ đây, theo công thức ta có $\chi^2 = 7.99$.

Bậc tự do của bảng là $df = (r-1)*(c-1) = (2-1)*(2-1) = 1$

Từ bảng Khi bình phương, ta có $\chi^2_\alpha = 3.84$.

Vì $\chi^2 = 7.99 > \chi^2_\alpha = 3.84$, nên chúng ta loại bỏ giả thiết không về tính độc lập và chấp nhận giả thiết thay thế về tính phụ thuộc. Do vậy, theo dữ liệu học thì việc thi đấu tennis phụ thuộc vào trời nắng hay không. Như vậy, chúng ta không thể loại bỏ tiền đề này.

- Độ ẩm bình thường

Thực tế:

	Thi đấu	Không thi đấu	Tổng biên
Độ ẩm bình thường	20	4	24
Độ ẩm không bình thường	16	16	32
Tổng biên	36	20	56

Mong đợi:

	Thi đấu	Không thi đấu
Độ ẩm bình thường	15.4	8.6
Độ ẩm không bình thường	20.6	11.4

Do tần xuất mong đợi lớn nhất *m=20.6* nên ta có thể chọn phép thử độc lập Khi bình phương.

Từ đây, theo công thức ta có $\chi^2 = 6.64$.

Bậc tự do của bảng là $df = (r-1)*(c-1) = (2-1)*(2-1) = 1$

Từ bảng Khi bình phương, ta có $\chi^2_\alpha = 3.84$.

Vì $\chi^2 = 7.99 > \chi^2_\alpha = 3.84$, nên chúng ta loại bỏ giả thiết không về tính độc lập và chấp nhận giả thiết thay thế về tính phụ thuộc. Do vậy, theo dữ liệu học thì việc thi đấu tennis phụ thuộc vào Độ ẩm có bình thường hay không. Như vậy, chúng ta không thể loại bỏ tiền đề này.

Xét tiếp hai tiền đề trong luật 4: Trời mưa và Có gió to.

- Trời mưa

  Thực tế:

	Thi đấu	Không thi đấu	Tổng biên
Trời mưa	12	8	20
Trời không mưa	24	12	36
Tổng biên	36	20	56

Mong đợi:

	Thi đấu	Không thi đấu
Trời mưa	12.9	7.1
Trời không mưa	32.1	12.9

Từ đây, theo công thức ta có $\chi^2 = 0.25$.

Vì $\chi^2 = 0.25 < \chi^2_\alpha = 3.84$, nên chúng ta chấp nhận giả thiết không về tính độc lập. Do vậy, theo dữ liệu học thì việc thi đấu tennis không phụ thuộc vào Trời mưa. Như vậy, chúng ta có thể loại bỏ tiền đề này trong các luật 4 và 5.

- Gió to

  Thực tế:

	Thi đấu	Không thi đấu	Tổng biên
Gió to	8	12	20
Không gió to	28	8	36
Tổng biên	36	20	56

Mong đợi:

	Thi đấu	Không thi đấu
Gió to	12.9	7.1
Không gió to	23.1	12.9

Từ đây, theo công thức ta có $\chi^2 = 7.99$.

Vì $\chi^2 = 7.99 > \chi^2_\alpha = 3.84$, nên chúng ta loại bỏ giả thiết không về tính không độc lập và chấp nhận giả thiết thay thế về tính phụ thuộc. Do vậy, theo dữ liệu học thì việc thi đấu tennis phụ thuộc vào Gió to. Như vậy, chúng ta không thể loại bỏ tiền đề này.

3.2.2. Loại bỏ các luật không cần thiết

Qua việc thử các tiền đề trong tập luật, chúng ta thấy rằng không thể loại bỏ hoàn toàn một luật nào mà chỉ loại bỏ được tiền đề trời mưa trong các luật 4 và 5. Bảng luật thu gọn được cho trong Bảng 5.

Luật	Nếu	Thì
1	Trời nắng Độ ẩm bình thường	Thi đấu
2	Trời nắng Độ ẩm cao	Không thi đấu
3	Trời nhiều mây	Thi đấu
4	Có gió to	Không thi đấu
5	Không có gió to	Thi đấu