

ĐỀ THI GIỮA KỲ
Học phần: NHẬP MÔN TRÍ TUỆ NHÂN TẠO
ĐỀ SỐ 1

Thời gian làm bài: 100 phút

Đề thi gồm: 02 trang

NỘI DUNG ĐỀ THI

Bài 1 (1.5đ): Một định nghĩa phổ biến về Trí tuệ Nhân tạo (AI) là ”một lĩnh vực nghiên cứu, và trong đó các máy tính được sử dụng để thực hiện các công việc mà hiện tại con người đang làm tốt hơn” (Ertel, 2017). Dựa trên kiến thức của bạn và bối cảnh phát triển của AI hiện đại (ví dụ: xe tự lái, AlphaGo, ChatGPT):

1. Phân tích những điểm mạnh và điểm yếu của định nghĩa trên.
2. Theo bạn, định nghĩa này còn phù hợp với bối cảnh AI ngày nay không? Tại sao?

Bài 2 (1đ): Sử dụng phương pháp **Resolution (Hợp giải)**, hãy chứng minh rằng tập mệnh đề sau là **không thỏa mãn được (unsatisfiable)**. Trình bày rõ các bước chuyển đổi sang dạng CNF (nếu cần) và quá trình hợp giải.

$$KB = \{A \vee B, \neg A \vee C, \neg B \vee C, \neg C\}$$

Bài 3 (1đ): Chuyển đổi câu logic vị từ bậc nhất sau đây sang **Dạng chuẩn hội (Conjunctive Normal Form - CNF)**. Trình bày rõ các bước thực hiện.

$$\forall x(\text{Student}(x) \Rightarrow \exists y(\text{Takes}(x, y) \wedge \text{IsCourse}(y)))$$

(Mọi sinh viên đều tham gia ít nhất một khóa học).

Bài 4 (1đ): Giải thích khái niệm ”**bán quyết định được (semi-decidable)**” của logic vị từ bậc nhất (PL1). Điều này khác với tính ”**quyết định được (decidable)**” của logic mệnh đề như thế nào? Nếu hàm ý thực tiễn của tính chất này đối với các hệ thống chứng minh định lý tự động.

Bài 5 (1.5đ): Cho chương trình PROLOG sau đây để tính giao thừa của một số không âm:

```
factorial(0, 1).
factorial(N, F) :-  
    N > 0,  
    N1 is N - 1,  
    factorial(N1, F1),  
    F is N * F1.
```

- Chương trình sẽ trả về kết quả gì cho truy vấn 'factorial(3, Result)'?
- Mô tả ngắn gọn từng bước thực thi (**truy vết**) của PROLOG để tìm ra kết quả đó.

Bài 6 (2đ): Cho cây trò chơi (**game tree**) sau đây. Nút **MAX (A)** cần đi nước tiếp theo. Các giá trị ở các nút lá là giá trị lượng giá (**evaluation value**) cho MAX. **Cấu trúc Cây Trò chơi (Game Tree):**

A (MAX) là gốc.

Con của A là B (MIN), C (MIN), D (MIN).

Con của B là E(3), F(5).

Con của C là G(2), H(9).

Con của D là I(1), J(2).

- Sử dụng thuật toán **Minimax**, giá trị tại nút gốc A là bao nhiêu? Nước đi tốt nhất cho MAX là nước nào (tới B, C, hay D)?
- Áp dụng thuật toán **cắt tỉa Alpha-Beta (Alpha-Beta Pruning)** trên cây này theo thứ tự từ trái sang phải. Những nhánh nào sẽ bị cắt tỉa? Giải thích tại sao.

Bài 7 (2đ): Xét Mạng Bayesian đơn giản sau: **Cấu trúc Mạng Bayesian:**

Cloudy (C) → Sprinkler (S)

Cloudy (C) → Rain (R)

Sprinkler (S) → WetGrass (W)

Rain (R) → WetGrass (W)

(Tức là C là cha của S và R; S và R là cha của W)

- Viết công thức tính **xác suất đồng thời** $P(C, S, R, W)$ theo **quy tắc chuỗi (chain rule)** cho Mạng Bayesian này.
- Dựa vào cấu trúc mạng, hãy xác định các **cặp biến độc lập có điều kiện**. Ví dụ: 'Sprinkler' và 'Rain' có độc lập với nhau không? Có độc lập có điều kiện với 'Cloudy' không? Giải thích.

- HẾT -

Cán bộ coi thi không giải thích gì thêm. Sinh viên không được sử dụng tài liệu.

Phần 2: Đáp án và Thang điểm

Câu 1 (1.5 điểm)

a) Phân tích điểm mạnh, điểm yếu (1.0 điểm)

- Điểm mạnh (0.5đ):

– **Dễ hiểu, trực quan:** Định nghĩa này dễ nắm bắt, tập trung vào so sánh năng lực giữa máy và người.

- **Hướng đến ứng dụng:** Nó nhấn mạnh vào việc tạo ra các hệ thống hữu ích, giải quyết các vấn đề thực tế mà con người đang làm.
- **Đo lường được:** Sự "tốt hơn" có thể được đo lường qua hiệu suất, tốc độ, độ chính xác trong nhiều tác vụ cụ thể.

- **Điểm yếu (0.5đ):**

- **Mục tiêu di động:** Khi máy tính làm tốt một việc gì đó (ví dụ: chơi cờ), người ta lại có xu hướng không coi đó là "trí tuệ" nữa. "Những việc con người làm tốt hơn" là một ranh giới luôn thay đổi.
- **Quá hẹp:** AI không chỉ là bắt chước con người. Nhiều kỹ thuật AI thành công (như tìm kiếm, tối ưu hóa) giải quyết vấn đề theo cách hoàn toàn khác con người, thậm chí siêu phàm (ví dụ: AlphaGo).
- **Không bao quát "trí tuệ":** Nó tập trung vào "công việc" (tasks) thay vì các khía cạnh khác của trí tuệ như ý thức, cảm xúc, hay khả năng học hỏi tổng quát.

b) Tính phù hợp (0.5 điểm)

- Định nghĩa này **vừa còn phù hợp, vừa có phần lỗi thời.**
- **Phù hợp:** Ở khía cạnh AI hẹp (Narrow AI), nó vẫn đúng. Các hệ thống như ChatGPT vượt qua con người trong việc tổng hợp và tạo văn bản nhanh chóng. Xe tự lái hướng tới việc lái xe an toàn hơn con người.
- **Lỗi thời:** AI hiện đại không chỉ nhắm đến việc "làm tốt hơn" mà còn "làm khác đi" và "làm những điều con người không thể". Các mô hình lớn khám phá những không gian giải pháp khổng lồ mà con người không thể tư duy hết. Do đó, một định nghĩa tập trung vào "hành động hợp lý" (acting rationally) - tức là hành động để đạt được mục tiêu một cách tối ưu - có lẽ bao quát và bền vững hơn.

Câu 2 (1.0 điểm)

- **Chuyển sang CNF (0.25đ):** Tập KB đã ở dạng CNF. Mỗi mệnh đề là một disjunction của các literal.
- **Quá trình hợp giải (0.75đ):**
 - (1) $A \vee B$
 - (2) $\neg A \vee C$
 - (3) $\neg B \vee C$
 - (4) $\neg C$
 1. Hợp giải (1) và (2): $B \vee C$ (0.25đ)
 2. Hợp giải (5) và (3): $C \vee C \equiv C$ (0.25đ)
 3. Hợp giải (6) và (4): $C \wedge \neg C \Rightarrow \perp$ (mệnh đề rỗng) (0.25đ)
- **Kết luận:** Vì ta suy ra được mệnh đề rỗng, tập KB ban đầu là không thỏa mãn được.

Câu 3 (1.0 điểm)

Câu gốc:

$$\forall x(\text{Student}(x) \Rightarrow \exists y(\text{Takes}(x, y) \wedge \text{IsCourse}(y)))$$

1. **Loại bỏ \Rightarrow (0.25đ):**

$$\forall x(\neg\text{Student}(x) \vee \exists y(\text{Takes}(x, y) \wedge \text{IsCourse}(y)))$$

2. **Di chuyển \exists ra ngoài (0.25đ):** Không cần di chuyển phủ định.

3. **Skolem hóa (0.25đ):** Biến y phụ thuộc vào x . Ta thay $\exists y$ bằng một hàm Skolem, ví dụ $C(x)$.

$$\forall x(\neg\text{Student}(x) \vee (\text{Takes}(x, C(x)) \wedge \text{IsCourse}(C(x))))$$

4. **Phân phối \vee qua \wedge (0.25đ):**

$$\forall x[(\neg\text{Student}(x) \vee \text{Takes}(x, C(x))) \wedge (\neg\text{Student}(x) \vee \text{IsCourse}(C(x)))]$$

5. **Loại bỏ \forall :**

$$(\neg\text{Student}(x) \vee \text{Takes}(x, C(x)))$$

$$(\neg\text{Student}(x) \vee \text{IsCourse}(C(x)))$$

Đây là dạng CNF.

Câu 4 (1.0 điểm)

- **Giải thích các khái niệm (0.5đ):**

- **Quyết định được (Decidable):** Một hệ thống logic được gọi là "quyết định được" nếu tồn tại một thuật toán luôn kết thúc và trả lời đúng cho câu hỏi: "Công thức F có phải là một định lý (hằng đúng) hay không?". Logic mệnh đề là "quyết định được" (ví dụ, bằng cách dùng bảng chân trị).
 - **Bán quyết định được (Semi-decidable):** Một hệ thống logic được gọi là "bán quyết định được" nếu tồn tại một thuật toán mà: (1) Nếu công thức F là một định lý, thuật toán sẽ dừng và trả lời "CÓ". (2) Nếu F không phải là một định lý, thuật toán có thể sẽ chạy vô tận mà không đưa ra câu trả lời. Logic vị từ bậc nhất (PL1) là "bán quyết định được".

- **Hàm ý thực tiễn (0.5đ):**

- Đối với một hệ thống chứng minh định lý tự động cho PL1, nếu ta đưa vào một công thức là hằng đúng, hệ thống được đảm bảo sẽ tìm ra bằng chứng và dùng lại (ví dụ, bằng phương pháp Resolution sẽ sinh ra mệnh đề rỗng).
 - Tuy nhiên, nếu ta đưa vào một công thức **không phải** là hằng đúng, hệ thống có thể sẽ chạy mãi mãi, liên tục sinh ra các mệnh đề mới mà không bao giờ tìm thấy mâu thuẫn và cũng không thể kết luận rằng không có bằng chứng. Điều này có nghĩa là ta không thể dựa vào hệ thống để chứng minh một công thức là **không** đúng; ta chỉ có thể dùng nó để chứng minh một công thức là đúng.

Câu 5 (1.5 điểm)

a) Kết quả (0.5đ)

Result = 6.

b) Mô tả các bước (1.0đ)

1. **Query:** factorial(3, Result).
2. Khớp với factorial(N, F). N=3 > 0 là đúng.
3. N1 is 3 - 1 = 2.
4. **Gọi đệ quy:** factorial(2, F1).
 - Khớp với factorial(N, F). N=2 > 0 là đúng.
 - N1 is 2 - 1 = 1.
 - **Gọi đệ quy:** factorial(1, F1_inner).
 - Khớp với factorial(N, F). N=1 > 0 là đúng.
 - N1 is 1 - 0 = 0.
 - **Gọi đệ quy:** factorial(0, F1_innermost).
 - * Khớp với điều kiện dừng factorial(0, 1). F1_innermost được gán là 1. Lời gọi này thành công.
 - Quay lại factorial(1,...): F1_inner giờ là 1. F is 1 * 1 = 1. Lời gọi này thành công.
 - Quay lại factorial(2,...): F1 giờ là 1. F is 2 * 1 = 2. Lời gọi này thành công.
 - 5. Quay lại factorial(3,...): F1 giờ là 2. F is 3 * 2 = 6.
 - 6. Result được gán là 6.

Câu 6 (2.0 điểm)

a) Thuật toán Minimax (1.0 điểm)

- Giá trị tại nút B = min(3, 5) = 3.
- Giá trị tại nút C = min(2, 9) = 2.
- Giá trị tại nút D = min(1, 2) = 1.
- Giá trị tại nút A = max(3, 2, 1) = **3**. (0.5đ)
- Nước đi tốt nhất cho MAX là đi tới nút **B**. (0.5đ)

b) Cắt tỉa Alpha-Beta (1.0 điểm)

- **Alpha (α):** giá trị tốt nhất hiện tại cho MAX (khởi tạo $-\infty$).
- **Beta (β):** giá trị tốt nhất hiện tại cho MIN (khởi tạo $+\infty$).

1. Xét nút A, nhánh B:

- Xét nút B (MIN), alpha = $-\infty$, beta = $+\infty$.
- Lượng giá E, thấy giá trị 3. MIN (B) sẽ không chọn nước đi nào > 3 . Cập nhật beta của B = 3.
- Lượng giá F, thấy giá trị 5. Vì $5 > 3$ (beta của B), MIN sẽ không chọn F. Giá trị của B là 3.
- MAX (A) giờ biết mình có thể đạt được ít nhất 3. Cập nhật alpha của A = 3.

2. Xét nút A, nhánh C:

- Xét nút C (MIN), alpha = 3 (thừa hưởng từ A), beta = $+\infty$.
- Lượng giá G, thấy giá trị 2.
- Tại đây, MIN (C) đã tìm thấy một nước đi có giá trị là 2. Vì $2 < 3$ (tức là $\text{giá}_\text{tri}_\text{C} \leq 2 < \text{alpha}_\text{A}$), MAX (A) sẽ không bao giờ chọn đi qua nhánh C, vì MAX đã có một lựa chọn tốt hơn (nhánh B, giá trị 3).
- **Nhánh H bị cắt tỉa (0.5đ).** Không cần lượng giá nút H.

3. Xét nút A, nhánh D:

- Xét nút D (MIN), alpha = 3, beta = $+\infty$.
- Lượng giá I, thấy giá trị 1.
- Tương tự như nhánh C, MIN (D) đã tìm thấy nước đi có giá trị 1. Vì $1 < 3$ ($\text{giá}_\text{tri}_\text{D} \leq 1 < \text{alpha}_\text{A}$), MAX sẽ không bao giờ chọn đi qua nhánh D.
- **Nhánh J bị cắt tỉa (0.5đ).** Không cần lượng giá nút J.

Câu 7 (2.0 điểm)

a) Công thức xác suất đồng thời (1.0 điểm)

Dựa trên quy tắc chuỗi và các quan hệ độc lập có điều kiện trong mạng, ta có:

$$P(C, S, R, W) = P(C) \cdot P(S|C) \cdot P(R|C) \cdot P(W|S, R)$$

(Giải thích: W độc lập với C khi biết S và R. S và R độc lập với nhau khi biết C)

b) Độc lập có điều kiện (1.0 điểm)

- **Sprinkler và Rain có độc lập không? Không.** Nếu ta không biết gì về trời mây (Cloudy), việc biết trời mưa (Rain) làm giảm khả năng vòi phun nước đang bật (Sprinkler), vì mưa là một lời giải thích khác cho việc cỏ ướt. Đây là hiện tượng "explaining away".
- **Sprinkler và Rain có độc lập có điều kiện với Cloudy không? Có.** Theo cấu trúc mạng, Sprinkler và Rain không có cung nối trực tiếp và chúng có chung một cha là Cloudy. Khi biết trạng thái của Cloudy (ví dụ, chắc chắn trời đang có mây), việc biết trời có mưa hay không không cung cấp thêm thông tin gì về việc vòi phun có đang bật hay không.
- **Ký hiệu xác suất:** $P(S|R, C) = P(S|C)$. (Biết C, thì S và R độc lập). (0.5đ cho mỗi ý đúng)