

ĐỀ THI GIỮA KỲ – LỜI GIẢI

Nhập môn Trí tuệ nhân tạo – Lớp MAT3508 3 và MAT3508 4

Học kỳ I – Năm học 2026–2027 – Ngày thi: 28/10/2026

Thời gian: 90 phút (không kể thời gian phát đề)

Câu 1 (1 điểm)

- (a) (0,5 điểm) Một tác tử có thể được mô tả một cách tổng quát như thế nào?

How can an agent be generally described?

Lời giải

Một cách tổng quát, **tác tử** là một hệ thống xử lý thông tin nhận dữ liệu đầu vào và tạo kết quả đầu ra tương ứng

- (b) (0,5 điểm) Vì sao một tác tử tất định có bộ nhớ không phải là một hàm (theo nghĩa toán học) từ tập các đầu vào đến tập các đầu ra?

Why is a deterministic agent with memory not a function (in the mathematical sense) from the set of inputs to the set of outputs?

Lời giải

Một hàm (theo nghĩa toán học) gán cho mỗi đầu vào **đúng một** đầu ra.

Đầu ra của tác tử có bộ nhớ không chỉ phụ thuộc vào đầu vào hiện tại mà còn phụ thuộc vào nội dung bộ nhớ, tức là vào các đầu vào trước đó. Vì vậy, cùng một đầu vào ở hai thời điểm khác nhau có thể cho hai đầu ra khác nhau, nên quan hệ giữa đầu vào và đầu ra không phải là một hàm.

Câu 2 (1,5 điểm)

- (a) (0,5 điểm) Dạng chuẩn hội (CNF) là gì?

What is conjunctive normal form (CNF)?

Lời giải

Một công thức ở **dạng chuẩn hội (CNF)** khi và chỉ khi nó là hội của các mệnh đề (clause):

$$K_1 \wedge K_2 \wedge \cdots \wedge K_m, \quad (1)$$

trong đó mỗi mệnh đề K_i là tuyển của các literal

$$K_i = L_{i1} \vee L_{i2} \vee \cdots \vee L_{in_i}. \quad (2)$$

Mỗi literal là một biến mệnh đề (positive literal) (ví dụ $P, Q, R, \dots$) hoặc phủ định của một biến mệnh đề (negative literal) (ví dụ $\neg P, \neg Q, \neg R, \dots$).

(b) (1 điểm) Biến đổi công thức sau về dạng chuẩn hội:

Transform the following formula into conjunctive normal form:

$$(A \wedge B) \Leftrightarrow (A \vee B)$$

Lời giải

Biến đổi lần lượt:

1. Khử $\Leftrightarrow$. Ta có: $((A \wedge B) \Rightarrow (A \vee B)) \wedge ((A \vee B) \Rightarrow (A \wedge B))$.
2. Khử $\Rightarrow$. Ta có: $(\neg(A \wedge B) \vee A \vee B) \wedge (\neg(A \vee B) \vee (A \wedge B))$.
3. Đưa phủ định vào trong (De Morgan): $(\neg A \vee \neg B \vee A \vee B) \wedge ((\neg A \wedge \neg B) \vee (A \wedge B))$.
4. Phân phối $\vee$ qua $\wedge$ ở vế sau: $(\neg A \vee \neg B \vee A \vee B) \wedge (\neg A \vee A) \wedge (\neg A \vee B) \wedge (A \vee \neg B) \wedge (\neg B \vee B)$.
5. Bỏ các mệnh đề luôn đúng (chứa một biến cùng phủ định của nó).

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

Dạng chuẩn hội không là duy nhất. Câu trả lời đúng về logic khi khử đều được full điểm.

Câu 3 (1,5 điểm)

(a) (0,5 điểm) Phép hợp nhất tổng quát nhất (most general unifier – MGU) là gì?

What is a most general unifier (MGU)?

Lời giải

Một phép thế σ cho các biến làm cho hai literal trở nên giống hệt nhau gọi là một **phép hợp nhất (unifier)**; khi đó hai literal được gọi là hợp nhất được. Một phép hợp nhất được gọi là **phép hợp nhất tổng quát nhất (MGU)** nếu mọi phép hợp nhất khác đều nhận được từ nó bằng cách thế thêm các biến.

(b) (1 điểm) Hợp nhất (nếu có thể) hai biểu thức sau, trong đó x, y là biến. Trình bày từng bước thuật toán hợp nhất, cho biết MGU và các biểu thức thu được sau khi hợp nhất.

Unify (if possible) the following expressions, where x and y are variables. Show each step of the unification algorithm, and give the MGU and the resulting expressions.

$$p(x, f(x)), \quad p(y, y)$$

Lời giải

Áp dụng thuật toán hợp nhất cho

$$W = \{p(x, f(x)), p(y, y)\}.$$

$k = 0, W_0 = W = \{p(x, f(x)), p(y, y)\}, \sigma_0 = \epsilon$ (trong đó ϵ là phép rỗng).

Tập sai khác (disagreement set) là $D_0 = \{x, y\}$. Do đó sử dụng phép thế $\sigma_1 = \{x/y\}$ (thế x bằng y). Thay vào W ta có

$$W_1 = \{p(y, f(y)), p(y, y)\}. \quad (3)$$

Tiếp tục, $D_1 = \{f(y), y\}$. Vì y xuất hiện trong $f(y)$ nên thuật toán dừng lại và kết luận không thể hợp nhất được.

Câu 4 (2 điểm)

Một nông dân cần đưa một con sói, một con dê và một cây bắp cải từ bờ trái sang bờ phải của một con sông. Thuyền chỉ chở được nông dân cùng nhiều nhất một trong ba thứ trên. Khi không có nông dân ở cùng bờ, sói sẽ ăn dê và dê sẽ ăn bắp cải. Mỗi trạng thái được biểu diễn bằng hạng thức `state(Farmer, Wolf, Goat, Cabbage)`, các thành phần nhận giá trị `left` (bờ trái) hoặc `right` (bờ phải). Cho chương trình GNU Prolog sau:

A farmer needs to take a wolf, a goat and a cabbage from the left bank to the right bank of a river. The boat can carry only the farmer and at most one of the three. When the farmer is not on the same bank, the wolf eats the goat and the goat eats the cabbage. Each state is represented by the term `state(Farmer, Wolf, Goat, Cabbage)`, whose arguments are `left` or `right` (the bank). Consider the following GNU Prolog program:

```
start :- action(state(left,left,left,left),
               state(right,right,right,right)).

action(Start,Goal) :-
    plan(Start,Goal,[Start],Path),
    nl, write('Solution:'), nl,
    write(Path), nl.

plan(Start,Goal,Visited,Path) :-
    go(Start,Next),
    safe(Next),
    \+ member(Next,Visited),      % not(member(...))
    plan(Next,Goal,[Next|Visited],Path).
plan(Goal,Goal,Path,Path).

go(state(X,X,Z,K),state(Y,Y,Z,K)) :- across(X,Y).    % farmer, wolf
go(state(X,W,X,K),state(Y,W,Y,K)) :- across(X,Y).    % farmer, goat
go(state(X,W,Z,X),state(Y,W,Z,Y)) :- across(X,Y).    % farmer, cabbage
go(state(X,W,Z,K),state(Y,W,Z,K)) :- across(X,Y).    % farmer

across(left,right).
across(right,left).

safe(state(_,W,Z,K)) :- across(W,Z), across(Z,K).
safe(state(B,B,B,_)).
safe(state(B,_,B,B)).
```

(a) (0,5 điểm) Cần nhập truy vấn nào để chạy chương trình?

What query should be entered to run the program?

Lời giải

Nhập truy vấn `start.` (phải có dấu chấm ở cuối).

(b) (0,5 điểm) Chạy chương trình trên, ta được kết quả:

Running the program above gives the following output:

```
Solution:
[state(right,right,right,right),state(left,right,left,right),
 state(right,right,left,right),state(left,right,left,left),
 state(right,right,right,left),state(left,left,right,left),
 state(right,left,right,left),state(left,left,left,left)]
```

Nhận xét về thứ tự các trạng thái trong danh sách. Từ đó, trình bày lời giải thành dãy các lần qua sông **theo đúng thứ tự thực hiện**, mỗi lần qua sông viết thành một câu, chẳng hạn:

Farmer and wolf cross from left to right.

Comment on the order of the states in the list. Based on that, write the solution as a sequence of river crossings in the order they are carried out, one sentence per crossing, for example:

Farmer and wolf cross from left to right.

Lời giải

Danh sách là một lời giải gồm 8 trạng thái, mỗi trạng thái cho biết nông dân, sói, dê và bắp cải đang ở bờ nào. Hai trạng thái liên nhau cách nhau đúng một lần qua sông, nên lời giải có 7 lần qua sông.

Danh sách **theo thứ tự ngược**: phần tử đầu `state(right,right,right,right)` là đích, phần tử cuối `state(left,left,left,left)` là xuất phát. Lý do là plan thêm trạng thái mới vào đầu danh sách (`[Next|Visited]`). Đọc từ cuối lên, lời giải là:

1. Farmer and goat cross from left to right
2. Farmer crosses from right to left
3. Farmer and wolf cross from left to right
4. Farmer and goat cross from right to left
5. Farmer and cabbage cross from left to right
6. Farmer crosses from right to left
7. Farmer and goat cross from left to right

- (c) (1 điểm) Để chương trình tự in lời giải thành các câu như ở ý (b) thay vì danh sách trạng thái, ta thay dòng `write(Path), nl.` trong vị từ `action` bằng `write_path(Path)`. và thêm vị từ `write_path` vào cuối chương trình như sau:

To make the program print the solution as sentences like those in (b) instead of a list of states, we replace the line `write(Path), nl.` in the predicate `action` by `write_path(Path)`. and add the predicate `write_path` at the end of the program as follows:

```
action(Start,Goal) :-
    plan(Start,Goal,[Start],Path),
    nl, write('Solution:'), nl,
    write_path(Path).           % replaces write(Path), nl

% ... plan, go, across, safe: unchanged ...

write_path([_]).
write_path([S2,S1|Rest]) :-
    write_path([S1|Rest]),
    write_move(S1,S2).

% ===== BEGIN: Complete the clauses of write_move/2 below =====
% (1) Farmer crosses from <X> to <Y>
% (2) Farmer and wolf cross from <X> to <Y>
% (3) Farmer and goat cross from <X> to <Y>
% (4) Farmer and cabbage cross from <X> to <Y>
% ===== END: write_move/2 =====
```

Hãy hoàn thiện chương trình: tại mỗi vị trí có chú thích (1)–(4), viết một mệnh đề của vị từ `write_move(State1,State2)`, trong đó `State1`, `State2` là trạng thái ngay trước và ngay sau một lần qua sông. Mỗi mệnh đề in đúng một dòng theo mẫu ở chú thích (<X>: bờ xuất phát, <Y>: bờ đến) và chỉ dùng `write/1`, `nl/0`. Bài làm chỉ cần ghi các mệnh đề này.

Complete the program: at each position marked by a comment (1)–(4), write one clause of the predicate `write_move(State1,State2)`, where `State1` and `State2` are the states just before and just after a river crossing. Each clause prints exactly one line following the pattern in its comment (<X>: bank of departure, <Y>: bank of arrival) and uses only `write/1` and `nl/0`. In your answer, write only these clauses.

Lời giải

```
write_move(state(X,W,Z,K), state(Y,W,Z,K)) :-
    write('Farmer crosses from '),
    write(X), write(' to '), write(Y), nl.
write_move(state(X,X,Z,K), state(Y,Y,Z,K)) :-
    write('Farmer and wolf cross from '),
    write(X), write(' to '), write(Y), nl.
write_move(state(X,W,X,K), state(Y,W,Y,K)) :-
    write('Farmer and goat cross from '),
    write(X), write(' to '), write(Y), nl.
write_move(state(X,W,Z,X), state(Y,W,Z,Y)) :-
    write('Farmer and cabbage cross from '),
    write(X), write(' to '), write(Y), nl.
```

Mỗi lần qua sông chỉ khớp đúng một trong bốn mẫu đầu mệnh đề, nên không cần dùng ! và thứ tự các mệnh đề không quan trọng.

Câu 5 (2 điểm)

Xét bài toán 8-puzzle với trạng thái đầu và trạng thái đích dưới đây (ô màu xám là ô trống; mỗi nước đi di chuyển ô trống sang một ô kề: lên (U), xuống (D), trái (L) hoặc phải (R)).

Consider the 8-puzzle with the start and goal states below (the grey square is the blank; a move shifts the blank to an adjacent square: up (U), down (D), left (L) or right (R)).

Trạng thái đầu

1		3
4	2	6
7	5	8

Trạng thái đích

1	2	3
4	5	6
7	8	

Dùng tìm kiếm theo chiều rộng (breadth-first search), hãy vẽ cây tìm kiếm từ trạng thái đầu cho đến khi tìm thấy trạng thái đích và cho biết đường đi tìm được (dãy nước đi của ô trống). Sinh các nút con theo thứ tự U, D, L, R, tránh chu trình độ dài 2 (không sinh nước đi quay lại trạng thái cha), và đánh số các nút theo thứ tự được sinh ra. Kiểm tra đích ngay khi sinh ra mỗi nút.

Using breadth-first search, draw the search tree from the start state until the goal state is found, and give the path found (the sequence of moves of the blank). Generate successors in the order U, D, L, R, avoiding cycles of length 2, and number the nodes in the order in which they are generated. Apply the goal test to each node as soon as it is generated.

Lời giải

Cách làm. Tìm kiếm theo chiều rộng giữ các nút đã sinh nhưng chưa được mở rộng trong một hàng đợi (vào trước, ra trước), nên các nút được mở rộng đúng theo thứ tự được sinh ra và cây được mở rộng lần lượt từng mức. Khi mở rộng một nút, thử các nước đi của ô trống theo thứ tự U, D, L, R; bỏ qua nước đi ra ngoài bảng và nước đi quay lại trạng thái cha. Mỗi nút con được đánh số theo thứ tự sinh ra và được kiểm tra ngay khi sinh; nếu là trạng thái đích thì thuật toán dừng.

Các bước.

Bước 1. Mở rộng nút 1 (trạng thái đầu, ô trống ở giữa hàng trên). Không đi U được vì ô trống ở hàng trên cùng. Sinh nút 2 (D: ô trống đổi chỗ với số 2), nút 3 (L: đổi chỗ với số 1) và nút 4 (R: đổi chỗ với số 3); không nút nào là đích. Hàng đợi: 2, 3, 4.

Bước 2. Mở rộng nút 2 (ô trống ở giữa bảng). Nút 2 có được từ nước D nên không đi U, vì sẽ quay lại nút 1. Sinh nút 5 (D, đổi chỗ với số 5), nút 6 (L, đổi chỗ với số 4) và nút 7 (R, đổi chỗ với số 6); không nút nào là đích. Hàng đợi: 3, 4, 5, 6, 7.

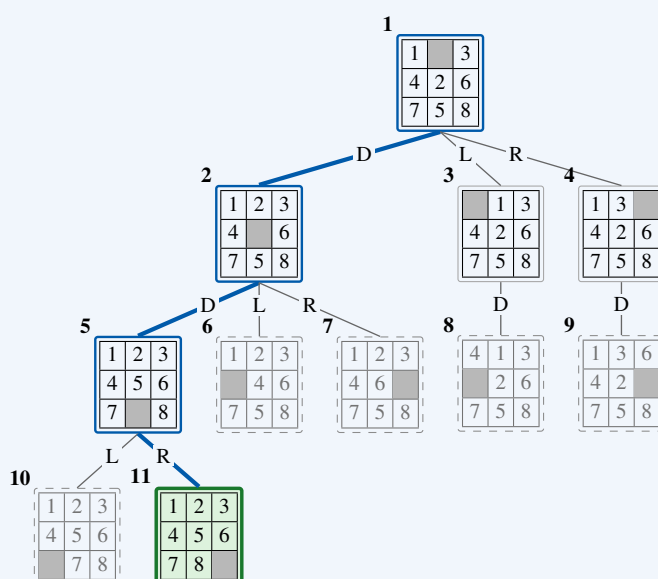
Bước 3. Mở rộng nút 3 (ô trống ở góc trên bên trái). U và L ra ngoài bảng, R quay lại nút 1, nên chỉ sinh nút 8 (D, đổi chỗ với số 4); nút 8 không phải đích. Hàng đợi: 4, 5, 6, 7, 8.

Bước 4. Mở rộng nút 4 (ô trống ở góc trên bên phải). U và R ra ngoài bảng, L quay lại nút 1, nên chỉ sinh nút 9 (D, đổi chỗ với số 6); nút 9 không phải đích. Hàng đợi: 5, 6, 7, 8, 9.

Bước 5. Mở rộng nút 5 (ô trống ở giữa hàng dưới). U quay lại nút 2, D ra ngoài bảng. Sinh nút 10 (L, đổi chỗ với số 7), không phải đích; rồi sinh nút 11 (R, đổi chỗ với số 8), đây là trạng thái đích nên thuật toán dừng. Các nút 6–10 đã được sinh nhưng chưa được mở rộng.

Lần ngược từ nút 11 theo nút cha (cha của 11 là 5, cha của 5 là 2, cha của 2 là 1), ta được đường đi $1 \rightarrow 2 \rightarrow 5 \rightarrow 11$ với các nước D, D, R.

Cây tìm kiếm và bảng dưới đây trình bày lại các bước trên.

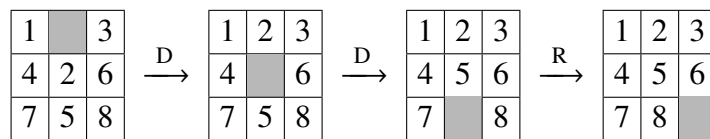


Cây tìm kiếm theo chiều rộng. Các nút được đánh số theo thứ tự sinh ra và được kiểm tra ngay khi sinh; chữ trên cạnh là nước đi của ô trống. Nét đứt: nút đã sinh nhưng chưa được mở rộng khi tìm thấy đích. Xanh dương: đường đi tìm được; xanh lá: nút đích.

Bước	Nút mở rộng	Nút con sinh ra (nước đi)	Kiểm tra đích khi sinh
1	1	2 (D), 3 (L), 4 (R)	không có đích
2	2	5 (D), 6 (L), 7 (R)	không có đích
3	3	8 (D)	không có đích
4	4	9 (D)	không có đích
5	5	10 (L), 11 (R)	10: không; 11: đúng $\Rightarrow$ dừng

Các bước tìm kiếm theo chiều rộng: các nút được mở rộng theo thứ tự được sinh ra. Ví dụ: nút 2 có được từ nước D nên không sinh nước U từ nút 2; nút 3 chỉ sinh được nước D vì ô trống ở góc trên bên trái (không đi U, L được) và nước R quay lại nút 1.

Đường đi tìm được $1 \rightarrow 2 \rightarrow 5 \rightarrow 11$: ô trống đi D, D, R (ô 2 đi lên, ô 5 đi lên, ô 8 sang trái).



Bài làm chỉ cần có cây tìm kiếm hoặc bảng với các trạng thái và số thứ tự nút đúng, chọn đúng nút để mở rộng ở mỗi bước (nút được sinh ra sớm nhất trong số các nút chưa được mở rộng) và kết quả đúng (đường đi $1 \rightarrow 2 \rightarrow 5 \rightarrow 11$, các nước D, D, R) là được điểm tối đa.

Câu 6 (2 điểm)

Cần dự đoán thời tiết buổi chiều dựa vào một vài quan sát thời tiết đơn giản vào buổi sáng cùng ngày. Mô hình xác suất đầy đủ, tức phân phối xác suất đồng thời của ba biến *Sky*, *Bar* và *Prec*, được cho trong bảng dưới đây.

You are to predict the afternoon weather from a few simple weather observations made in the morning of the same day. The complete probabilistic model, i.e. the joint probability distribution of the three variables Sky, Bar and Prec, is given in the table below.

<i>Sky</i>	<i>Bar</i>	<i>Prec</i>	$P(\text{Sky}, \text{Bar}, \text{Prec})$
Clear	Rising	Dry	0,40
Clear	Rising	Raining	0,07
Clear	Falling	Dry	0,08
Clear	Falling	Raining	0,10
Cloudy	Rising	Dry	0,09
Cloudy	Rising	Raining	0,11
Cloudy	Falling	Dry	0,03
Cloudy	Falling	Raining	0,12

Sky: Bầu trời buổi sáng quang đãng (clear) hay nhiều mây (cloudy).

The sky is clear or cloudy in the morning.

Bar: Khí áp kế buổi sáng báo áp suất tăng (rising) hay giảm (falling).

The barometer is rising or falling in the morning.

Prec: Buổi chiều trời mưa (raining) hay khô ráo (dry).

It is raining or dry in the afternoon.

(a) (0,5 điểm) Phát biểu công thức xác suất có điều kiện $P(A | B)$, với $P(B) > 0$.

State the formula for the conditional probability $P(A | B)$, where $P(B) > 0$.

Lời giải

Với $P(B) > 0$:

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

Tương đương: $P(A \wedge B) = P(A | B) P(B)$ (quy tắc nhân).

- (b) (0,5 điểm) Tính $P(Prec = \text{dry} | Sky = \text{clear}, Bar = \text{rising})$.

Compute $P(Prec = \text{dry} | Sky = \text{clear}, Bar = \text{rising})$.

Lời giải

Để gọn, viết $P(\text{clear}, \text{rising}, \text{dry})$ thay cho $P(Sky = \text{clear}, Bar = \text{rising}, Prec = \text{dry})$; các xác suất khác viết tương tự. Cộng hai dòng đầu của bảng (hai dòng có $Sky = \text{clear}, Bar = \text{rising}$):

$$P(\text{clear}, \text{rising}) = 0,40 + 0,07 = 0,47.$$

Theo định nghĩa xác suất có điều kiện:

$$P(\text{dry} | \text{clear}, \text{rising}) = \frac{P(\text{clear}, \text{rising}, \text{dry})}{P(\text{clear}, \text{rising})} = \frac{0,40}{0,47} = \frac{40}{47} \approx 0,851.$$

- (c) (1 điểm) Mỗi buổi sáng, một thiết bị cảm biến dự báo buổi chiều có mưa hay không. Gọi S là sự kiện thiết bị báo mưa. Khi buổi chiều trời mưa, thiết bị báo mưa với xác suất $P(S | Prec = \text{raining}) = 0,9$; khi buổi chiều khô ráo, thiết bị vẫn báo mưa với xác suất $P(S | Prec = \text{dry}) = 0,2$. Dùng công thức Bayes, tính xác suất buổi chiều trời mưa khi thiết bị báo mưa, $P(Prec = \text{raining} | S)$.

Every morning, a sensor predicts whether it will rain in the afternoon. Let S be the event that the sensor predicts rain. When it rains in the afternoon, the sensor predicts rain with probability $P(S | Prec = \text{raining}) = 0.9$; when the afternoon is dry, it still predicts rain with probability $P(S | Prec = \text{dry}) = 0.2$. Using Bayes' formula, compute the probability that it rains in the afternoon given that the sensor predicts rain, $P(Prec = \text{raining} | S)$.

Lời giải

Ký hiệu R là sự kiện $Prec = \text{raining}$; khi đó $\neg R$ là sự kiện $Prec = \text{dry}$. Cộng bốn dòng có $Prec = \text{raining}$ trong bảng:

$$P(R) = 0,07 + 0,10 + 0,11 + 0,12 = 0,40, \quad P(\neg R) = 1 - 0,40 = 0,60.$$

Theo công thức xác suất đầy đủ:

$$\begin{aligned} P(S) &= P(S | R) P(R) + P(S | \neg R) P(\neg R) \\ &= 0,9 \cdot 0,40 + 0,2 \cdot 0,60 = 0,36 + 0,12 = 0,48. \end{aligned}$$

Theo công thức Bayes:

$$P(R | S) = \frac{P(S | R) P(R)}{P(S)} = \frac{0,36}{0,48} = 0,75.$$

Như vậy, khi thiết bị báo mưa, xác suất buổi chiều trời mưa tăng từ 0,40 lên 0,75.

— HẾT —