CÂU HỎI LÝ THUYẾT

Câu 1: So sánh tìm kiếm mù với tìm kiếm có tri thức bổ sung?

=> Tk mù không có thông tin định hướng còn tk có tri thức có thông tin hướng dẫn(Thông tin quá khứ, tương lai)

Câu 2: So sánh tìm kiếm chiều rộng với tìm kiếm chiều sâu ?

|  |  |  |
| --- | --- | --- |
|  | Chiểu rộng | Chiều sâu |
| OPEN | FIFO(Queue). Duyệt hết các nút ở cùng độ sâu rồi mới duyệt đến các nút ở độ sâu tiếp theo | LIFO(Stack). Duyệt hết các nút con cháu... rồi mới duyệt các nút ở cùng độ sâu nếu chưa tìm thấy |
| Hiệu quả | Khi cây trạng thái đích nằm gần gốc của cây tk | Khi trạng thái đích nằm sâu trong cây tk |
| Độ phức tạp | Tốn nhiều bộ nhớ hơn, chậm hơn  trong thực tế |  |
| Kết quả | Chắc chắn sẽ tìm ra kết quả nếu có | Không tìm thấy nếu cây có độ sâu vô hạn hoặc không gian lặp |

Câu 4: So sánh TK tốt nhất đầu tiên với TK nhánh cận ?

|  |  |
| --- | --- |
| TK tốt nhất đầu tiên | TK nhánh cận |
| TK theo chiều rộng với hàm đánh giá h(u) | Sự cải tiến của TK leo đồi kết hợp với hàm đánh giá f(u) |
| Các đỉnh trong OPEN đc sx tăng dần của hàm đánh giá h(u) | Sx danh sách L theo thứ tự tâng dần của hàm đánh giá f(u) và chèn L vào đầu danh sách |
| Khi tìm thấy đích thì dừng lại | Tìm thấy đích thì tiếp tục tìm các đường đi tốt hơn. |

Câu 7: So sánh TK A\* với TK nhánh cận ?

|  |  |
| --- | --- |
| TK A\* | TK nhánh cận |
| Tìm thấy thì dừng | Tìm thấy vẫn tìm tiếp đến khi tìm được chi phí ít nhất để đến đích |
| TK tốt nhất đầu tiên + f(u) | TK leo đồi + f(u) |
| Sx các nút trong OPEN theo thứ tự tăng dần của hàm đánh giá f(u) | Sx danh sách L theo thứ tự tâng dần hàm đánh giá f(u) và đó chèn L vào đầu danh sách |

Câu 8: So sánh TK leo đồi với TK nhánh cận ?

|  |  |
| --- | --- |
| TK leo đồi | TK nhánh cận |
| - TK theo độ sâu dưới sự định hướng của hàm đánh giá h(u) | - Sự cải tiến của TK leo đồi cộng hàm đánh giá f(u) |
| - Gặp đích thì sẽ dừng tìm kiếm | - Gặp đích thì vẫn tiếp tục tk đường đi tốt hơn. |
| S/x ds các nút kề u theo thứ tự tăng dần của hàm đánh giá và chèn ds đó vào đầu OPEN | |

Câu 3: Hàm đánh giá ? g(u), h(u) là gì ?

- Hàm đánh giá là hàm ước lượng, đánh giá mức tốt/xấu, khả năng về đích của mỗi trạng thái

- Với 1 trạng thái u:

+ g(u) là cho phí quá khứ(chi phí đã đi từ điểm xuất phát- đã xác định)

+ h(u) là chi phí tương lai(chi phí còn lại để tới đích), giá trị này càng nhỏ càng tốt.

+ g(u)+h(u): là tổng chi phí đi từ trạng thái xuất phát tới đích có đi qua trạng thái đó,

giá trị này càng nhỏ càng tốt

- Các kĩ thuật sử dụng trong tk với tri thức bổ sung:

+ TK kinh nghiệm: sử dụng hàm đánh giá f(u)=h(u)

-> TK tốt nhất đầu tiên = TK chiều rộng + h(u)

-> TK leo đồi = TK chiều sâu + h(u)

+ TK tối ưu: sử dụng hàm đánh giá f(u)= g(u)+ h(u)

-> TK A\* = TK tốt nhất đầu tiên + f(u)

-> TK nhánh cận = TK leo đồi + f(u)

+ TK có đối thủ

Câu 5: So sánh TK tốt nhất đầu tiên và TK Beam ?

- TK tốt nhất đầu tiên: Duyệt u, phát triển các đỉnh v kề u(đưa tất cả vào open)

- TK Beam: Duyệt u, phát triển k đỉnh tốt nhất ở mỗi độ sâu trong open(đưa những đỉnh n vào OPEN)

=>TK Beam giống tk BFS nhưng hạn chế chỉ phát triển k đỉnh tốt nhất ở mỗi độ sâu.

Câu 6: So sánh TK kinh nghiệm và TK tối ưu ?

=> TK kinh nghiệm sử dụng hàm đánh giá f(u)=h(u) còn TK tối ưu dùng hàm đánh giá

f(u)=h(u)+g(u) và TK tối ưu quan tâm thêm chi phí đường đi nghiệm còn TK kinh nghiệm quan tâm đến chọn đỉnh và rẽ nhánh

Câu 9: Không gian trạng thái của bài toán(trò chơi) ?

- Là đồ thị có hướng trong đó:

+ Mỗi nút là 1 trạng thái

+ Mỗi cạnh ứng với 1 phép chuyển đổi trạng thái.

* Để xây dựng không gian trạng thái cho 1 bài toán ta cần xác định:

1. Trạng thái đầu

- Trạng thái xuất phát. Một bài toán có thể có nhiều trạng thái xuất phát

2. Tập trạng thái đích

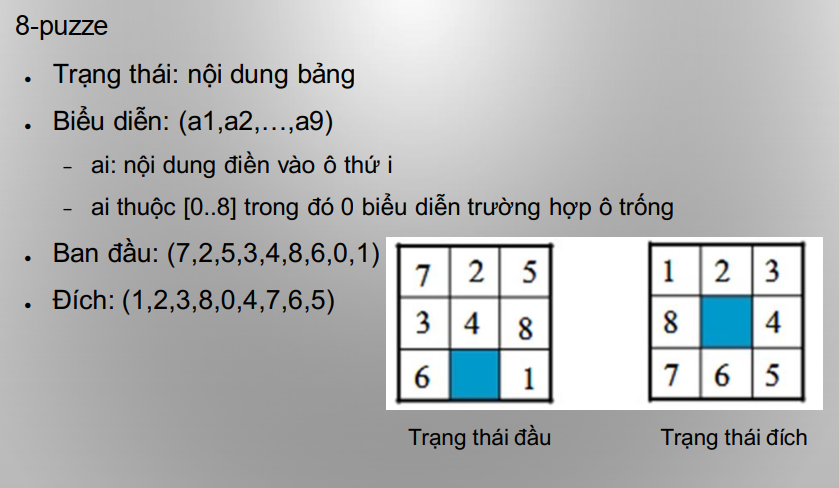
- Trạng thái mà bài toán được giải. Một bài toán có thể có nhiều trạng thái.

3. Các bước chuyển

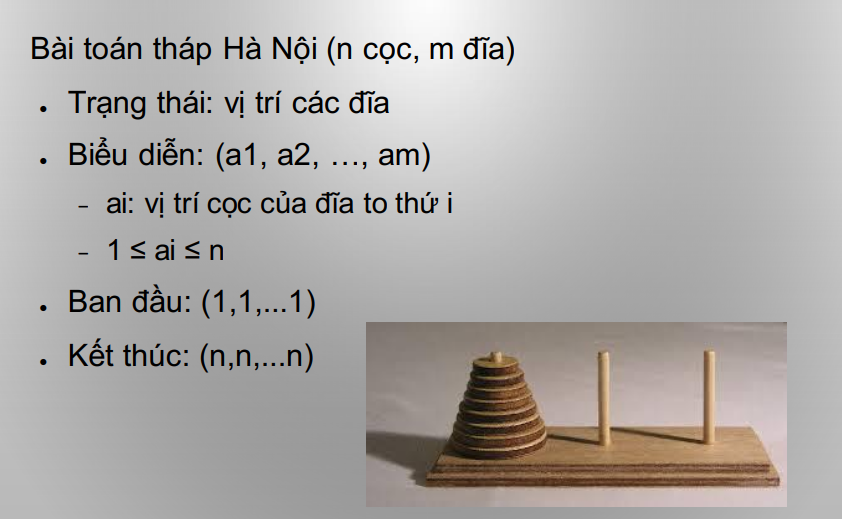
- Thao tác để chuyển từ 1 trạng thái này sang 1 trạng thái khác.

- v gọi là 1 trạng thái kề/con của u nếu có 1 phép chuyển từ u tới v

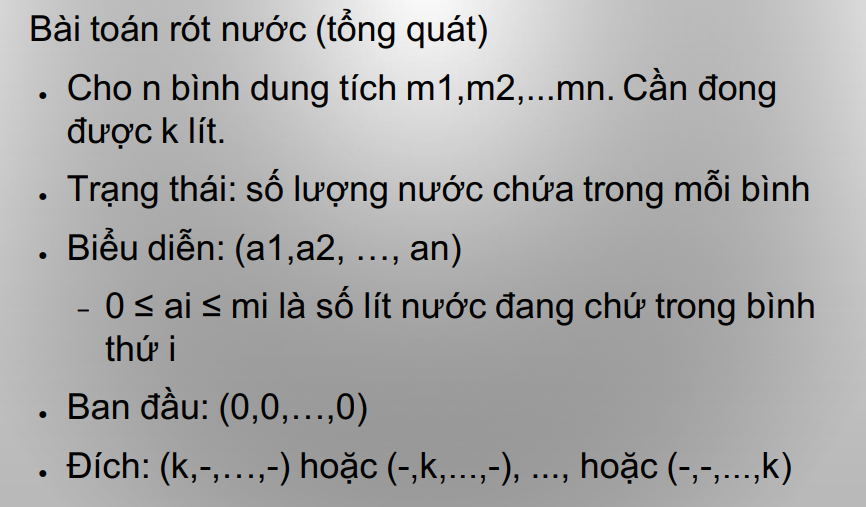
VD: 8-puzzle: có 4 thao tác chuyển là up, down, left, right



Bài toán tháp HN :



Bài toán đổ nước:

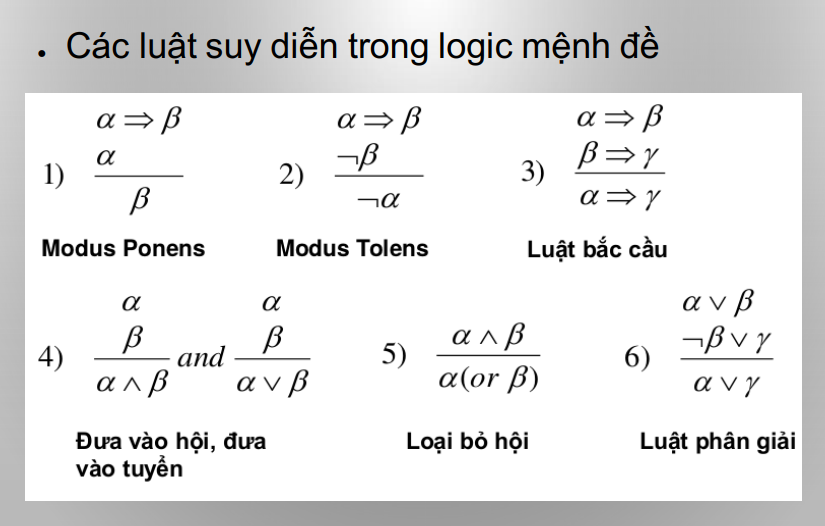


Câu 10: Các phương pháp chứng minh mệnh đề ?

- C1: Sử dụng bảng chân lý

- C2: Phương pháo chứng minh phản chứng của Robinson

- C3: Sử dụng luật suy diễn



- C4: Suy diễn tiến, suy diễn lùi

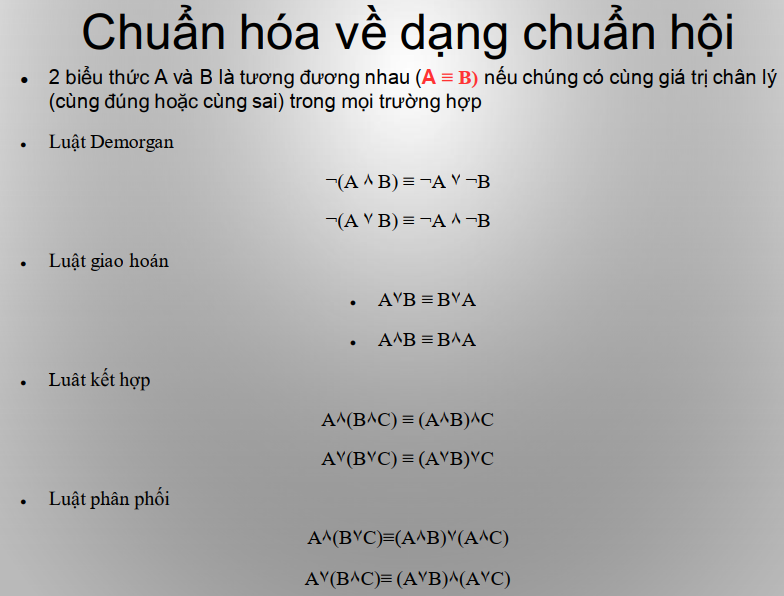
Câu 11: Các bước chứng minh bằng pp phản chứng Robinson ?

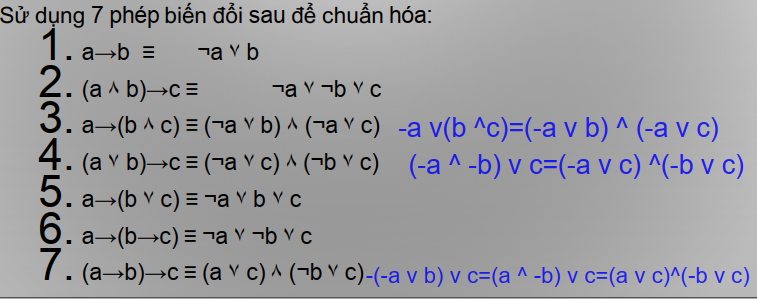
Chứng minh: KL là đúng

- Giả sử KL sai(-KL)

B0: Đưa thêm -KL vào tập giả thiết

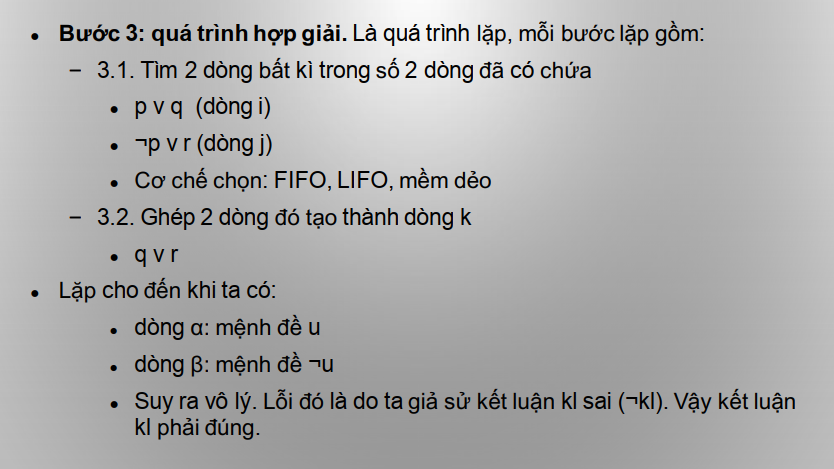
B1: Chuẩn hóa về dạng chuẩn hội





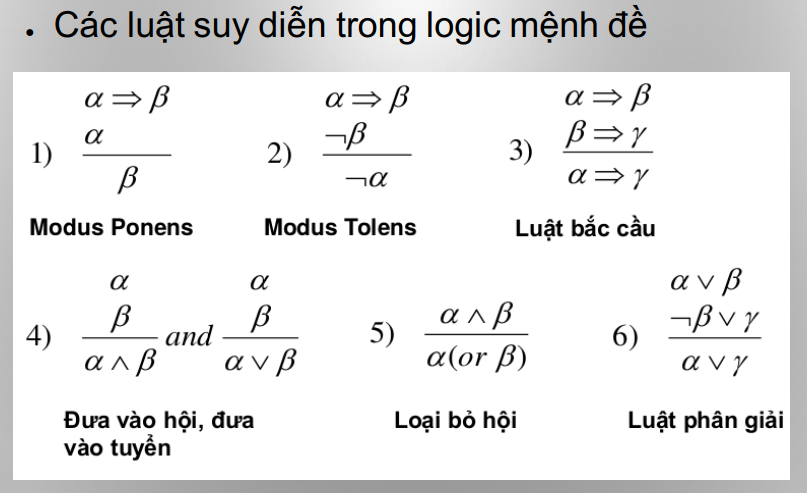
B2: Tách dòng

B3: Hợp giải



Câu 12: Các luật suy diễn trong logic mệnh đề? Suy diễn tiến? Suy diễn lùi ?

* Các luật suy diễn trong logic mđ:



- Suy diễn tiến, lùi:

Xuất phát từ giả thiết dần đi đến kết luận: suy diễn tiến (forward inference)

Xuất phát từ kết luận lùi dần đến giả thiết: suy diễn lùi (backward inference)

Câu 13: Độ sâu và chiều cao của cây ?

* Độ sâu của một nút là số cạnh từ nút đến nút gốc của cây. Một nút gốc sẽ có độ sâu bằng 0.
* Chiều cao của một nút là số cạnh trên con đường dài nhất từ nút đến một chiếc lá. Một nút lá sẽ có chiều cao bằng 0.

Câu 14: Chứng minh trong logic vị từ ?

