Chương 4:
Biểu diễn tri thức

Nội dung

- Giới thiệu về tri thức
- Biểu diễn và ánh xạ
- Các cách tiếp cận
- Các vấn đề trong biểu diễn tri thức
- Vấn đề khung
- …

Tri thức là gì?

- Dữ liệu là các con số, ký hiệu mà máy tính có thể lưu trữ, biểu diễn, xử lý. Bản thân dữ liệu không có ý nghĩa.
- Chỉ khi con người cảm nhận, tư duy thì dữ liệu mới có một ý nghĩa nhất định, đó chính là thông tin.
- Tri thức là kết tinh, cô đọng, chắt lọc của thông tin. Tri thức hình thành do quá trình xử lý thông tin mang lại.

Phân loại tri thức

- Các định lý tóan học, định luật vật lý là các tri thức mang tính khẳng định sự kiện.
- Các phương pháp điều chế hóa học, thuật toán là tri thức mang tính thủ tục.
- Các nhận định, kết luận về sự kiện, hiện tượng là tri thức mô tả.
- Các ước lượng, suy đoán hình thành qua kinh nghiệm là tri thức heuristic

Nhu cầu xử lý tri thức?

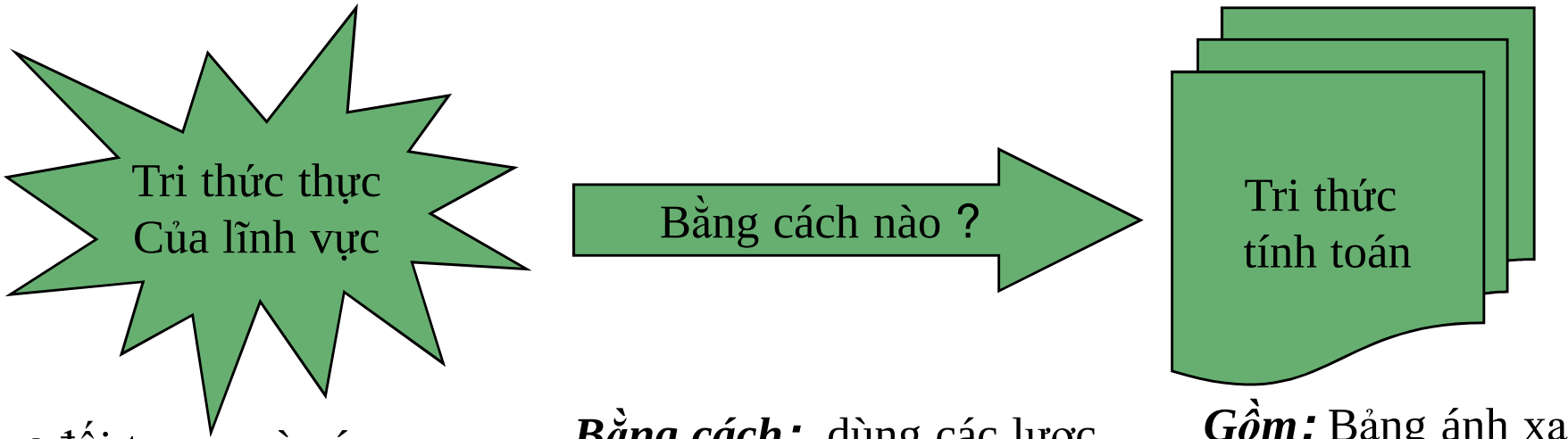
- Trí tuệ, sự thông minh phải dựa trên nền tảng của tri thức. Tuy nhiên, nó còn phụ thuộc vào việc vận dụng, xử lý tri thức.
- Biểu diễn tri thức là việc đưa tri thức vào máy tính. Và chỉ có ý nghĩa nếu công việc tiếp theo: "xử lý tri thức được thực hiện".

Ví dụ về một hệ tri thức

- Cho 2 bình rỗng X, Y có thể tích lần lượt là Vx, Vy. Dùng 2 bình này để đong ra z lít nước.
- Cụ thể với Vx=5, Vy=7 và z=4, ta làm như sau:
  - Múc đầy bình 7
  - Đổ qua cho đầy bình 5.
  - Đổ hết nước trong bình 5
  - Đổ phần còn lại trong bình 7 qua bình 5
  - Múc đầy bình 7
  - Đổ từ bình 7 qua cho đầy bình 5
  - Phần còn lại trong bình 7 là 4 lít

Biểu diễn tri thức

- Là phương pháp mã hoá tri thức, nhằm thành lập cơ sở tri thức cho các hệ thống dựa trên tri thức



Gồm: đối tượng và các quan hệ giữa chúng trong lĩnh vực

Bằng cách: dùng các lược đồ biểu diễn (scheme).
→ Chọn dùng lược đồ cho loại tri thức là vấn đề quan trọng

Gồm: Bảng ánh xạ giữa:
Đối tượng thực → đối tượng tính toán
Quan hệ thực → quan hệ tính toán

Lược đồ biểu diễn tri thức

- Lược đồ logic
  - Dùng các biểu thức trong logic hình thức, như phép toán vị từ, để biểu diễn tri thức.
  - Các luật suy diễn áp dụng cho loại lược đồ này
  - Ngôn ngữ lập trình hiện thực tốt nhất cho loại lược đồ này là: PROLOG
- Lược đồ thủ tục
  - Biểu diễn tri thức như tập các chỉ thị lệnh để giải quyết vấn đề
  - Các chỉ thị lệnh trong lược đồ thủ tục chỉ ra bằng cách nào giải quyết vấn đề

Lược đồ biểu diễn tri thức…

- Lược đồ mạng
  - Biểu diễn tri thức như là đồ thị; các đỉnh như là các đối tượng hoặc khái niệm, các cung như là quan hệ giữa chúng
  - Các ví dụ về loại lược đồ này gồm: **mạng ngữ nghĩa**
- Lược đồ cấu trúc
  - Là một mở rộng của lược đồ mạng; bằng cách cho phép các nút có thể là một CTDL phức tạp gồm các khe (slot) có tên và trị hay một thủ tục
  - Kịch bản (script), khung (frame), đối tượng (object) là ví dụ của lược đồ này

Biểu diễn và ánh xạ

- Tri thức của lĩnh vực:
  - Là toàn bộ những hiểu biết về lĩnh vực đó
  - Gồm: khái niệm, đối tượng, quan hệ giữa chúng, luật tồn tại giữa chúng, …
  - Hiện tồn tại 1 số lược đồ ghi nhận tri thức
- Để giải bài toán AI cần:
  - Tri thức về bài toán (có thể nhiều)
  - Phương tiện để xử lý tri thức như: retrieve, update, infer,

Biểu diễn và ánh xạ…

- Hình thức hóa tri thức

Dạng thực	Dạng hình thức
- **Facts** (sự kiện): sự thật trong lĩnh vực	- **Representations** (sự biểu diễn): dạng biểu diễn của sự kiện theo lược đồ được chọn.
	
Cái cần biểu diễn	Cái có thể xử lý được

Biểu diễn và ánh xạ …

- Hai mức cấu trúc cho facts/representations
  - Mức tri thức:
    - Mức mà các sự kiện, gồm cách hành xử của agent (tác tử) và mục tiêu hiện tại, được mô tả.
  - Mức ký hiệu:
    - Mức mà sự biểu diễn của các đối tượng đã được chọn trong mức tri thức được viết ra ở dạng ký hiệu để có thể xử lý được bằng chương trình

Biểu diễn và ánh xạ …

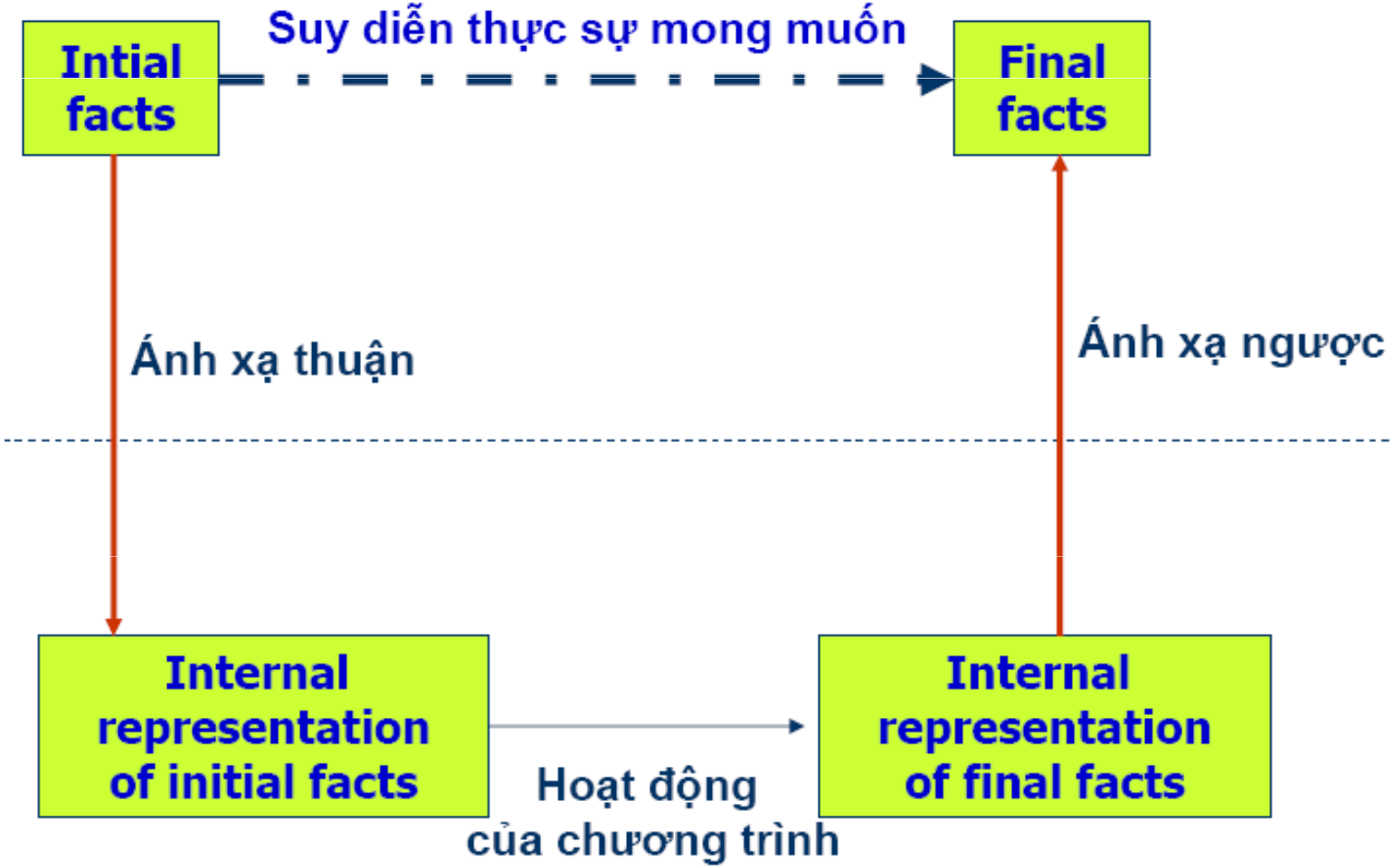
- Ví dụ:
  - Câu tiếng Anh:
    - “Spot is a dog”
    - “Every dog has a tail”
  - Có thể được biểu diễn ở nhiều lược đồ
- Dạng logic (chương sau):
  - 1. dog(Spot).
  - 2. $\forall X(dog(X) \rightarrow hastail(X))$.
  - Từ đó câu: “Spot has a tail”, có thể thu được qua các bước:
  - 3. Từ 2, X=“Spot”: $dog(Spot) \rightarrow hastail(Spot)$.
  - 4. Từ 1, 3: hastail(Spot).
  - Ánh xạ ngược → “Spot has a tail”.

Biểu diễn và ánh xạ …

- Dạng mạng ngữ nghĩa (chương sau):

Biểu diễn và ánh xạ …

- Mô hình giải quyết vấn đề của con người và máy



Các cách tiếp cận

- Bốn thuộc tính của hệ thống biểu diễn tri thức:
  - Khả năng biểu diễn tất cả các tri thức cần thiết cho lĩnh vực đó.
  - Khả năng xử lý các cấu trúc sẵn có để sinh ra các cấu trúc mới tương ứng với tri thức mới được sinh ra từ tri thức cũ.
  - Khả năng thêm vào cấu trúc những tri thức thông tin bổ sung mà nó có thể được dùng để hướng dẫn cơ chế suy luận theo hướng có nhiều triển vọng nhất.
  - Khả năng thu được thông tin mới dễ dàng.
    - Trường hợp đơn giản nhất là chèn trực tiếp tri thức mới (do con người) vào cơ sở tri thức. Lý tưởng nhất là chương trình có thể kiểm soát việc thu được tri thức.

Các cách tiếp cận …

- Năng lực hiện nay:
  - Không một hệ thống nào có thể tối ưu tất cả các khả năng trên cho mọi kiểu tri thức.
  - Nhiều kỹ thuật dùng cho biểu diễn tri thức cùng tồn tại.
  - Chương trình thường dùng nhiều hơn 1 kỹ thuật biểu diễn.

Các cách tiếp cận …

- Tri thức quan hệ đơn giản:
  - Biểu diễn các sự kiện (facts) dạng khai báo như tập quan hệ đã dùng trong CSDL quan hệ - xem ví dụ sau

Player	Height	Weight	Bats-Throws
Hank Aaron	6-0	180	Right-Right
Willie Mays	5-10	170	Right-Right
Babe Ruth	6-2	215	Left-Left
Ted Williams	6-3	205	Left-Right

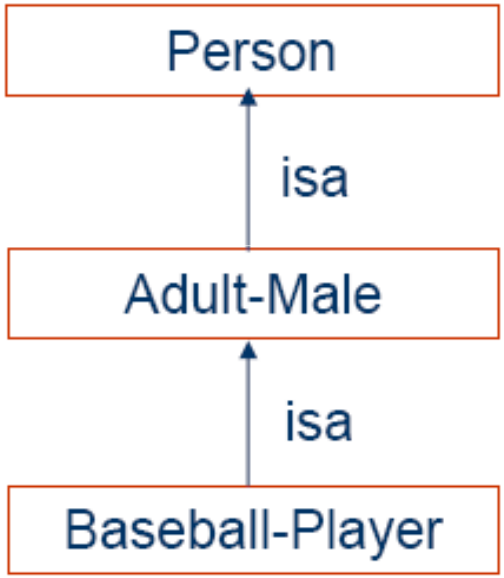
  - Dạng biểu diễn này:
    - **Đơn giản ← khả năng suy luận trên chính nó rất yếu.**
    - **Thường được dùng làm đầu vào cho các động cơ suy diễn mạnh hơn.**

Các cách tiếp cận …

- Tri thức có khả năng thừa kế:
  - Một dạng bổ sung cơ chế suy diễn vào cơ sở tri thức quan hệ nói trên, đó là: thừa kế thuộc tính.
  - Thừa kế thuộc tính:
    - Tổ chức các đối tượng thành các lớp (class).
    - Các lớp được sắp xếp vào hệ thống phân cấp (hierachy) – có lớp cha (tổng quát) và lớp con (cụ thể).
    - → Các lớp con thừa kế các thuộc tính từ lớp cha.

Các cách tiếp cận …

- Tri thức có khả năng thừa kế (tt.):
  - **Line:**
    - Thuộc tính
  - **Box:**
    - Đối tượng, Trị (Value) của thuộc tính của đối tượng.
  - **Arrow:**
    - Từ đối tượng sang trị của thuộc tính.



Quan hệ giữa các lớp dùng mạng ngữa nghĩa.

Các cách tiếp cận …

- Tri thức suy diễn:
  - Thừa kế thuộc tính ở trên là 1 dạng suy diễn
  - Logic truyền thống: cung cấp dạng suy diễn mạnh hơn
  - Tri thức suy diễn: cần thủ tục suy diễn
  - Thủ tục suy diễn: nhiều dạng
    - **Forward (tiến): Đi từ sự kiện đến kết luận**
    - **Backward (lùi): Đi từ kết luận đến sự kiện đã cho**
    - **Thủ tục thường dùng: resolution – xem chương 5.**

Các cách tiếp cận …

- Tri thức thủ tục:
  - Tri thức trong các ví dụ trước: Tĩnh, dạng khai báo
  - Một dạng tri thức khác: chỉ ra hành động được thi hành khi điều kiện nào đó thoả → tri thức thủ tục
  - Cách biểu diễn trong chương trình
    - **Viết bằng các NNLT (LISP chẳng hạn)**
    - **⇒ Máy sẽ thực thi mã để thực hiện công việc**
  - Trở ngại
    - **Khó viết CT suy diễn về hành vi của CT khác**
    - **Cập nhật/debug số lượng lớn mã → khó khăn**

Các cách tiếp cận …

- Tri thức thủ tục (tt):

```
(defun fun1 (lis)
   (cond
      ((null lis) 0)
      ((not (listp (car lis)))
           (cond
               ((eq (car lis) nil) (fun1 (car lis)))
               (T (+ 1 (fun1 (cdr lis))))
           )
       )
       (T (+ (fun1 (car lis)) (fun1 (cdr lis))))
   )
)
```

Các cách tiếp cận …

- Tri thức thủ tục (tt): dùng luật sinh (production rule)
  - Luật sinh và cách sử dụng chúng: là định hướng hoạt động hơn các dạng biểu diễn nói trước đây.
  - Tuy phân biệt đâu là tri thức khai báo hay thủ tục là một công việc khó khăn.

```
IF:
        ninth inning, AND
        score is close, AND
        less than 2 outs, AND
        first base is vacant, AND
        batter id better hitter than next batter
THEN
        walk the batter.
```

Các vấn đề trong biểu diễn tri thức

- Có những thuộc tính cơ bản nào của đối tượng mà chúng xuất hiện trong mọi lĩnh vực không?
- Nếu có: đó là những thuộc tính nào?
- Có chắc chắn là chúng sẽ được xử lý thích hợp trong từng cơ chế được đề nghị không?
- Có quan hệ quan trọng nào tồn tại cùng với thuộc tính không?

Các vấn đề trong biểu diễn tri thức

- Tri thức được biểu diễn đến mức chi tiết nào?
- Có tồn tại những *primitive* cơ bản mà qua đó tất cả tri thức được biểu diễn?
- Sử dụng *primitives* có ích không?
- Tập các đối tượng được biểu diễn như thế nào?
- Với số lượng lớn tri thức được chứa trong CSDL, Bằng cách nào truy xuất những thành phần cần thiết?

Các vấn đề trong biểu diễn tri thức

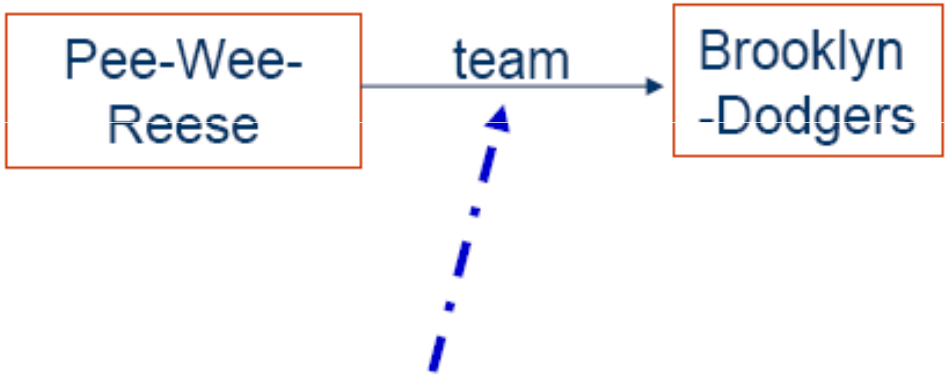
- **Các thuộc tính quan trọng:**
  - 1. Instance:
    - Cho biết quan hệ thành viên giữa đối tượng và lớp nó thuộc vào
  - 2. Isa
    - Cho biết một lớp là con của lớp khác
    - Cặp thuộc tính trên cho phép khả năng thừa kế thuộc tính
    - Chúng có thể được gọi và biểu diễn khác nhau trong nhiều hệ thống tri thức

Các vấn đề trong biểu diễn tri thức

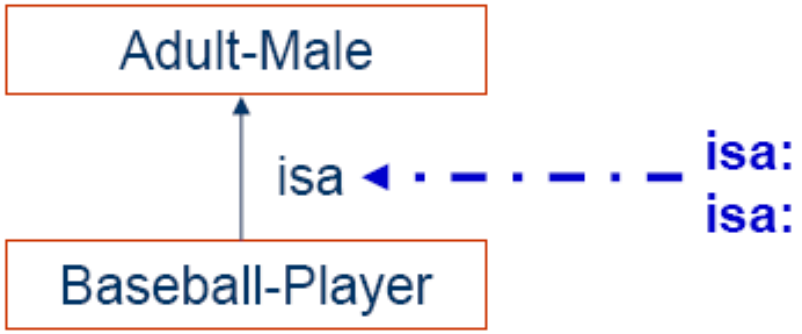
- Các quan hệ cùng với các thuộc tính:
  - Thuộc tính: entity | relationship
  - Có tính chất quan trọng:
    - Đảo
    - Tồn tại trong một hệ thống Isa
    - Các kỹ thuật để suy diễn giữa các giá trị
    - Các thuộc tính đơn trị

Các vấn đề trong biểu diễn tri thức

- Các quan hệ cùng với các thuộc tính:



Team: thuộc của đối tượng <Pee-Wee-Reese>
Team: quan hệ giữa <Pee-Wee-Reese> và <Brooklyn-Dodgers>



isa: thuộc tính của Baseball-Player
isa: quan hệ từ <Baseball-Player> đến Adult-Male

Vấn đề khung

- Khung
  - Mỗi frame mô tả một *đối tượng (object)*.
  - Một frame bao gồm 2 thành phần cơ bản là **slot** và **facet.**
  - Một **slot** là một thuộc tính đặc tả đối tượng được biểu diễn bởi frame. Ví dụ : trong frame mô tả xe hơi, có hai slot là *trọng lượng* và *loại máy*.
  - Mỗi slot có thể chứa một hoặc nhiều **facet.**
  - Các facet (đôi lúc được gọi là slot "con") đặc tả một số thông tin hoặc thủ tục liên quan đến thuộc tính được mô tả bởi slot. Facet có nhiều loại khác nhau, sau đây là một số facet thường gặp: *value*, *default value*, *range*,…

Vấn đề khung…

- Bằng cách nào biểu diễn hiệu quả chuỗi trạng thái cho bài toán tìm kiếm?
- Bài toán robot:
  - on(Plant12, Table34).
  - under(Table34, Window13).
  - in(Table34, Room15).

  → 1 trạng thái = danh sách các facts trên.

  → bất tiện: danh sách dài.

  từ trạng thái A → B: nhiều facts không thay đổi.
- Vấn đề khung: bài toán về biểu diễn facts thay đổi cùng với những facts không được biết.

Vấn đề khung …

- Sử dụng các tiền đề khung:
  - Mô tả tất cả những cái sẽ không thay đổi khi áp dụng 1 toán tử cụ thể nào đó để chuyển từ trạng thái n → n+1
  - Ví dụ:
    - "Vật X có màu Y tại trạng thái S1 thì cũng có màu Y tại trạng thái S2 khi di chuyển X từ S1 → S2'
    - color(X,Y, S1) ^ move(X,S1,S2) → color(X,Y,S2).
    - Bất tiện:

      → Số tiền đề nhiều.

Vấn đề khung …

- Sử dụng giả định:
  - Những cái thay đổi: ghi tường minh hoặc được dẫn ra 1 cách logic từ những cái thay đổi.

- Hai cách tiếp cận dùng cho backtrack trên chuỗi trạng thái:
  - Không thay đổi mô tả đầu. Ghi nhận sự thay đổi cụ thể tại node cần thay đổi.
  - Thay đổi mô tả đầu. Ghi nhận những gì cần làm khi undo tại trạng thái đó.