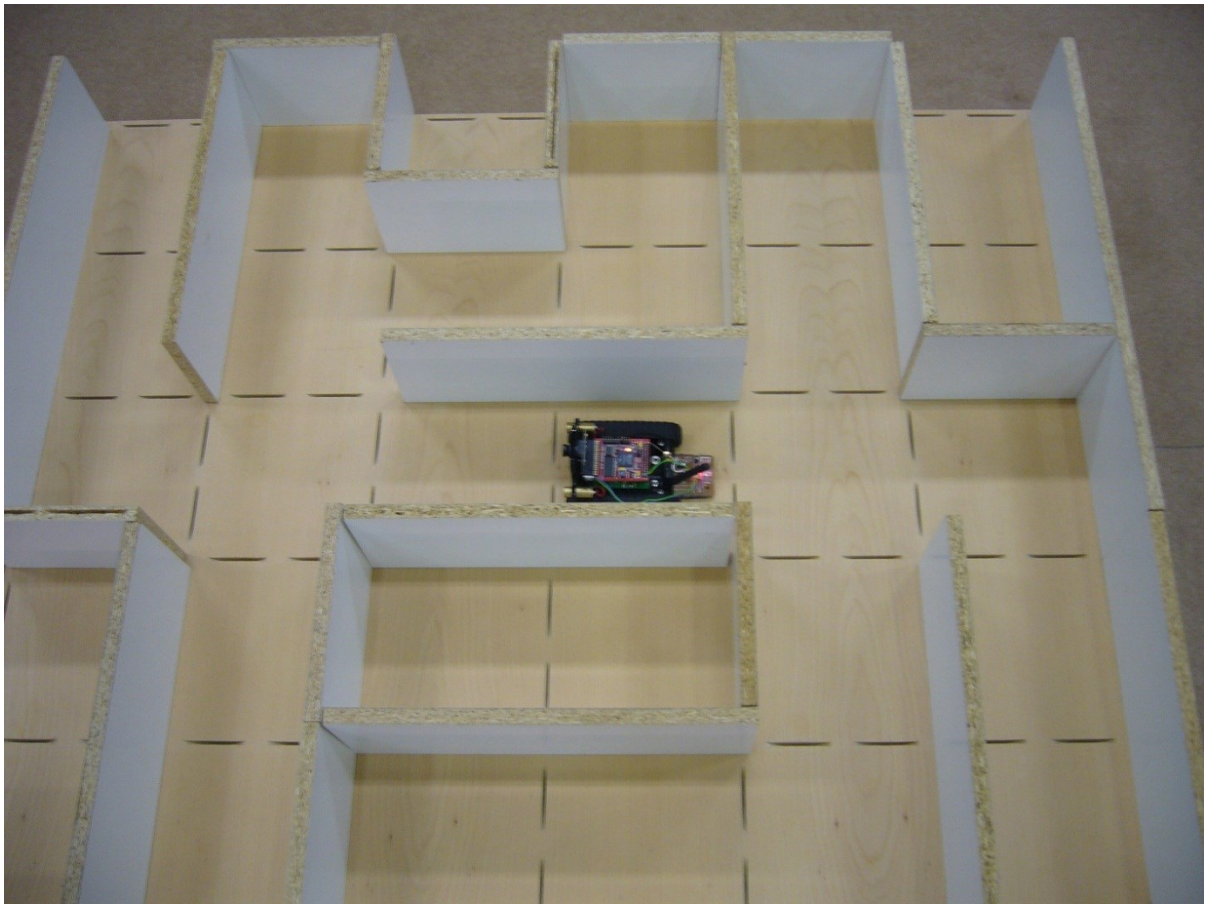
HỌC TĂNG CƯỜNG

NỘI DUNG

1. Giới thiệu
2. Học tăng cường
3. Các thành phần của RL
4. Ví dụ (K – Armed Bandit)
5. Model-Based Learning

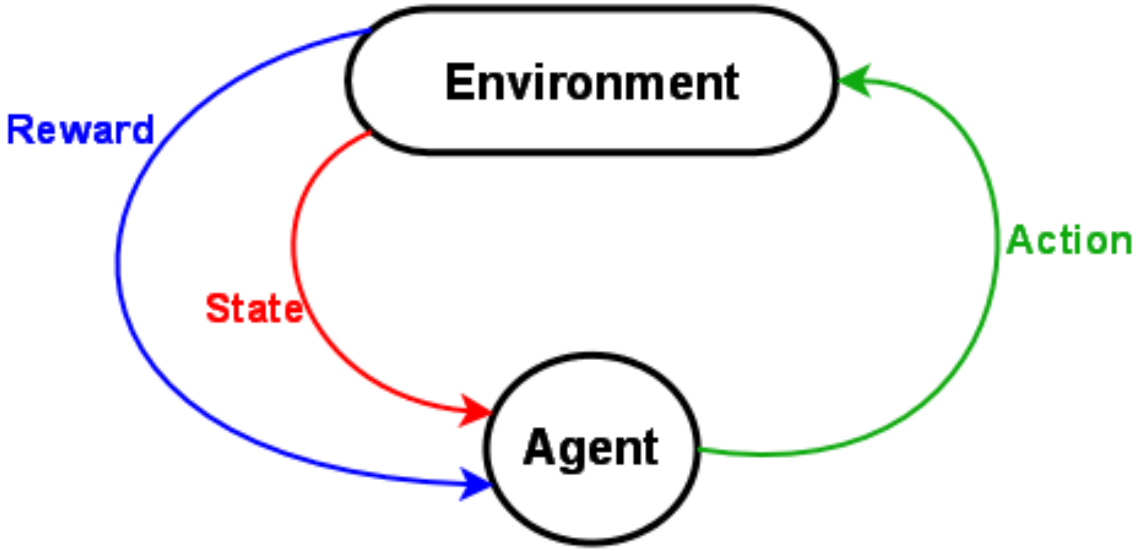
Giới thiệu

- Robot trong mê cung: Một chuỗi các hành động để tìm được lối ra.



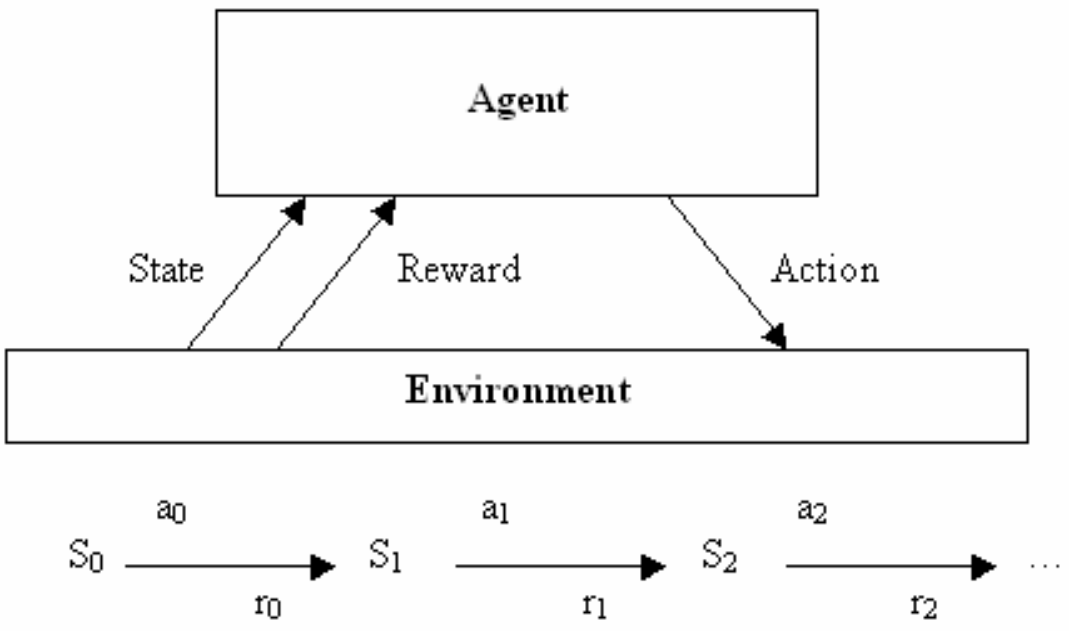
Giới thiệu

- Hai ứng dụng có một số điểm chung: đó là có 1 ***decision maker*** được gọi là ***agent***, được đặt trong một môi trường (***environment***).
- ***Agent*** tương tác với môi trường. Tại một trạng thái bất kì của môi trường, ***agent*** thực hiện 1 hành động để thay đổi trạng thái và nhận được 1 điểm thưởng.



Học Tăng Cường
(Reinforcement Learning)

- Học tăng cường nghiên cứu cách thức một *agent* trong một môi trường nên chọn thực hiện các hành động nào để cực đại hóa điểm thưởng (reward) nào đó về lâu dài.



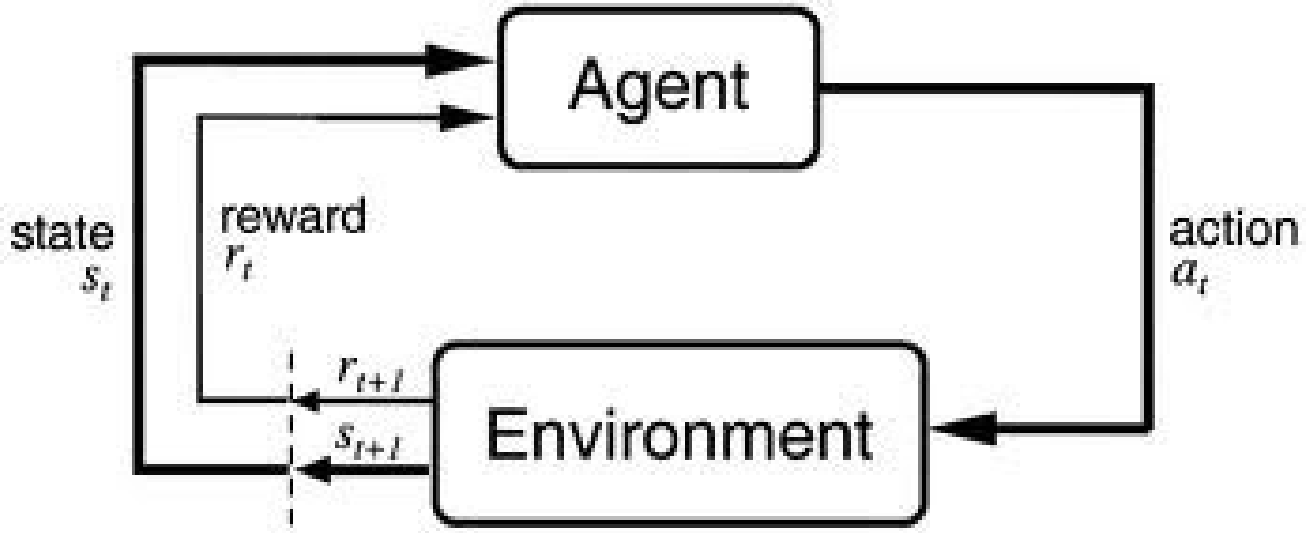
Goal: learn to choose actions that maximize:

$r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$, where $0 \leq \gamma < 1$

Học Tăng Cường

Một cách hình thức, mô hình học tăng cường bao gồm:

- S: tập các trạng thái của môi trường
- A: tập các hành động
- R: tập các điểm thưởng (reward)



Các thành phần của RL

- Policy π: *l*à một ánh xạ từ tập trạng thái của môi trường tới tập các hành động
- Markov Decision Process là bộ bốn:

$$(S, A, R(s_{t+1}|s_t, a_t), P(s_{t+1}|s_t, a_t)), \text{ trong đó:}$$

  - S là tập các trạng thái.
  - A là tập các hành động (nói cách khác, A_s là tập hữu hạn các hành động có thể từ trạng thái s).
  - $P(s_{t+1}|s_t, a_t)$ là xác suất hành động a trong trạng thái s tại thời điểm t sẽ chuyển qua trạng thái s_{t+1} tại thời điểm t+1.
  - $R(s_{t+1}|s_t, a_t)$ là điểm thưởng trực tiếp (hay điểm thưởng kỳ vọng trực tiếp) nhận được sau khi chuyển từ trạng thái s_t sang trạng thái s_{t+1} với xác suất $P(s_{t+1}|s_t, a_t)$

Các thành phần của RL

- **Giá trị của 1 policy:** $V^{\pi}(s_t)$
- **Finite-horizon:**

$$V^{\pi}(s_t) = E\left[r_{t+1} + r_{t+2} + \cdots + r_{t+T}\right] = E\left[\sum_{i=1}^{T} r_{t+i}\right]$$

- **Infinite-horizon:**

$$V^{\pi}(s_t) = E\left[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots\right] \qquad \left[\sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i}\right]$$

$0 \le \gamma < 1$ là hệ số giảm

- Chiến lược tối ưu: $\pi^* = \text{argmax}\, V^*$
- Hàm giá trị tối ưu: $V^*(s_t) = \max_{\pi}{}^{\pi} V^{\pi}(s_t), \forall\, s_t$

$$V^*(s_t) = \max_{\pi} V^{\pi}(s_t), \forall s_t$$

$$= \max_{a_t} E\left[\sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i}\right]$$

$$= \max_{a_t} E\left[r_{t+1} + \gamma \sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i+1}\right]$$

$$= \max_{a_t} E\left[r_{t+1} + \gamma V^*(s_{t+1})\right]$$

Bellman's equation

$$V^*(s_t) = \max_{a_t}\left(E[r_{t+1}] + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) V^*(s_{t+1})\right)$$

$$V^*(s_t) = \max_{a_t} Q^*(s_t, a_t)$$ giá trị của a_t ở trạng thái s_t

$$Q^*(s_t, a_t) = E[r_{t+1}] + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})$$

Ví dụ: Single State: K-armed Bandit

- Chọn lever trả ra điểm thưởng cao nhất.

  $Q(a)$: giá trị của hành động a

  Điểm thưởng là r_a

- Nếu điểm thưởng xác định thì:

  Đặt $Q(a) = r_a$

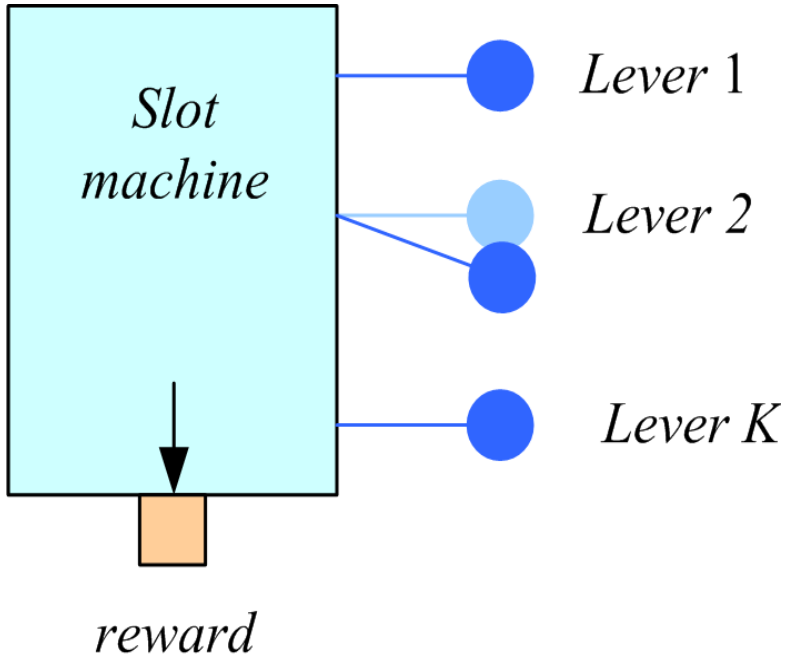
  Chọn a^* nếu

  $Q(a^*)=\max_a Q(a)$

- Nếu điểm thưởng không xác định hoặc ngẫu nhiên thì:

  Dự đoán giá trị của hành động a tại thời điểm t là $Q_t(a)$:

$$Q_{t+1}(a) \leftarrow Q_t(a) + \eta\left[r_{t+1}(a) - Q_t(a)\right]$$



Model-Based Learning

- $P(s_{t+1} \mid s_t, a_t), p(r_{t+1} \mid s_t, a_t)$ được biết trước.
- Không cần phải khám phá (exploration)
- Có thể được giải bằng quy hoạch động
- Giải phương trình sau:

$$V^*(s_t) = \max_{a_t}\left(E[r_{t+1}] + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) V^*(s_{t+1}) \right)$$

- Policy tối ưu:

$$\pi^*(s_t) = \arg\max_{a_t}\left(E[r_{t+1}|s_t, a_t] + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) V^*(s_{t+1}) \right)$$

Value Iteration

- Để tìm các chính sách tối ưu, chúng ta có thể sử dụng hàm giá trị tối ưu, và có một thuật toán lặp đi lặp lại đã được chứng minh là hội tụ về các giá trị V * đúng.
- Giá trị hội tụ nếu giá trị chênh lệch tối đa giữa hai lần lặp lại nhỏ hơn một ngưỡng δ nhất định:

$$\max_{s \in S} \left| V^{(l+1)}(s) - V^{l}(s) \right| < \delta$$

Initialize $V(s)$ to arbitrary values
Repeat
  For all $s \in \mathcal{S}$
    For all $a \in \mathcal{A}$
      $Q(s,a) \leftarrow E[r|s,a] + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V(s')$
    $V(s) \leftarrow \max_a Q(s,a)$
Until $V(s)$ converge

Policy Iteration

- Lưu trữ và cập nhật các chiến lược thay vì làm điều này gián tiếp qua các giá trị.
- Ý tưởng là để bắt đầu với 1 chiến lược và cải thiện liên tục cho đến khi không có thay đổi.

Initialize a policy π arbitrarily

Repeat

$\pi \leftarrow \pi'$

Compute the values using π by

solving the linear equations

$$V^{\pi}(s) = E[r|s, \pi(s)] + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, \pi(s)) V^{\pi}(s')$$

Improve the policy at each state

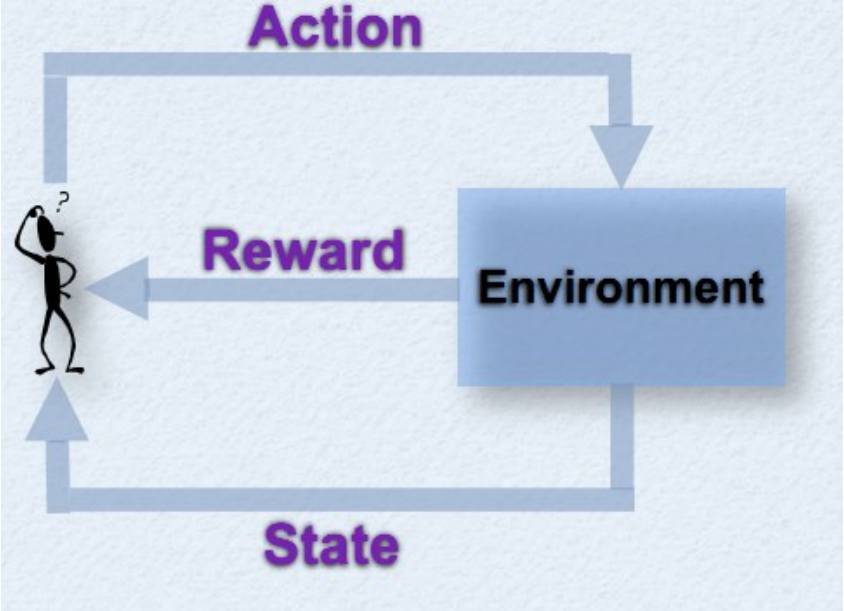
$$\pi'(s) \leftarrow \arg\max_a (E[r|s, a] + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) V^{\pi}(s'))$$

Until $\pi = \pi'$

NỘI DUNG

- Giải thuật Q learning trong môi trường xác định.
- Thí dụ minh họa cho giải thuật.
- Chứng minh hội tụ
- Chiến lược chọn hành động.
- Chiến lược cập nhật.

Reinforcement learning



Tìm chiến lược(policy) tối ưu khi không biết hàm dịch chuyển trạng thái hoặc mô hình điểm thưởng?

Temporal different learning

+ Môi trường, $P(s_{t+1}|s_t, a_t)$, $P(r_{t+1}|s_t, a_t)$, là không biết; (model-free learning).
+ "temporal difference" có nghĩa là sự khác nhau giữa giá trị hành động ở hiện tại và giá trị giảm dần từ tình trạng kế tiếp.
+ Một phiên bản của temporal different learning là Q learning. (giới thiệu Watkins, 1989)

Q Learning

Ký hiệu Q(s,a): hàm ước lượng giá trị tích lũy lớn nhất khi thực hiện hành động a tại trạng thái s. Xét trong môi trường xác định:

$$Q(s\ ,a\ ) \equiv r\ (s,a) + \gamma V^{*}(\delta(s,a))$$

Hàm tối ưu policy:

$$\pi^{*}(s) = \arg\max_{a}\left(Q(s,a)\right)$$

- Cách nào tính được Q(s,a) mà không dùng r(s,a) và δ(s,a) ?

Hàm Q: loại bỏ $V^*(\delta(s,a))$

Theo định nghĩa hàm Q(s,a) và $V^*(s)$. Ta có:

$$V^*(s) = \max_{a'} Q(s,a')$$

Như vậy thì:

$$Q(s,a) \equiv r(s,a) + \gamma V^*(\delta(s,a))$$

Được viết lại:

$$Q(s,a) \equiv r(s,a) + \gamma \max_{a'} Q(\delta(s,a),a').$$

Hàm Q: Loại bỏ r(s,a) và δ(s,a)

- Ở tại trạng thái s, ta chỉ cần biết giá trị tạm thời r có được do thực hiện một hành động a và chỉ cần biết trạng thái kết quả s' nào đó.
- Thay thế r(s,a) và δ(s,a) bằng r và s':

$$\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a').$$

Ký hiệu: $\hat{Q}(s,a)$ ước lượng giá trị hàm Q.

Giải thuật Q learning

Xét trong môi trường xác định

1. **Khởi tạo:** $\hat{Q}(s,a) = 0 \ \forall s \in S, a \in A.$
2. **Quan sát trạng thái khởi tạo s.**
3. **Lặp vô tận:**

   a. Chọn và thực hiện một hành động a

   b. Nhận được điểm thưởng r.

   c. Quan sát trạng thái kết quả s'.

   d. Cập nhật $\hat{Q}(s,a)$ theo:

$$\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a').$$

   e. Lấy s ← s'.

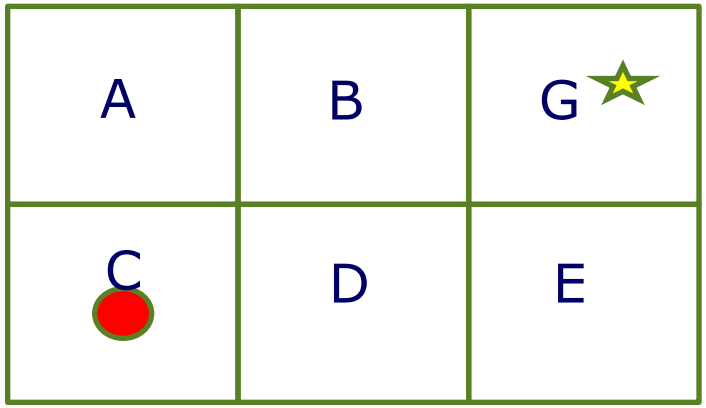
Ví dụ: Gridworld

Bài toán: Cho một dãy các căn phòng nằm kề nhau. Đặt một con robot đặt ở trong một căn phòng bất kỳ. Mục tiêu là giúp cho con robot đi đến căn phòng mong muốn với đường đi là ngắn nhất.

Ví dụ Gridworld

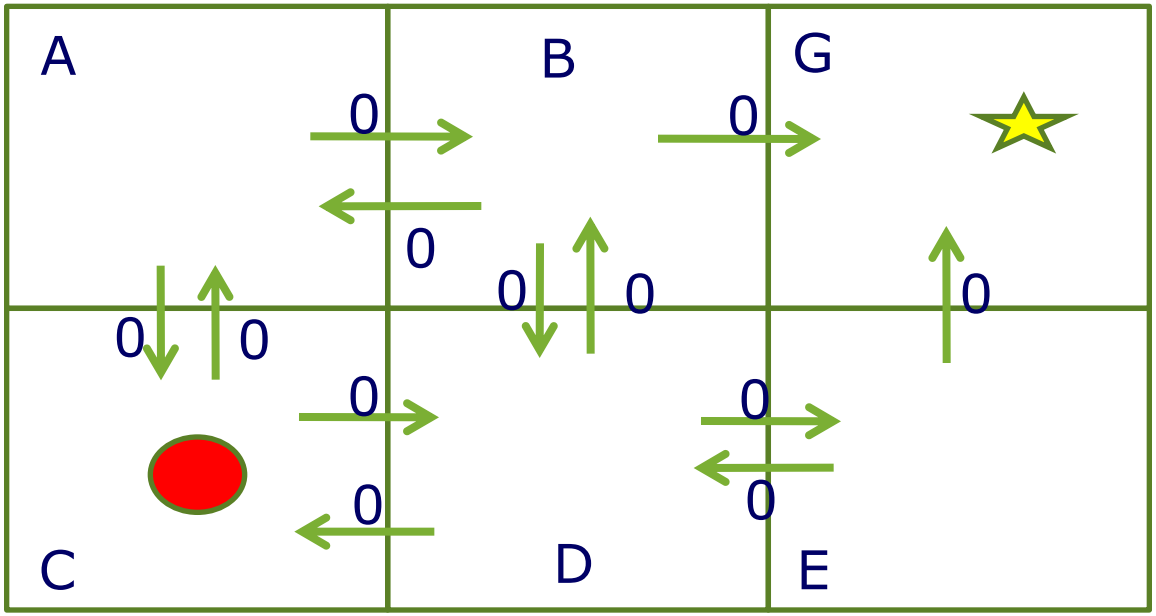
Cho biết $\gamma = 0.9$ và bảng giá trị điểm thưởng:

	A	B	C	D	E	G
A	----	0	0	----	----	----
B	0	----	----	0	----	100
C	0	----	----	0	----	----
D	----	0	0	----	0	----
E	----	----	----	0	----	100
G	----	----	----	----	----	0



Ví dụ Gridworld

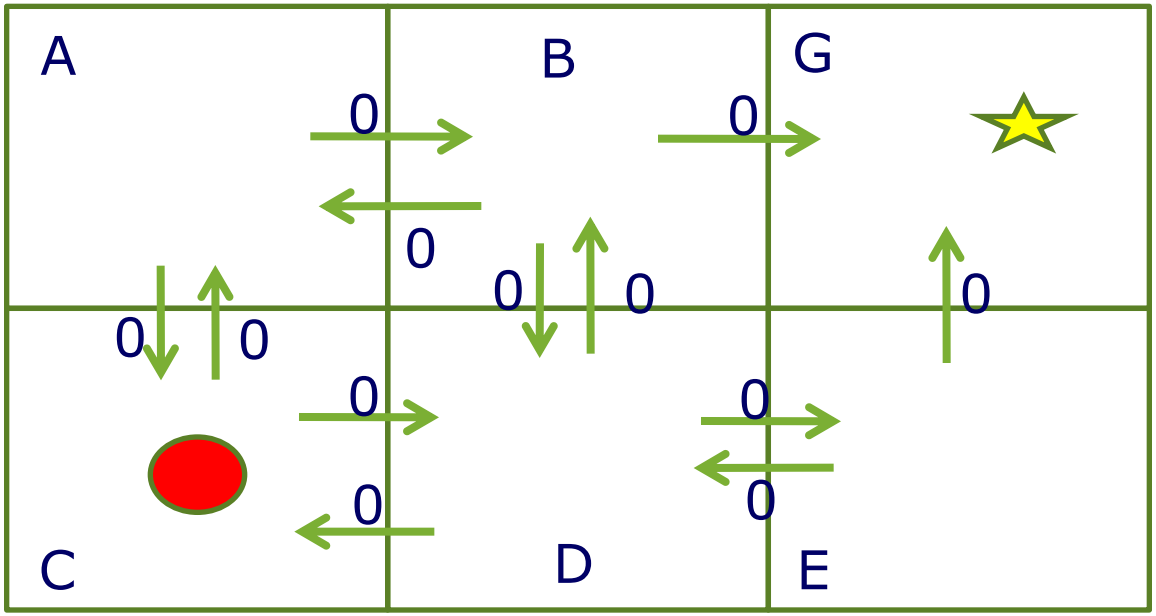
Bước 1: Khởi tạo $\hat{Q}(s,a)=0$ cho mọi cặp (s,a).



	A	B	C	D	E	G
A	----	0	0	----	----	----
B	0	----	----	0	----	100
C	0	----	----	0	----	----
D	----	0	0	----	0	----
E	----	----	----	0	----	100
G	----	----	----	----	----	0

Ví dụ Gridworld

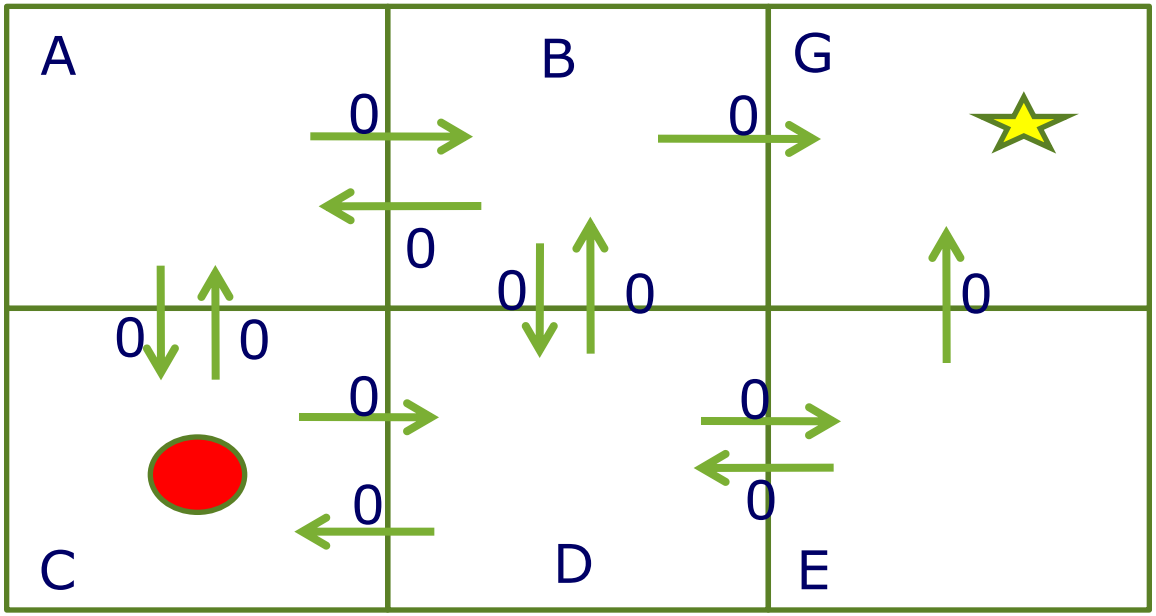
Bước 2: Quan sát tại trạng thái C. Có 2 hành động:



	A	B	C	D	E	G
A	----	0	0	----	----	----
B	0	----	----	0	----	100
C	0	----	----	0	----	----
D	----	0	0	----	0	----
E	----	----	----	0	----	100
G	----	----	----	----	----	0

Ví dụ Gridworld

Bước 3: Vòng lặp. Giả sử chọn hành động lên:



	A	B	C	D	E	G
A	----	0	0	----	----	----
B	0	----	----	0	----	100
C	0	----	----	0	----	----
D	----	0	0	----	0	----
E	----	----	----	0	----	100
G	----	----	----	----	----	0

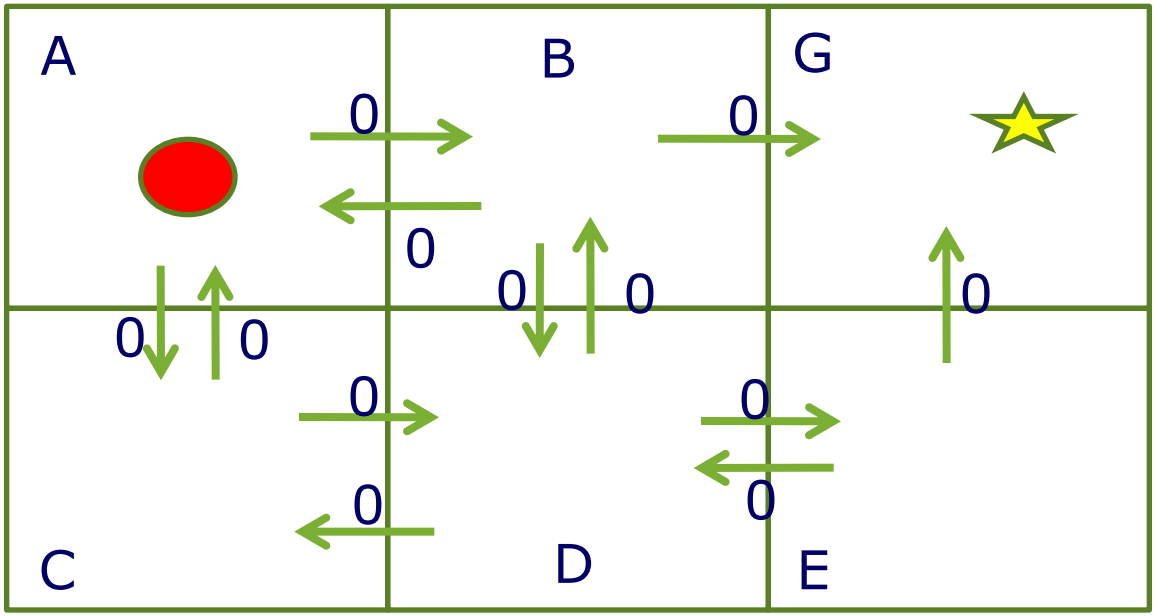
$$\hat{Q}(C, a_{up}) \leftarrow r + \gamma \max_{a'} \hat{Q}(A, a').$$

$$\hat{Q}(C, a_{up}) \leftarrow 0 + 0.9 \max\{0, 0\}$$

$$\hat{Q}(C, a_{up}) \leftarrow 0$$

Ví dụ Gridworld

Bước 3: Từ A giả sử chọn hành động qua B.



	A	B	C	D	E	G
A	----	0	0	----	----	----
B	0	----	----	0	----	100
C	0	----	----	0	----	----
D	----	0	0	----	0	----
E	----	----	----	0	----	100
G	----	----	----	----	----	0

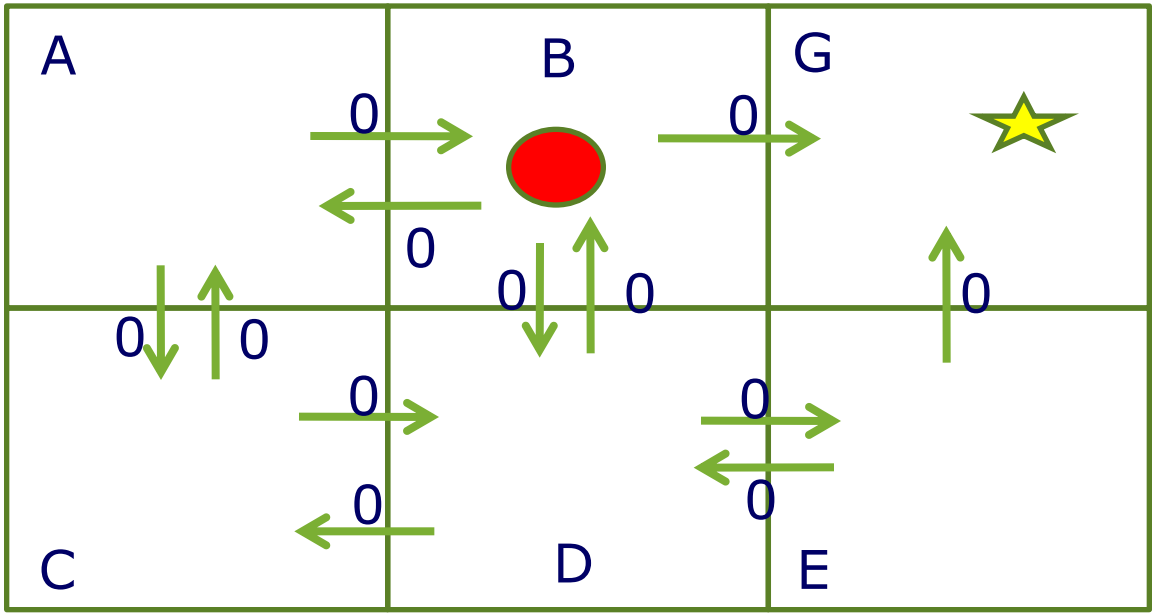
$$\hat{Q}(A, a_{right}) \leftarrow r + \gamma \max_{a'} \hat{Q}(B, a').$$

$$\hat{Q}(A, a_{right}) \leftarrow 0 + 0.9 \max\{0, 0, 0\}$$

$$\hat{Q}(A, a_{right}) \leftarrow 0$$

Ví dụ Gridworld

Bước 3: Giả sử chọn hành động qua phải.



	A	B	C	D	E	G
A	----	0	0	----	----	----
B	0	----	----	0	----	100
C	0	----	----	0	----	----
D	----	0	0	----	0	----
E	----	----	----	0	----	100
G	----	----	----	----	----	0

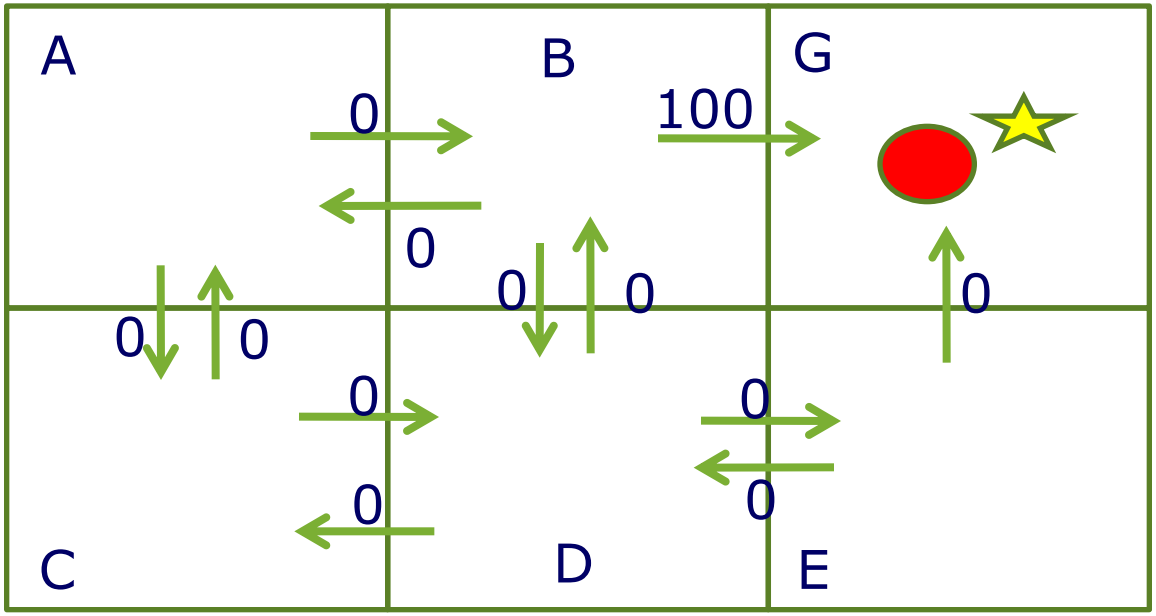
$$\hat{Q}(B, a_{right}) \leftarrow r + \gamma \max_{a'} \hat{Q}(G, a').$$

$$\hat{Q}(B, a_{right}) \leftarrow 100 + 0.9 \max\{0\}$$

$$\hat{Q}(B, a_{right}) \leftarrow 100$$

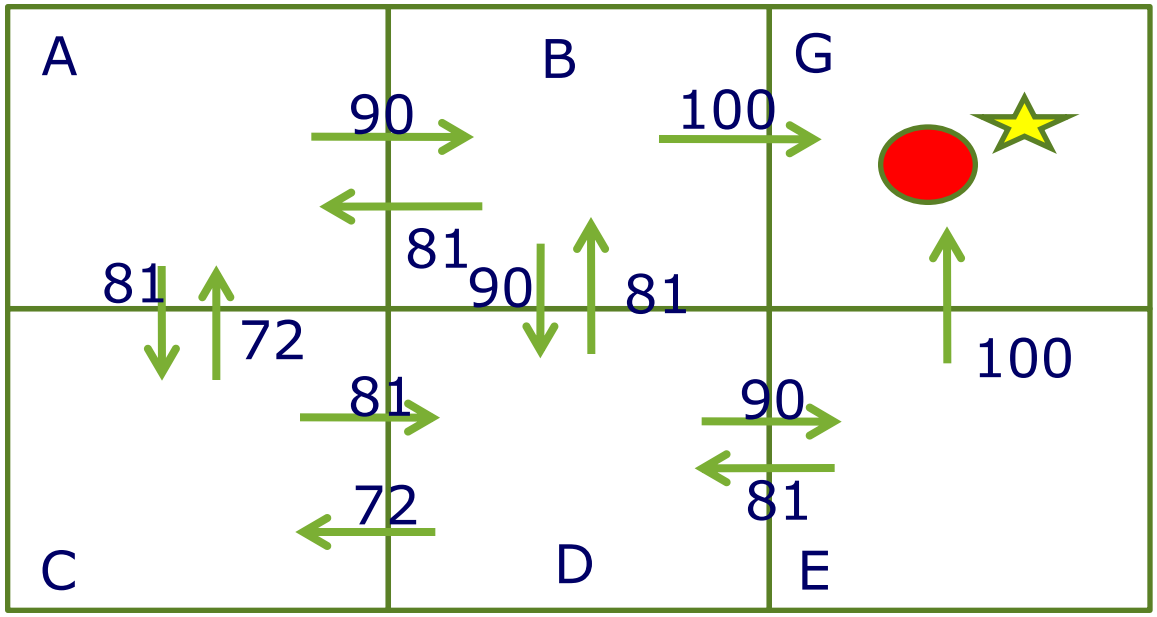
Ví dụ Gridworld

Sau 1 giai đoạn:



Ví dụ Gridworld

Sau k giai đoạn:



1. Thực hiện bao nhiêu giai đoạn?
2. Có thể tìm được lời giải tối ưu?

Chứng minh $\hat{Q}$ hội tụ về Q:

1. Công thức tính $\hat{Q}(s,a)$ trên sử dụng trong tiến trình markov quyết định (MDP).

2. Đối với giá trị điểm thưởng không âm (r >0) thì:

$$(\forall s,a,n)\ \hat{Q}_{n+1}(s,a) \geq \hat{Q}_n(s,a)$$

$$(\forall s,a,n)\ 0 \leq \hat{Q}_n(s,a) \leq Q_n(s,a)$$

Chứng minh $\hat{Q}$ hội tụ về Q:

Trong trường hợp tổng quát:

$\hat{Q}$ được chứng minh là hội tụ về Q, khi:

- Tiến trình là tiến trình markov quyết định. (MDP)
- Điểm thưởng r có giá trị biên là c.

$$(\forall s, a) \mid r(s,a) \mid < c$$

- Mỗi hành động đều có thể thực hiện được cho mọi trạng thái bất kỳ. Do đó, dẫn tới cặp trạng thái-hành động sẽ được đi qua là vô hạn.

Chứng minh $\hat{Q}$ hội tụ về Q:

Cách chứng minh:

Với mọi cặp trạng thái hành động (s,a) đều được viếng thăm ở mọi giai đoạn. Với mỗi giai đoạn thì giá trị sai khác của $\hat{Q}(s,a)$ so với Q(s,a) sẽ giảm đi bởi thừa số γ.

Chứng minh $\hat{Q}$ hội tụ về Q:

Đặt ký hiệu:

+$\hat{Q}_n$ là giá trị của $\hat{Q}$ cập nhật lần thứ n.

+ Δ_n là giá trị độ lỗi lớn nhất có trong $\hat{Q}_n$ so với Q.

$$\Delta_n = \max_{s,a} | \hat{Q}_n(s,a) - Q(s,a) |$$

Chứng minh $\hat{Q}$ hội tụ về Q:

Cho bất kỳ cặp trạng thái-hành động $\hat{Q}_n(s,a)$ cập nhật cho lần thứ n+1, giá trị lỗi trong ước lượng $\hat{Q}_{n+1}(s,a)$ là:

$$
\begin{aligned}
|\hat{Q}_{n+1}(s,a) - Q(s,a)| &= |(r + \gamma \max_{a'} \hat{Q}_n(s',a')) \\
&\qquad - (r + \gamma \max_{a'} Q(s',a'))| \\
&= \gamma |\max_{a'} \hat{Q}_n(s',a') - \max_{a'} Q(s',a')| \\
&\le \gamma \max_{a'} |\hat{Q}_n(s',a') - Q(s',a')| \\
&\le \gamma \max_{s'',a'} |\hat{Q}_n(s'',a') - Q(s'',a')| \\
|\hat{Q}_{n+1}(s,a) - Q(s,a)| &\le \gamma \Delta_n
\end{aligned}
$$

Chứng minh $\hat{Q}$ hội tụ về Q:

Như vậy:

$$|\hat{Q}_{n+1}(s,a) - Q(s,a)| \leq \gamma \Delta_n$$

Nghĩa là giá trị sai khác của $\hat{Q}_{n+1}(s,a)$ với Q(s,a) nhỏ hơn γ lần giá trị sai khác lớn nhất của $\hat{Q}_n(s,a)$ trước khi cập nhật.

Với giá trị $0 < \gamma < 1.0$.

Khởi tạo Δ_0, sau k lần đi qua <s,a>, thì độ sai khác sẽ là $\gamma^k \Delta_0$, và khi $k \rightarrow \infty$, $\Delta_k \rightarrow 0$.

Chiến lược chọn hành động

1. Sử dụng phương pháp tham lam. Cho trạng thái s, chọn $\text{argmax}_a Q(s,a)$.

→ Dẫn tới việc tìm lời giải cục bộ do thám hiểm(exploit) đi tới đích sớm và bỏ qua những hướng đi khác.

→ Ngăn chặn agent đi qua toàn bộ cặp trạng thái-hành động thường xuyên.

Chiến lược chọn hành động

2. Sử dụng xác suất:

   ε-greedy:

   + Với xác suất nhỏ hơn ε, chọn hành động một cách ngẫu nhiên giữa các hành động có thể, **explore**.

   + Ngược lại với xác suất lớn hơn ε, chọn hành động tốt nhất, **exploit**.

Chiến lược chọn hành động

Xác suất để chọn hành động a cho trạng thái s:

$$P(a|s) = \frac{\exp(\hat{Q}(s,a))}{\sum_{b=1}^{\mathcal{A}} \exp(\hat{Q}(s,b))}$$

+ exp là hàm chuyển đổi giá trị thành xác suất để cho xác suất chọn các hành động a ∈ A và trạng thái s ∈ S lớn hơn 0.

Ngoài ra còn có:

$$P(a|s) = \frac{\exp[Q(s,a)/T]}{\sum_{b=1}^{\mathcal{A}} \exp[Q(s,b)/T]}$$

T temperature, T lớn → exploration; T nhỏ → exploitation.

Chiến lược cập nhật

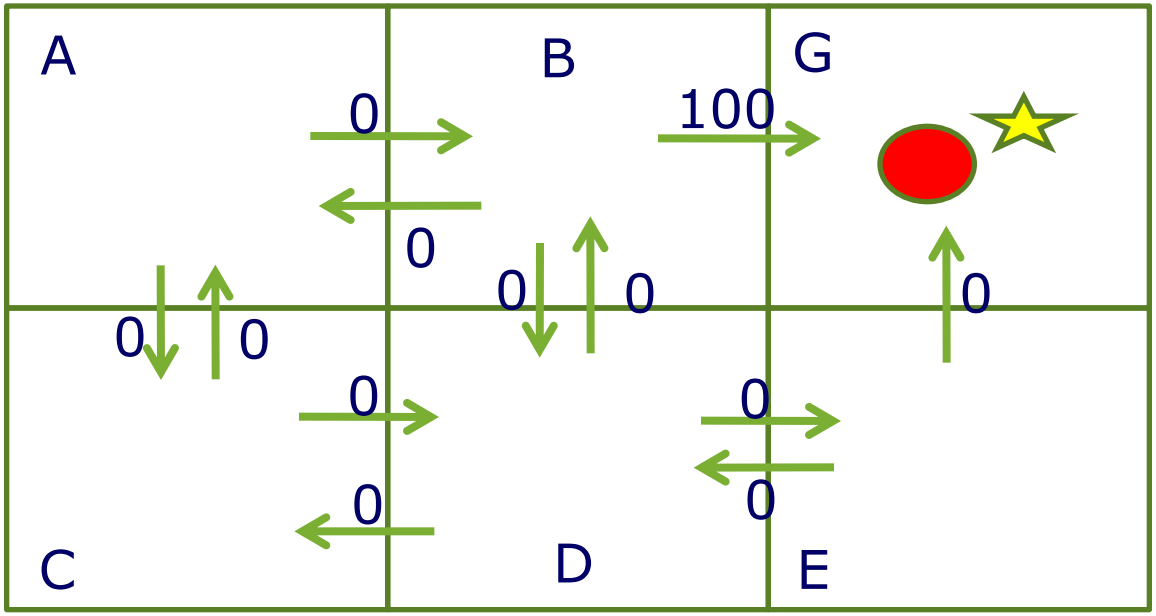
Không cần thứ tự cho (s,a) để giá trị hội tụ. Tuy nhiên, việc cập nhật này không hiệu quả.

1. Thực hiện cập nhật theo thứ tự ngược lại, khi kết thúc một giai đoạn (hay agent đã đến đích).

2. Lưu trữ dãy chuyển trạng thái – hành động.

Chiến lược cập nhật 1:

Ví dụ: C→A→B→G. Cập nhật theo chiều ngược lại



	A	B	C	D	E	G
A	----	0	0	----	----	----
B	0	----	----	0	----	100
C	0	----	----	0	----	----
D	----	0	0	----	0	----
E	----	----	----	0	----	100
G	----	----	----	----	----	0

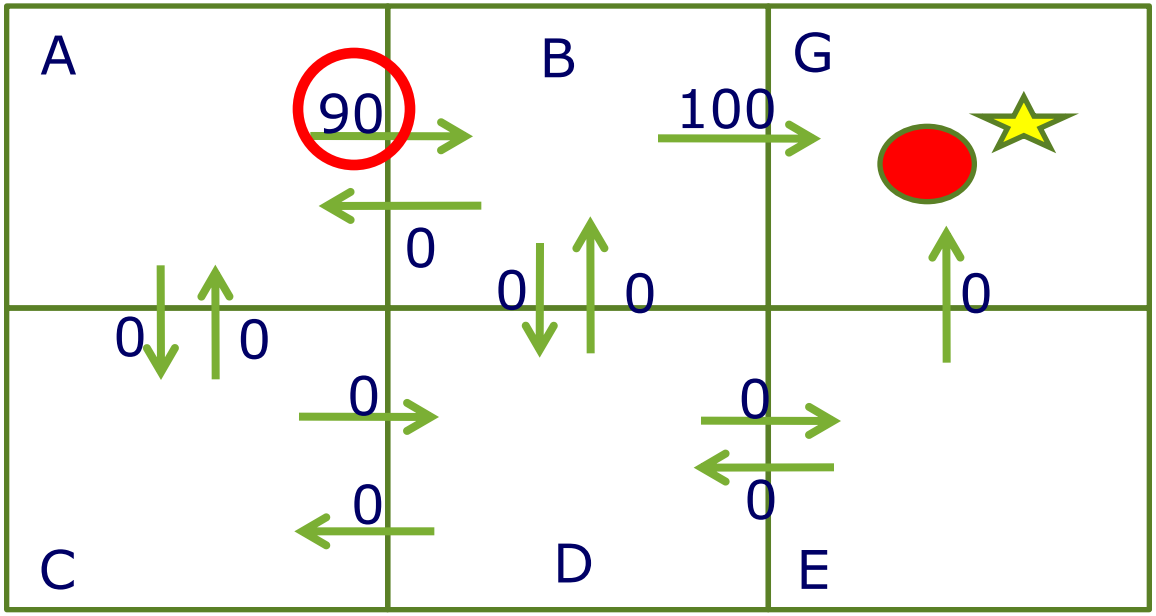
$$\hat{Q}(B, a_{right}) \leftarrow r + \gamma \max_{a'} \hat{Q}(G, a').$$

$$\hat{Q}(B, a_{right}) \leftarrow 100 + 0.9 \max\{0,0,0\}$$

$$\hat{Q}(B, a_{right}) \leftarrow 100$$

Chiến lược cập nhật 1:

Ví dụ: C→A→B→G. Cập nhật theo chiều ngược lại



	A	B	C	D	E	G
A	----	0	0	----	----	----
B	0	----	----	0	----	100
C	0	----	----	0	----	----
D	----	0	0	----	0	----
E	----	----	----	0	----	100
G	----	----	----	----	----	0

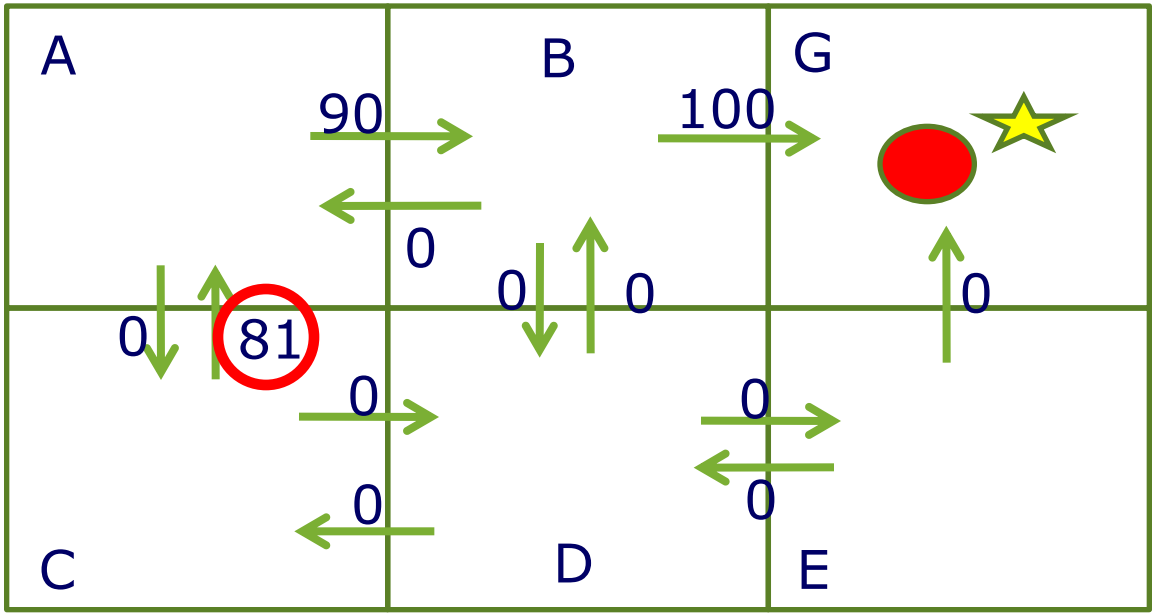
$$\hat{Q}(A, a_{right}) \leftarrow r + \gamma \max_{a'} \hat{Q}(B, a').$$

$$\hat{Q}(A, a_{right}) \leftarrow 0 + 0.9 \max\{100, 0, 0\}$$

$$\hat{Q}(A, a_{right}) \leftarrow 90$$

Chiến lược cập nhật 1:

Ví dụ: C→A→B→G. Cập nhật theo chiều ngược lại



	A	B	C	D	E	G
A	----	0	0	----	----	----
B	0	----	----	0	----	100
C	0	----	----	0	----	----
D	----	0	0	----	0	----
E	----	----	----	0	----	100
G	----	----	----	----	----	0

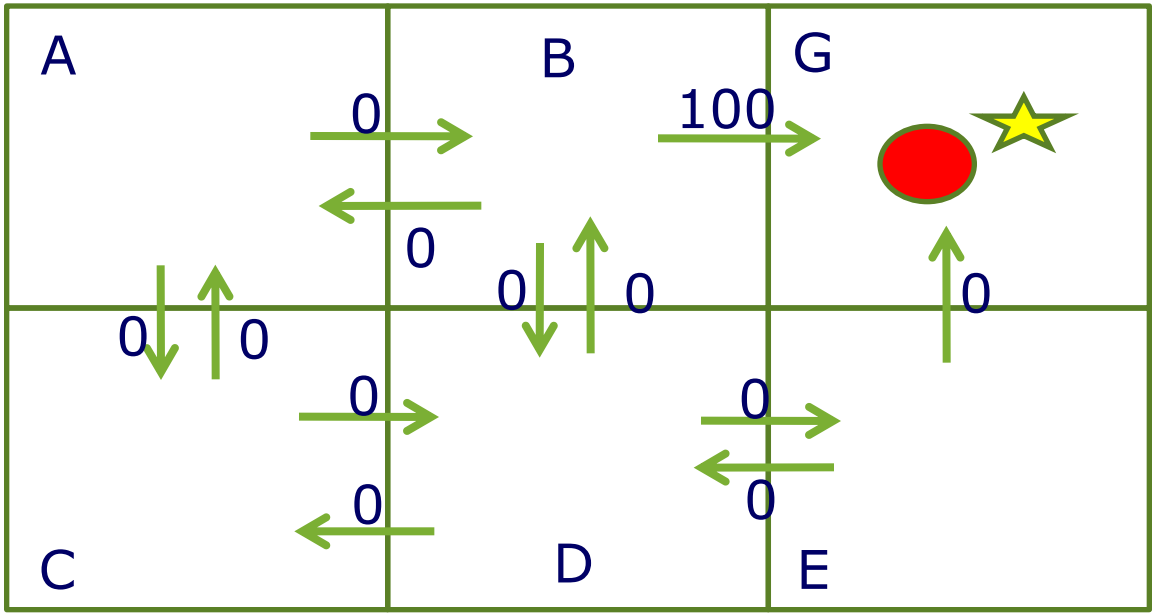
$$\hat{Q}(C, a_{up}) \leftarrow r + \gamma \max_{a'} \hat{Q}(A, a').$$

$$\hat{Q}(C, a_{up}) \leftarrow 0 + 0.9 \max\{90, 0, 0\}$$

$$\hat{Q}(C, a_{up}) \leftarrow 81$$

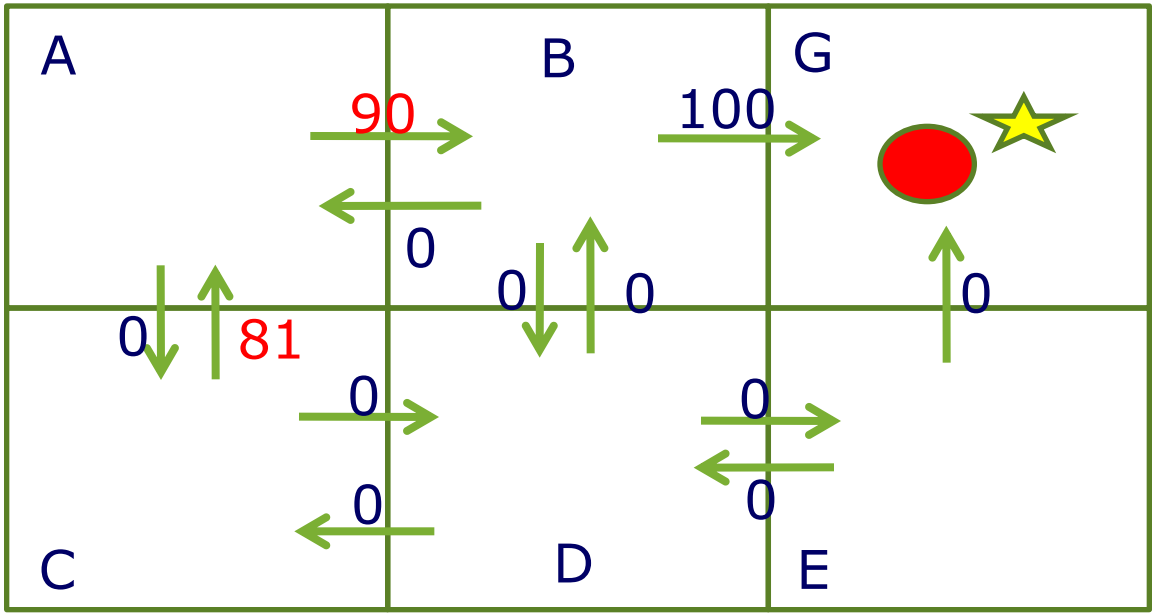
Chiến lược cập nhật 2:

Ví dụ: huấn luyện C→A→B→G lần 1.



Chiến lược cập nhật 2:

Ví dụ: huấn luyện C→A→B→G lần n.



Kết luận

- Học tăng cường là phương pháp tìm ra một chính sách nhằm đạt được điểm thưởng tối đa từ bất kỳ trạng thái bắt đầu nào.
- Khi học tăng cường ta không cần biết trước kiến thức về vấn đề cần học. Khác với "Dynamic Programming" cần phải có kiến thức trước.
- Cách đánh giá việc học thông qua điểm thưởng cho hành động tương ứng.
- Ứng dụng quyết định Markov trong quá trình học để tích lũy thông tin ra quyết định.
- Luôn hội tụ cho cả MDP có tính quyết định và không có tính quyết định