HỆ THỐNG TÌM KIẾM TRONG MÔI TRƯỜNG ĐỐI KHÁNG (ADVERSARIAL)

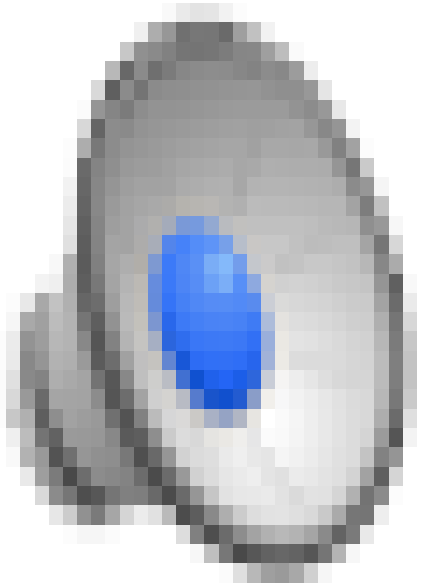
Tổng thể

- Dẫn nhập
- Bài toán chơi game đối kháng
- Tìm lời giải cho bài toán chơi game đối kháng: thuật toán Minimax

Dẫn nhập

- Ở các buổi trước, trong game Pacman, hầu như ta chỉ xét ngữ cảnh: Pacman chỉ có một mình trong mê cung
- Buổi này, ta sẽ xét ngữ cảnh khó hơn - ngữ cảnh có đối kháng: trong mê cung còn có ma → việc tìm kiếm sẽ khó hơn

Xem video: Pacman vs ma



Một số game đối kháng khác

- Tic-Tac-Toe
  - Hai người chơi: một người đánh X, một người đánh O
  - Mỗi người lần lượt đánh X (hoặc O) vào ô trống trên bàn cờ 3×3
  - Trò chơi kết thúc khi:
    - Có một người tạo được một hàng ngang (hoặc dọc, hoặc chéo) chỉ chứa toàn X (hoặc O)
    - Hoặc không còn chỗ nào để đánh nữa

- Trình chơi Tic-Tac-Toe của AI hiện nay?
  - Vô đối ☺

Nguồn ảnh: Wikipedia

Một số game đối kháng khác

- Checker
  - Hai người chơi
  - Mỗi người lần lượt di chuyển một quân cờ của mình theo đường chéo về phía trước một ô; nếu quân đến được hàng cuối thì sẽ được phong vua (xem thêm luật chơi ở đây)
  - Trò chơi kết thúc khi một trong hai người không còn quân để đi

- Trình chơi Checker của AI hiện nay?
  - Vô đối ☺

Nguồn ảnh: Wikipedia

Một số game đối kháng khác

- Cờ vua: ...
- Trình chơi cờ vua của AI hiện nay?
  - Hơn chuyên gia

Nguồn ảnh: Wikipedia

Bài toán chơi game đối kháng

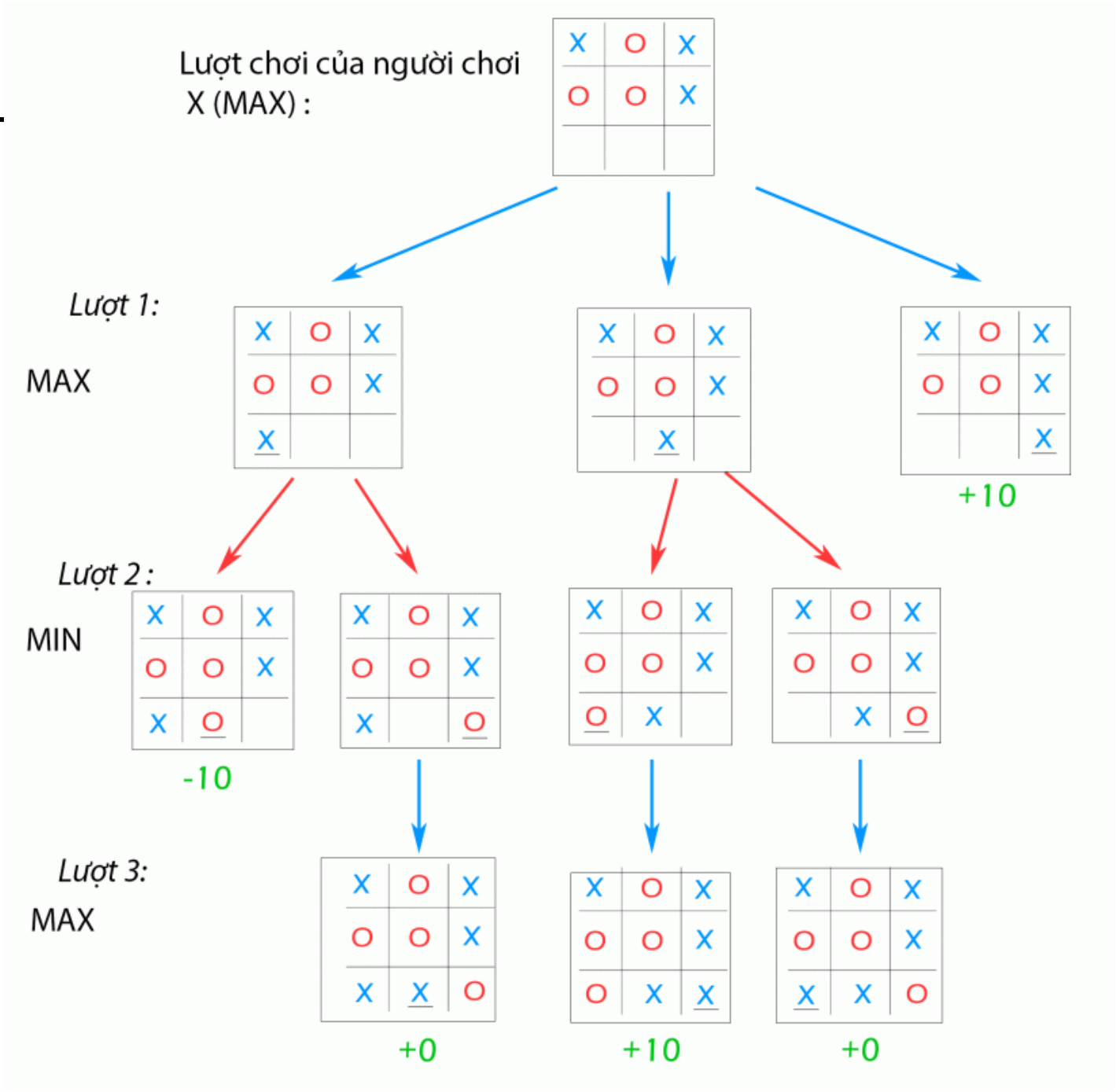
- Cho các thành phần của game:
  - Tập trạng thái: S ($s_0 \in S$ là trạng thái bắt đầu)
  - Tập người chơi (lượt chơi luân phiên nhau): $P = \{1, \dots, N\}$
  - Tập hành động: A (có thể tập A sẽ phụ thuộc vào mỗi người chơi cũng như là mỗi trạng thái)
  - Hàm chuyển trạng thái (transition function; tương tự hàm successor): $S \times A \to S$
  - Hàm kiểm tra trạng thái kết thúc (terminal test): $S \to \{0, 1\}$
  - Hàm cho biết lợi ích (utility) của trạng thái kết thúc đối với mỗi người chơi: $S \times P \to \mathbb{R}$
- Với một người chơi, ta muốn tìm lời giải là một chính sách chơi (policy): $S \to A$

Trước mắt, ta sẽ xét $N = 2$ người chơi …

… và mối quan hệ giữa 2 người chơi là thuần đối kháng (zero sum): với trạng thái kết thúc, lợi ích của người chơi 1 + lợi ích của người chơi 2 = 0

Tổng thể

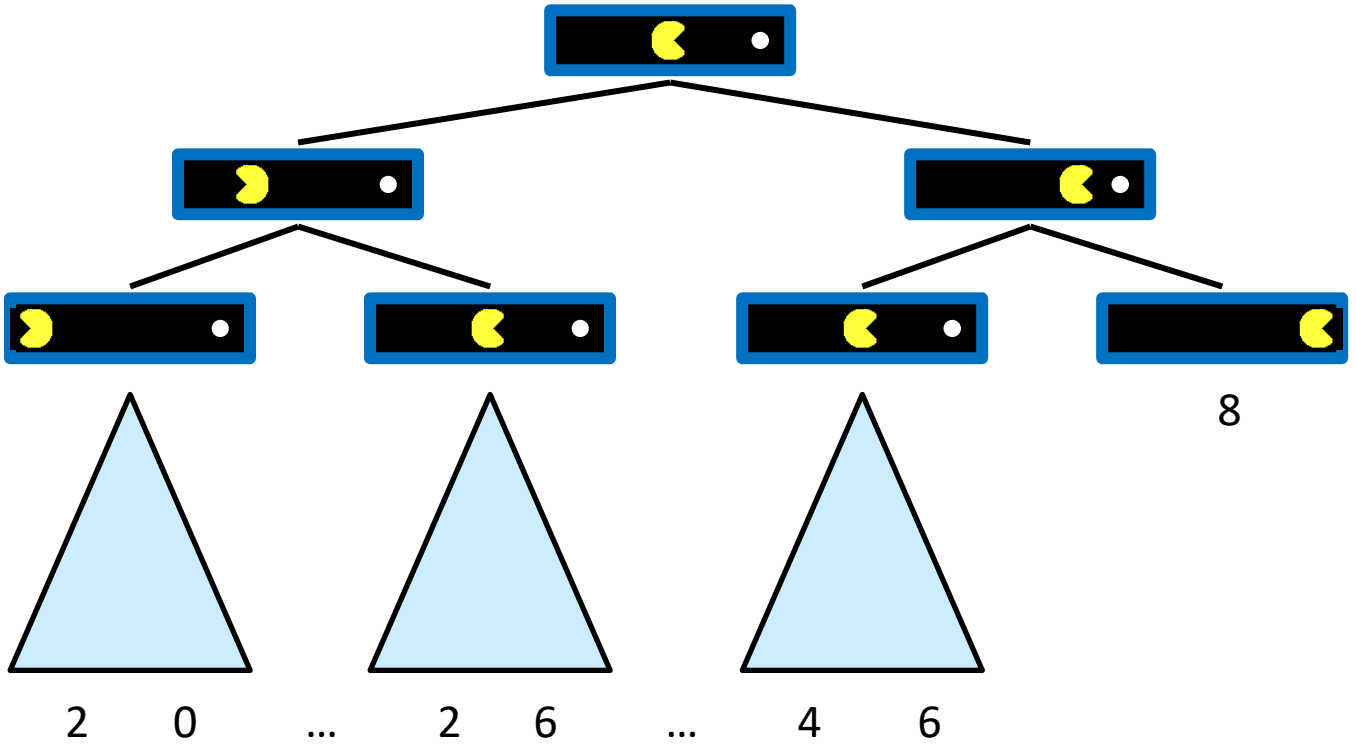
- Dẫn nhập
- Bài toán chơi game đối kháng
- Tìm lời giải cho bài toán chơi game đối kháng: thuật toán Minimax


Lượt chơi của người chơi
X (MAX) :
Lượt 1:
MAX
+10
Lượt 2 :
MIN
-10
Lượt 3:
MAX
+0
+10
+0

Cây khi chỉ có một người chơi

Làm sao để biết từ trạng thái hiện tại ta nên thực hiện hành động nào?

- Nhìn về tương lai phía trước: hình dung các trường hợp có thể xảy ra khi ta thực hiện các hành động
- Nhìn bao xa? Cho tới khi game kết thúc, vì khi đó ta mới biết được lợi-ích/điểm-số-cuối-cùng (utility)



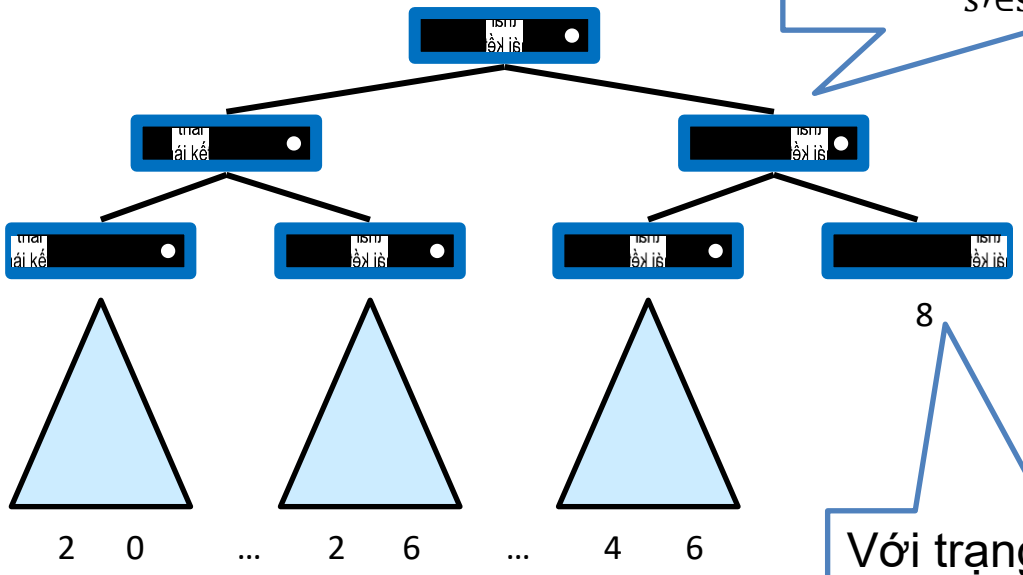
Nguồn ảnh: UC Berkely. CS188 Intro to AI

Giá trị (value) của một trạng thái

Giá trị của một trạng thái: là lợi-ích/điểm-số-cuối-cùng (utility) tốt nhất có thể đạt được từ trạng thái đó

Từ giá trị trạng thái, ta có thể có được chính sách chơi:
Vd, ở trạng thái hiện tại, ta có thể thực hiện 2 hành động: hành động L đi tới trạng thái có giá trị là 6, hành động R đi tới trạng thái có giá trị là 8. Nên chọn hành động nào?

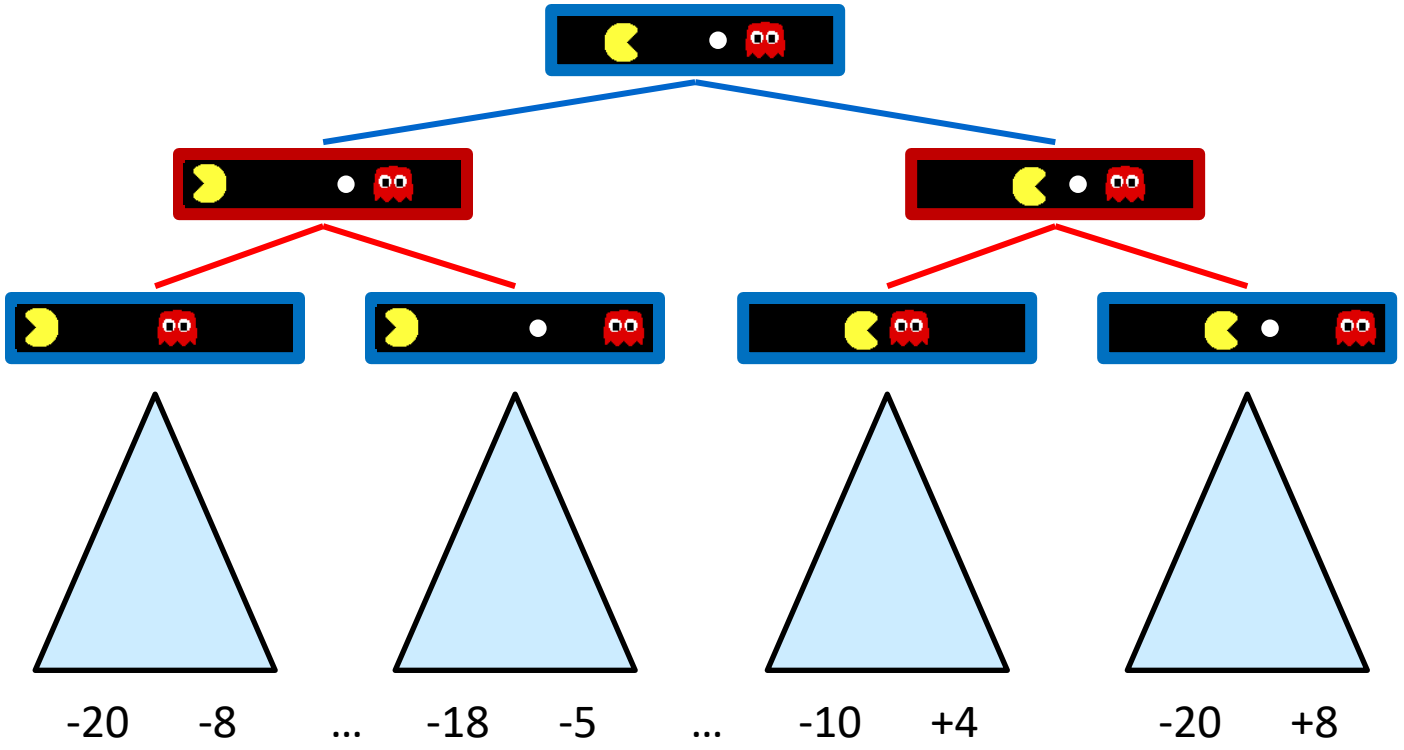
Với trạng thái không phải là trạng thái kết thúc:
$V(s) = \max_{s' \in \text{successors}(s)} V(s')$



Với trạng thái kết thúc:
$V(s) = \text{known}$

Nguồn ảnh: UC Berkely. CS188 Intro to AI

Cây khi có hai người chơi đối kháng



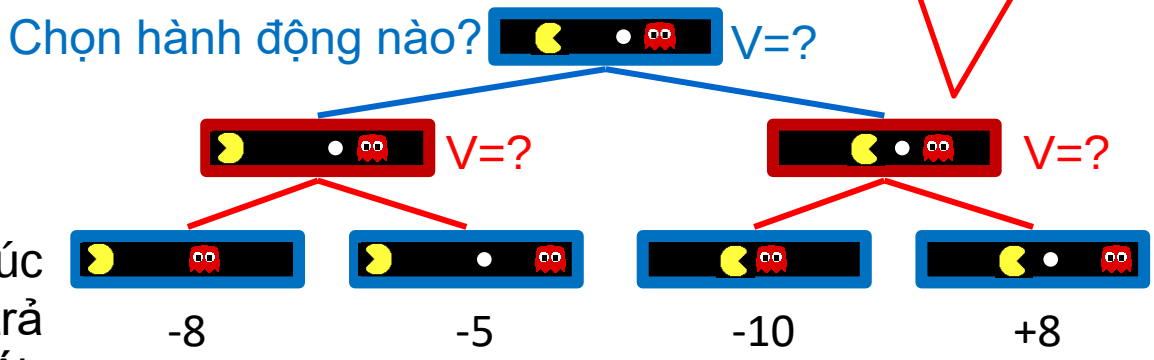
Nguồn ảnh: UC Berkely. CS188 Intro to AI

Giá trị minimax của một trạng thái

Giá trị minimax của một trạng thái: là lợi-ích/điểm-số-cuối-cùng (utility) tốt nhất có thể đạt được từ trạng thái đó với giả định đối thủ chơi tối ưu

Với trạng thái dưới sự kiểm soát của Pacman:
$V(s) = \max_{s' \in \text{successors}(s)} V(s')$

Với trạng thái dưới sự kiểm soát của ma:
$V(s) = \min_{s' \in \text{successors}(s)} V(s')$



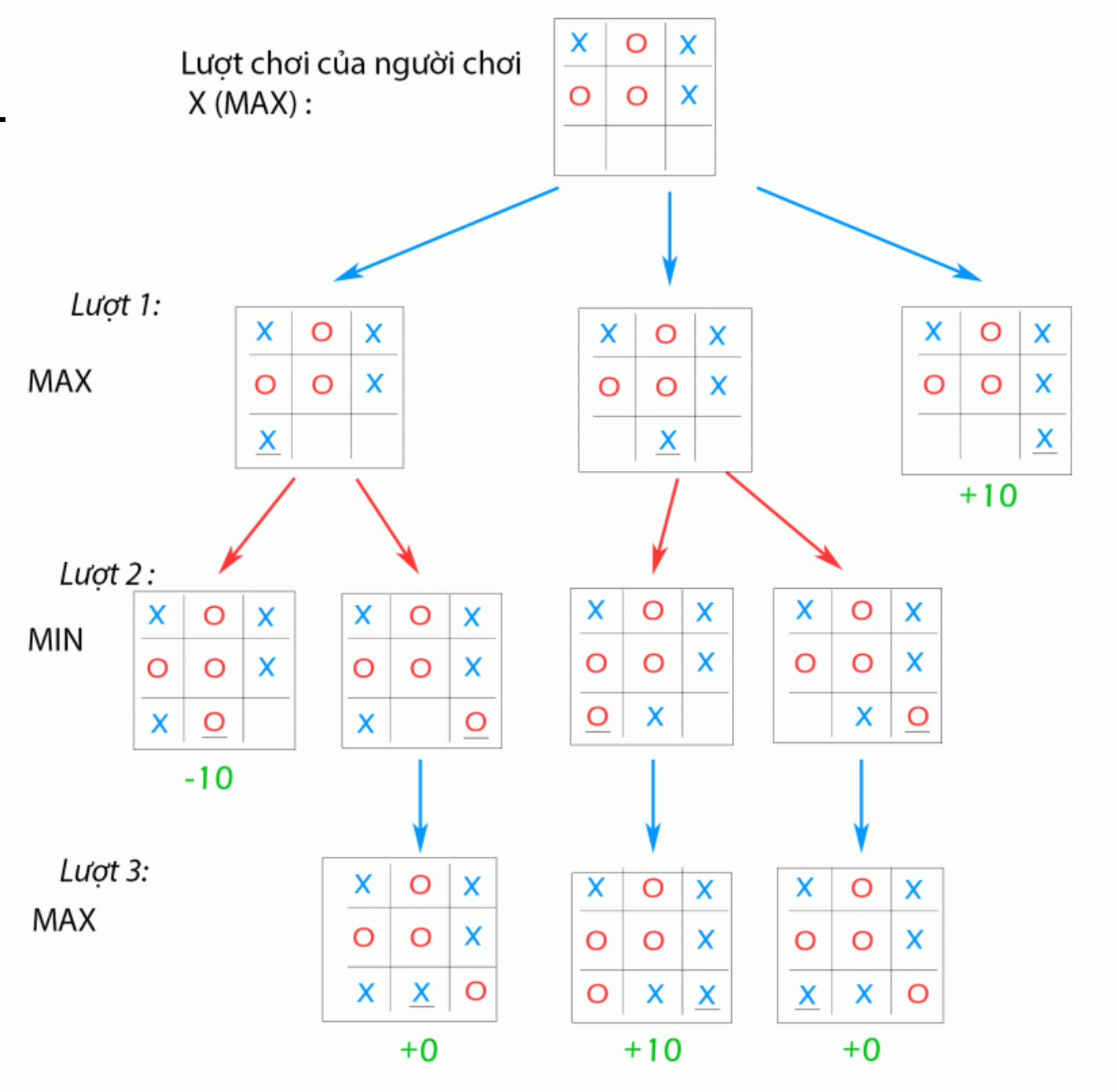
Giả sử game sẽ kết thúc ở thời điểm này, và trả về lợi-ích/điểm-số-cuối-cùng (utility)

Với trạng thái kết thúc:
$V(s) = \text{known}$

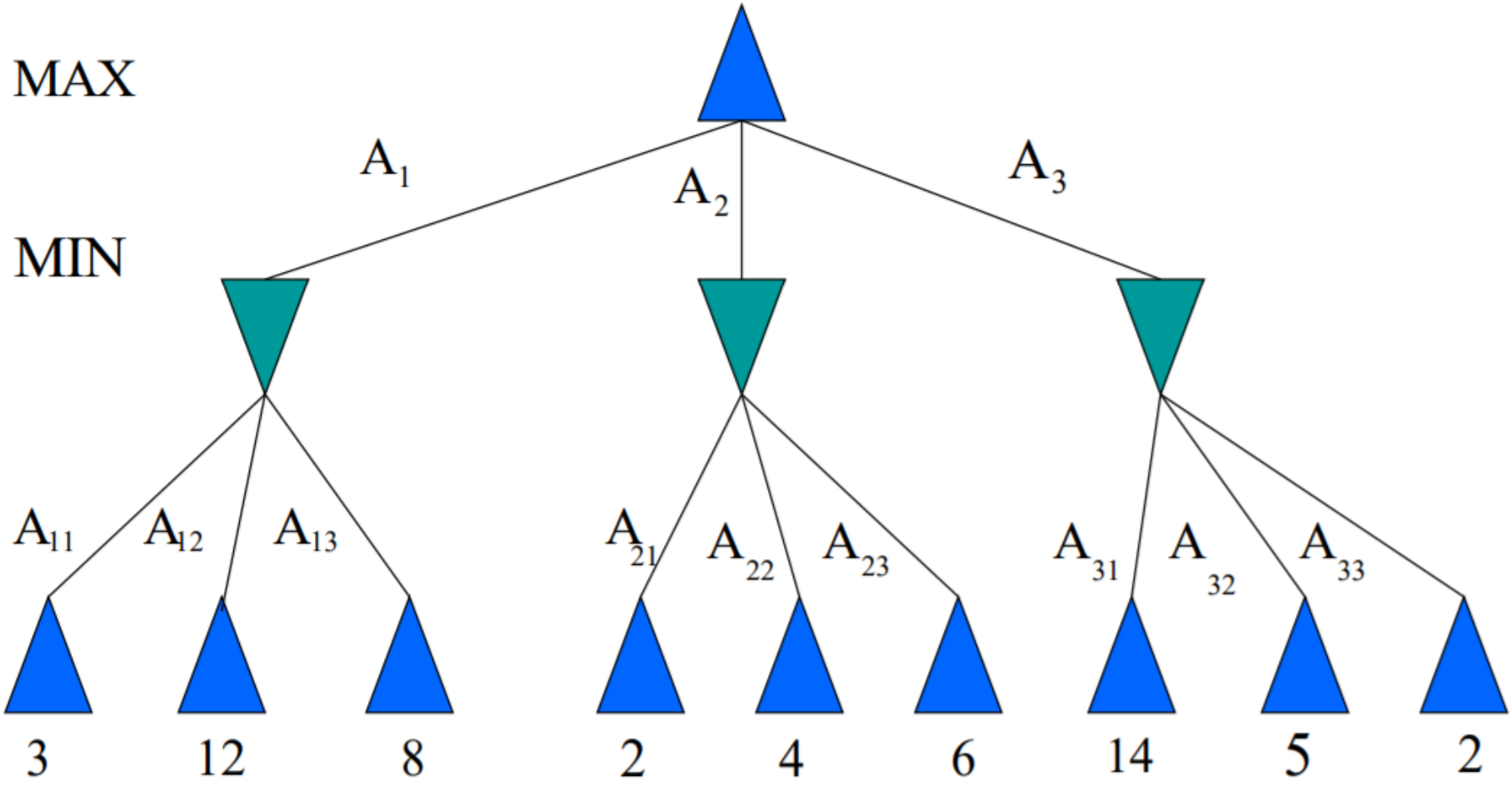
Nguồn ảnh: UC Berkely. CS188 Intro to AI

Thuật toán Minimax

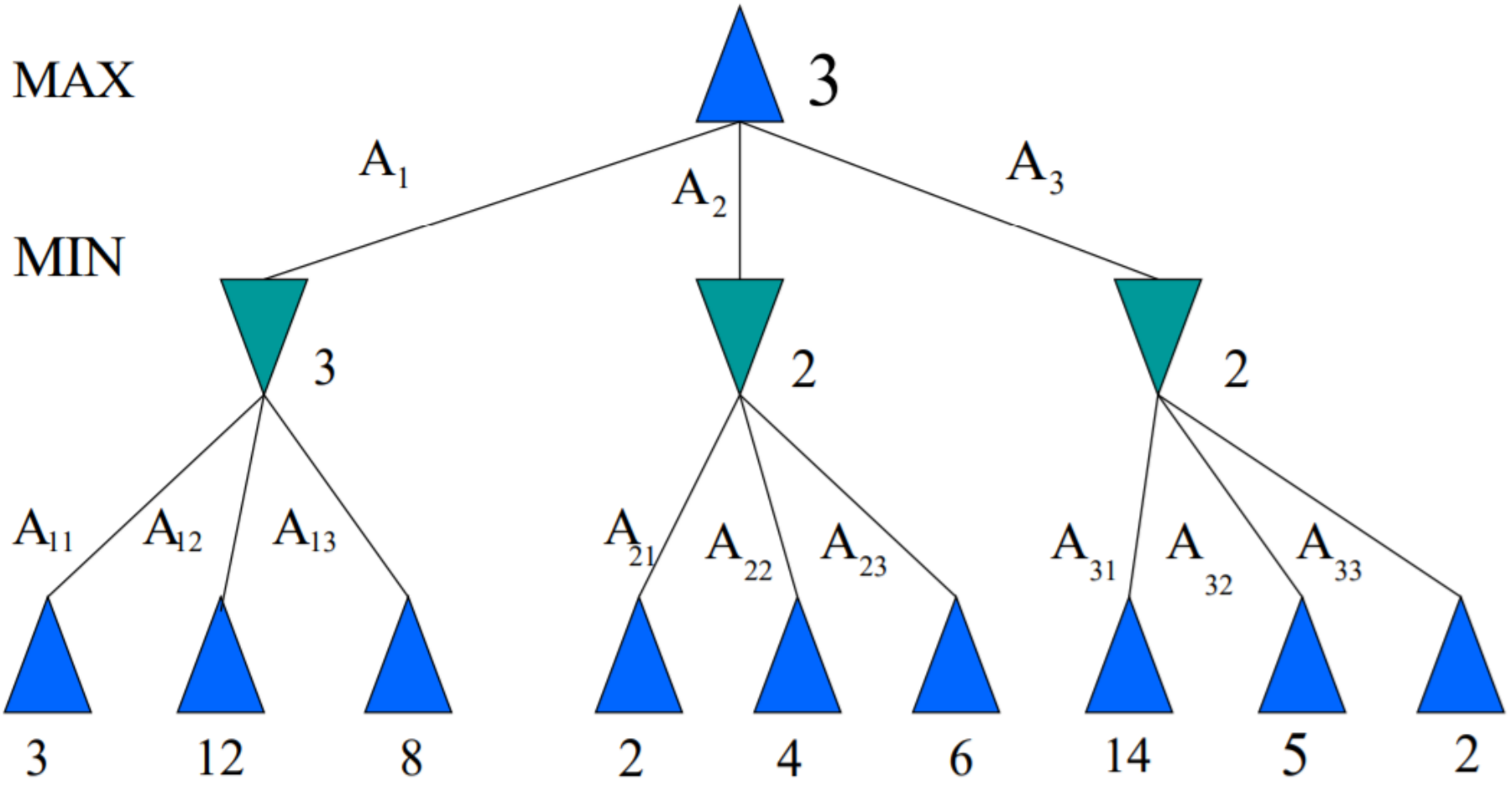
- - Mở rộng toàn bộ cây.
- - Xác định giá trị của các node lá.
- - Đối với các node không phải lá:
  - Nếu đang là bước minimize, node hiện tại sẽ có giá trị là giá trị nhỏ nhất trong các node con của nó.
  - Nếu đang là bước maximize, node hiện tại sẽ có giá trị là giá trị lớn nhất trong các node con của nó


Lượt chơi của người chơi
X (MAX) :
Lượt 1:
MAX
+10
Lượt 2 :
MIN
-10
Lượt 3:
MAX
+0
+10
+0

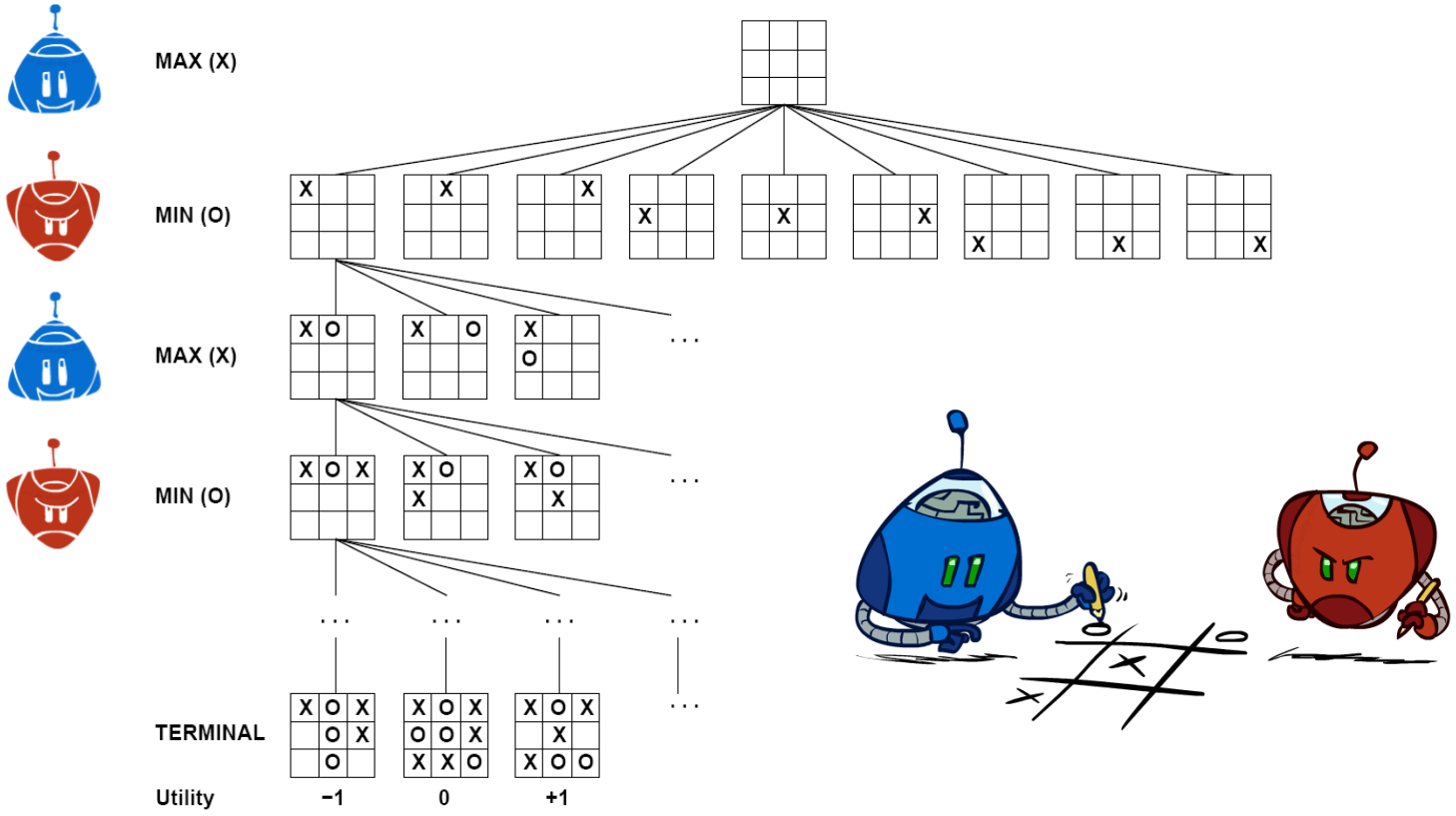
Ví dụ


MAX
MIN
A1
A2
A3
A11
A12
A13
A21
A22
A23
A31
A32
A33
3
12
8
2
4
6
14
5
2

Ví dụ


MAX
MIN
3
A1
A2
A3
3
2
2
A11
A12
A13
A21
A22
A23
A31
A32
A33
3
12
8
2
4
6
14
5
2

Cây trong game Tic-Tac-Toe

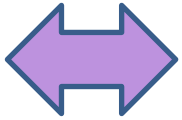


Nguồn ảnh: UC Berkely. CS188 Intro to AI

Thuật toán minimax

Để biết ta (giả sử là người chơi max) nên thực hiện hành động nào từ trạng thái hiện tại, ta có thể tính giá trị của trạng thái hiện tại bằng hàm max-value ở dưới …

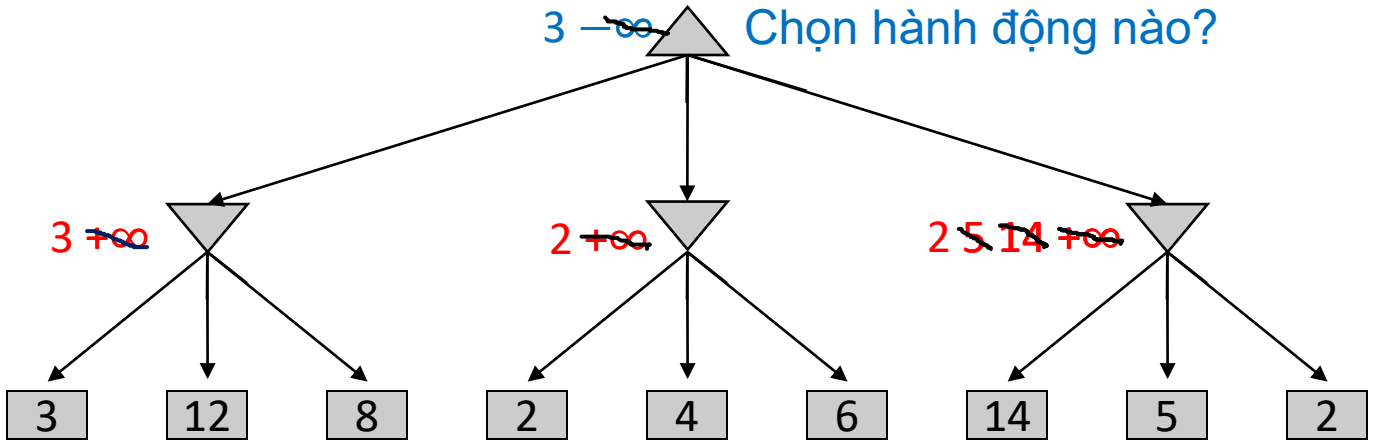
```
def max-value(state):
    if the state is a terminal state:
        return the state's utility
    initialize v = -∞
    for each successor of state:
        v = max(v, min-value(successor))
    return v
```



```
def min-value(state):
    if the state is a terminal state:
        return the state's utility
    initialize v = +∞
    for each successor of state:
        v = min(v, max-value(successor))
    return v
```

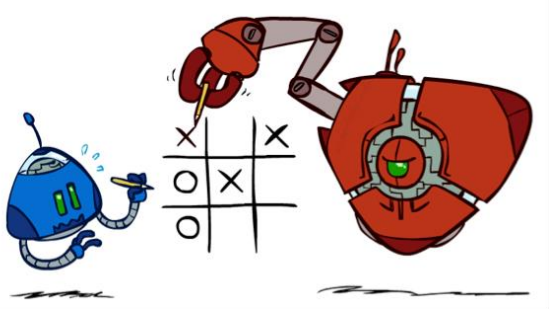
Nguồn: UC Berkely. CS188 Intro to AI

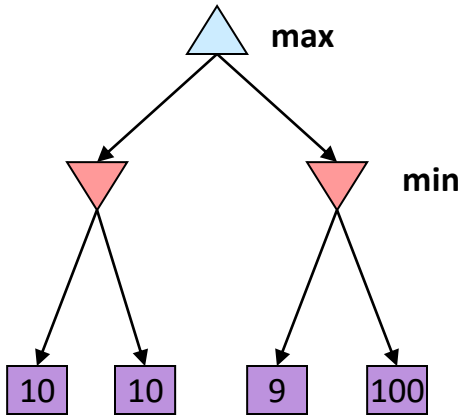
Thuật toán minimax

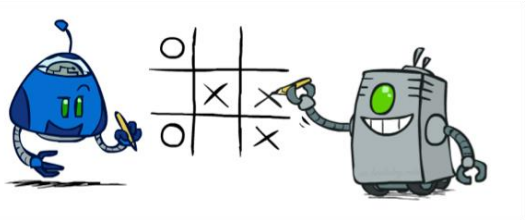


Nguồn: UC Berkely. CS188 Intro to AI

Bình luận về thuật toán minimax



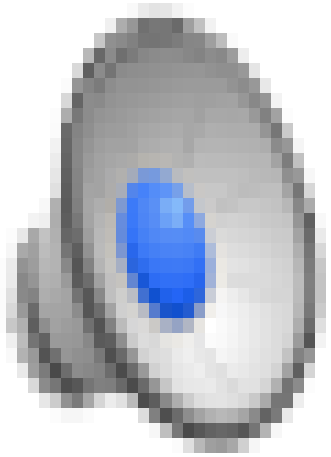




Lời giải sẽ tối ưu nếu đối thủ cũng chơi tối ưu
Nếu đối thủ không chơi tối ưu thì sao?

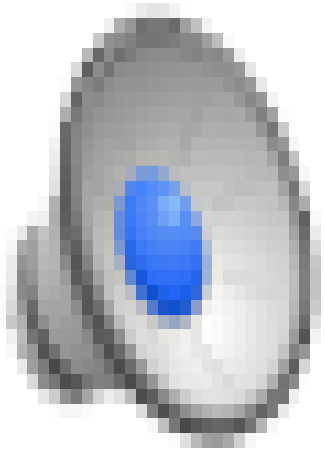
Nguồn ảnh: UC Berkely. CS188 Intro to AI

Xem video: ma chơi tối ưu



Nguồn: UC Berkely. CS188 Intro to AI

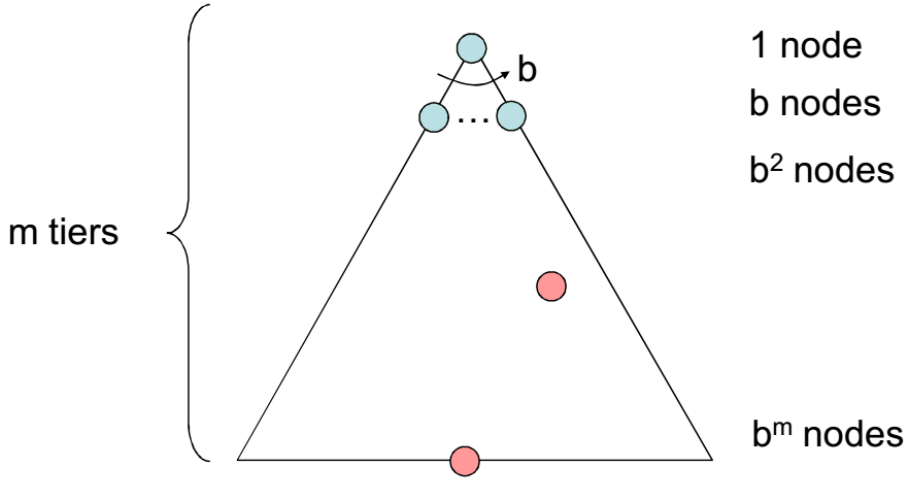
Xem video: ma chơi không tối ưu



Nguồn: UC Berkely. CS188 Intro to AI

Bình luận về thuật toán minimax

- Số node (trạng thái) phải duyệt?
  - Nguyên cả cây!
  - Giả sử cây có hệ số phân nhánh b và có độ sâu m → Số node của cây: $1 + b + b^2 + \ldots + b^m = O(b^m)$
  - Cờ vua có $b \approx 35$, $m \approx 100$ → Minimax sẽ chạy mãi mãi …
- Ta cần cải tiến thuật toán minimax để không phải duyệt toàn bộ cây



Nguồn ảnh: UC Berkely. CS188 Intro to AI