Chương 4: Tìm kiếm đối kháng – trò chơi

Giảng viên: Nguyễn Văn Hòa

Khoa CNTT - ĐH An Giang

Nội dung

- Trò chơi
- Trò chơi đối kháng và tìm kiếm
- Thuật toán MINIMAX
- Cắt tỉa α-β

Trò chơi

- Trò chơi một trong những đặc tính được xem là "thông minh" của con người
- Trò chơi là phiên bản "F1" của AI
- Đã đạt được những thành tựu đáng kể
- Ở đây ta chỉ xem xét các dạng trò chơi trí tuệ, đối kháng (board game)

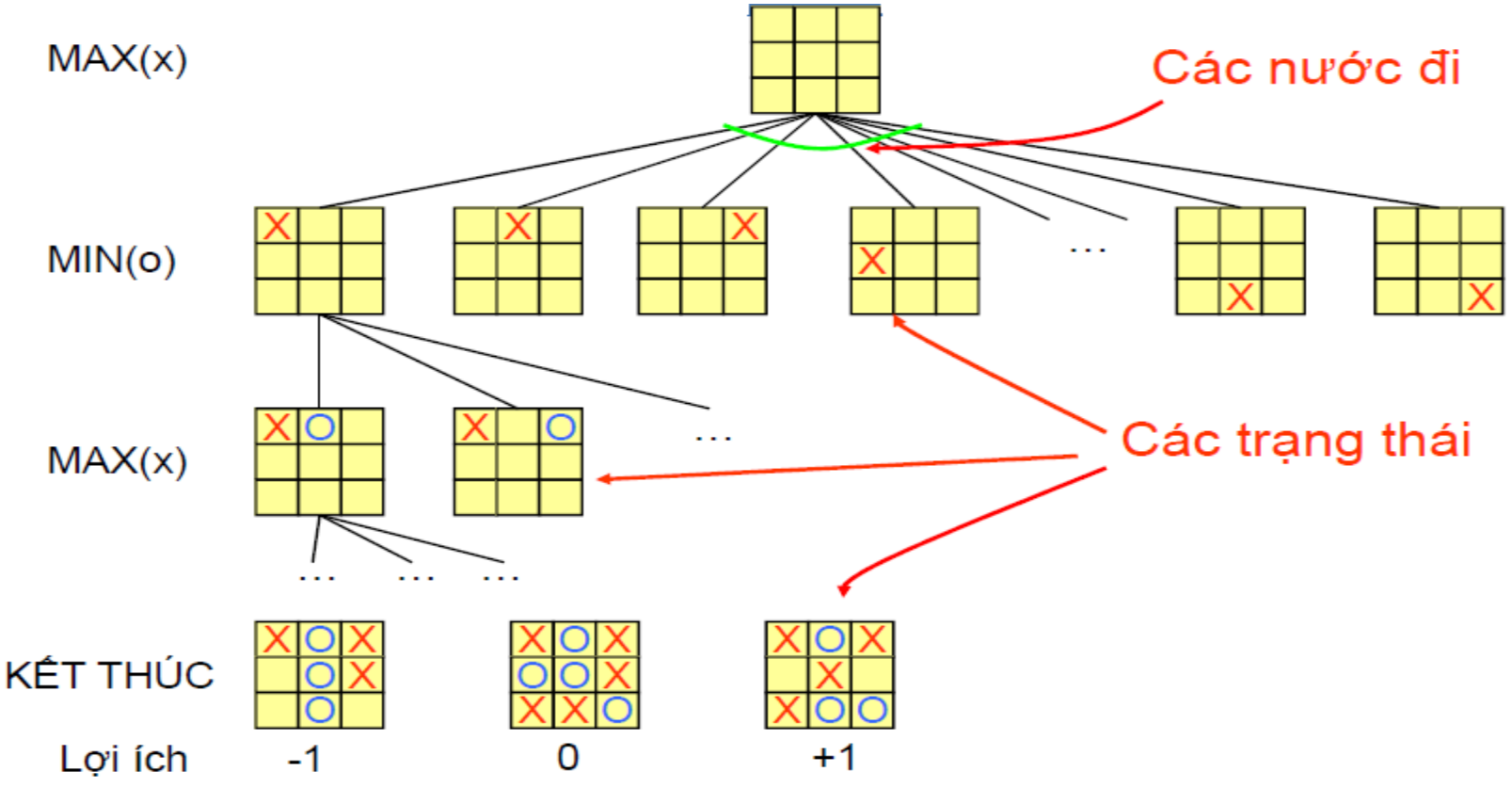
Trò chơi

- Cờ vua
  - 1997, DeepBlue đánh bại Gary Kasparov trong trận đấu 6 ván
  - Bí quyết
    - Tìm kiếm vét cạn với độ sâu cao nhất có thể
    - Tính được $2x10^8$ nước đi trong 1 giây so với 2 nước của Kasparov
    - 99.99% nước đi được xem là "ngu ngốc"
    - Hàm ước lượng giá (heuristic) rất phức tạp

Trò chơi đối kháng và tìm kiếm

- Các thành phần
  - Tập trạng thái: tập "cấu hình" hợp lệ của trò chơi
  - Trạng thái bắt đầu, trạng thái kết thúc
  - Hàm **succs:** các nước đi hợp lệ
  - Hàm lợi ích: đánh giá trạng thái kết thúc
- Hai người chơi: MAX vs. MIN
- Không tìm đường đi mà tìm nước đi "tốt nhất"
- Nước đi của MAX phụ thuộc vào nước đi của MIN và ngược lại

Ví dụ cây tìm kiếm trò chơi: TicTacToe



Chiến lược tìm kiếm đối kháng

- Đặc điểm
  - Hai người thay phiên đi (xen kẽ)
  - Hai người biết thông tin đầy đủ về nhau
  - Mỗi người tìm kiếm nước đi tốt nhất
  - Nước đi tốt nhất là nước đi dẫn đến phần thắng.
  - Biểu diễn KGTT bằng: cây trò chơi.
- Thuật toán tiêu biểu: MINIMAX

Thuật toán MINIMAX

- Những người chơi là tối ưu
  - MAX tối đa hóa hàm lợi ích
  - MIN tối thiểu hóa hàm lợi ích
  - Chiến lược của MAX phụ thuộc vào chiến lược của MIN ở bước sau
- Giá trị MINIMAX-VALUE: tiện ích ở trạng thái kết thúc tương ứng với đường đi, với giả sử hai người chơi luôn tối ưu

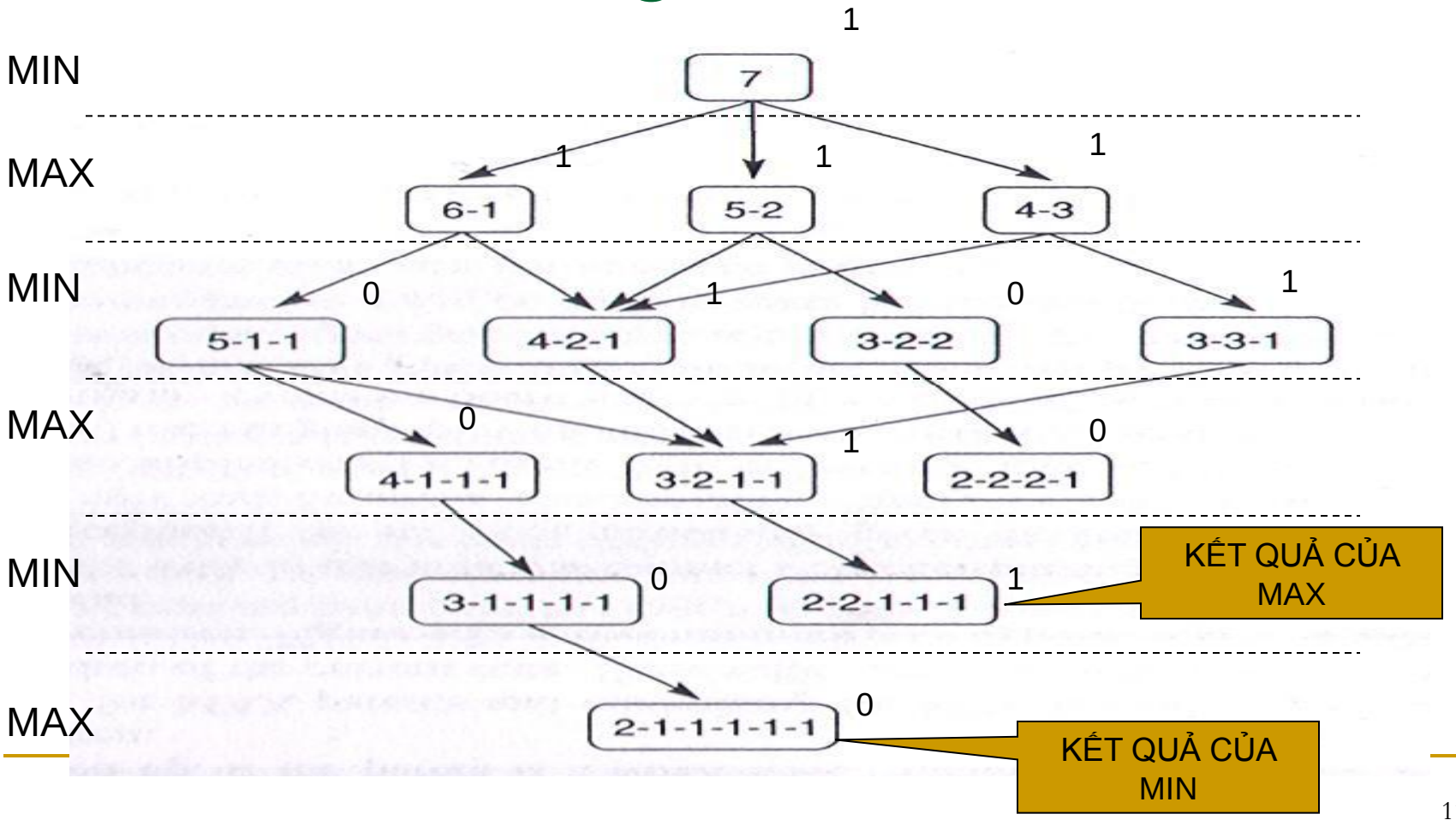
Giá trị MINIMAX

- MINIMAX-VALUE(n)=
  - Utility(n) nếu n là trạng thái kết thuc
  - max{MINIMAX-VALUE(s)) | s’ ∈ succs (s)}
    - Nếu n là nút max
  - min{MINIMAX-VALUE(s)) | s’ ∈ succs (s)}
    - Nếu n là nút min

Ví dụ trò chơi đối kháng: Nim

- Trò chơi Nim
  - Có n (n>2) đồng xu
  - Mỗi nước đi, người chơi chia các đồng xu này thành hai đống nhỏ có số lượng mỗi đống khác nhau
  - Người thua sẽ là người cuối cùng không chia được theo yêu cầu của bài toán
- Phân tích
  - Tính toán phản ứng của đối thủ là khó khăn chủ yếu
  - Cách giải quyết là giả thiết đối thủ cũng sử dụng kiến thức về không gian trạng thái

Trò chơi Nim: giá trị tại các nút



Trò chơi Nim

- Hai đấu thủ: MIN và MAX
- Trong đó MAX luôn tìm cách tối đa ưu thế của mình và MIN tìm mọi cách để đưa MAX vào thế khó khăn nhất
- Mỗi mức trên KGTT ứng với một đấu thủ
- Để chỉ dẫn được cách đi, chúng ta sẽ gán cho các nút lá là 1 nếu MAX thắng, là 0 nếu MIN thắng
- Gán giá trị cho các nút: truyền ngược các trị từ các nút lá về gốc theo qui tắc
  - Nếu đỉnh ở mức MAX, gán trị cho đỉnh này bằng giá trị lớn nhất trong các giá trị của các con của nó
  - Nếu đỉnh ở mức MIN, gán trị cho đỉnh này bằng giá trị bé nhất trong các trị của các con của nó

Giải thuật MINIMAX

```
function MINIMAX-DECISION (state) returns an action
    v ← MAX-VALUE (state)
    return the action in succs(state) with the value v
```

```
function MAX-VALUE (state) returns an utility value
    if TERMINAL-TEST (state) then return the UTILITY (state)
    v ← -∞
    for each s in succs(state) do
        v ← MAX(v, MIN-VALUE(s))
    return v
```

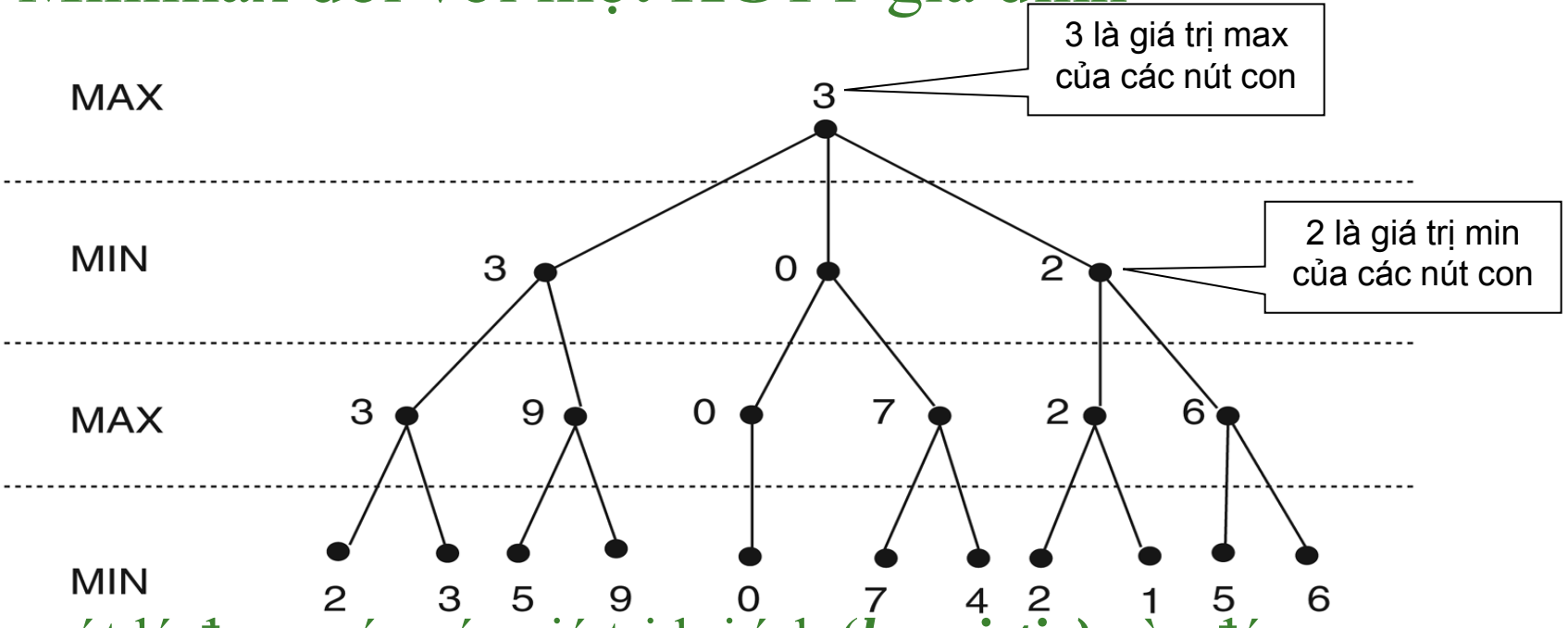
```
function MIN-VALUE (state) returns an utility value
    if TERMINAL-TEST (state) then return the UTILITY (state)
    v ← +∞
    for each s in succs(state) do
        v ← MIN(v, MAX-VALUE(s))
    return v
```

Đánh giá giải thuật MINIMAX

- Đầy đủ? Có (nếu cây tìm kiếm là hữu hạn)
- Tối ưu? Có (với một đối thủ đối ưu)
- Độ phức tạp thời gian: $\Phi(b^d)$
- Độ phức tạp không gian: $\Phi(b^d)$ (tìm kiếm theo chiều sâu)
- Với cờ vua: $b \approx 35$, $d \approx 100$ với một ván thông thường → không tìm được lời giải tối ưu

Minimax với độ sâu lớp cố định

- Minimax đối với một KGTT giả định



Các nút lá được gán các giá trị lợi ích (***heuristic)*** nào đó

Còn giá trị tại các nút trong là các giá trị nhận được dựa trên giải thuật Minimax *(min hay max cua các nút con)*

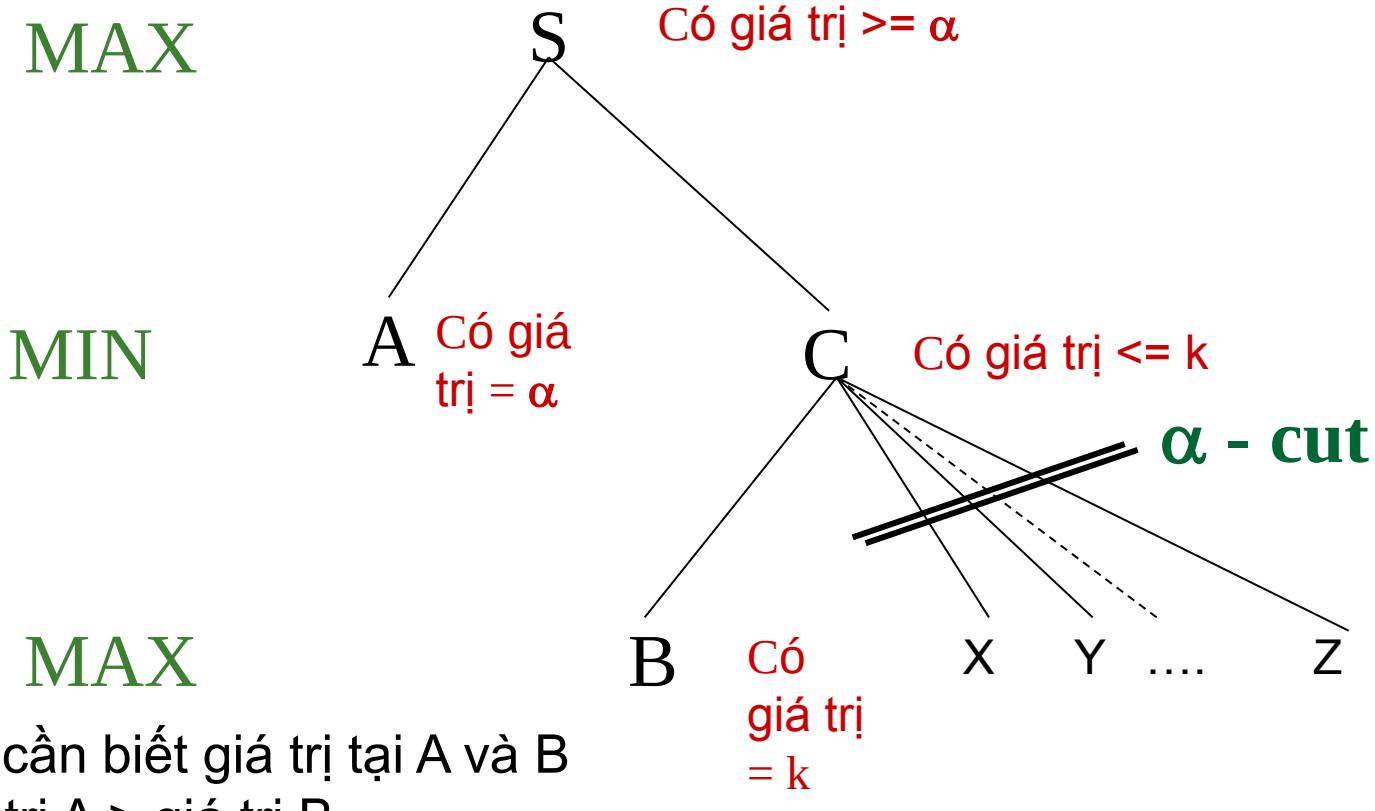
Cắt tỉa α-β (alpha-beta)

- Ta có thể làm gì để hạn chế số lượng TT phải kiểm tra ngoài việc hạn chế số mức d đi vì số trạng thái vẫn còn quá lớn
- Cờ vua: nhân tố nhánh b=35; d=3 có 35*35*35=42.785 trạng thái
- Giảm bớt các trạng thái cần khảo sát mà vẫn không ảnh hưởng gì đến việc giải quyết bài toán
- Cắt tỉa các nhánh không cần khảo sát

Chiến lược cắt tỉa α-β

- Tìm kiếm theo kiểu depth-first
- Nút MAX có 1 giá trị α (luôn tăng)
- Nút MIN có 1 giá trị β (luôn giảm)
- Tìm kiếm có thể kết thúc dưới bất kỳ
  - Nút MIN nào có **β ≤ α** của bất kỳ nút cha MAX nào
  - Nút MAX nào có **α ≥ β** của bất kỳ nút cha MIN nào
- Cắt tỉa α-β thể hiện *mối quan hệ giữa các nút ở mức **n** và **n+2***, mà tại đó toàn bộ cây có gốc tại mức **n+1** có thể cắt bỏ

Cắt tỉa α (vị trí MAX)



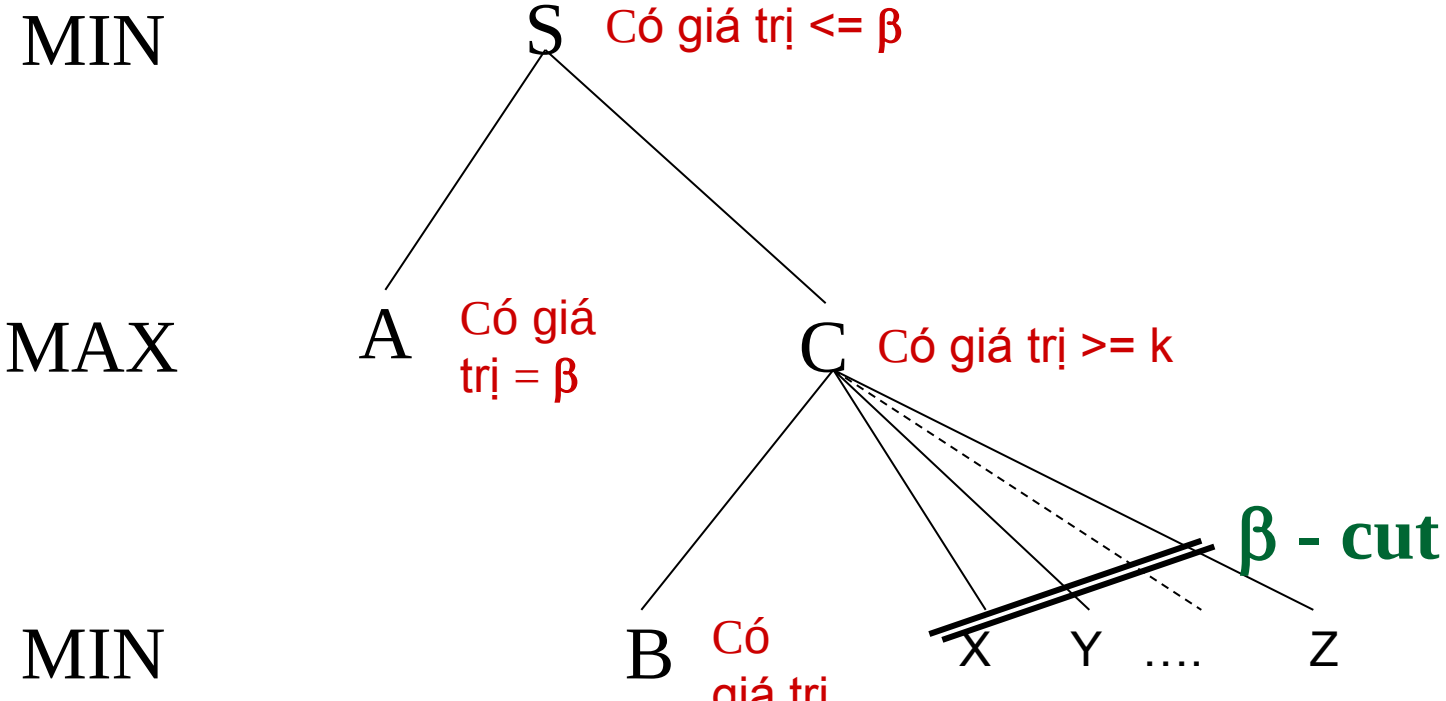
Điều kiện 1: Chỉ cần biết giá trị tại A và B

Điều kiện 2: Giá trị A > giá trị B

Điều kiện 3: X, Y, .. , Z ở vị trí Max

Bỏ những cây con có gốc là X,Y,…, Z

Cắt tỉa β (vị trí Min)



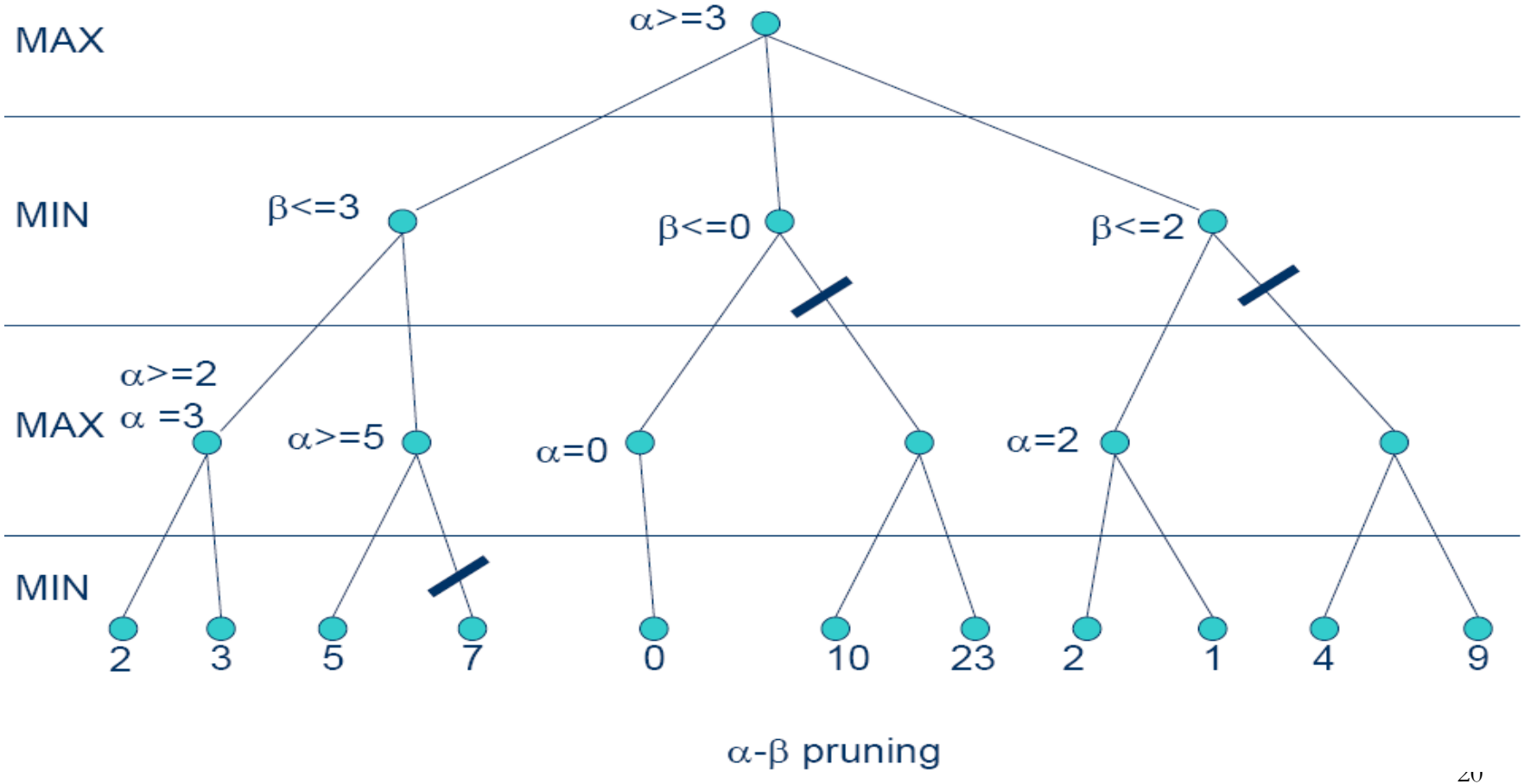
Điều kiện 1: Chỉ cần biết giá trị tại A và B

Điều kiện 2: Giá trị A < giá trị B

Điều kiện 3: X, Y, .. , Z ở vị trí Min

Bỏ những cây con có gốc là X,Y,…, Z

Cắt tỉa α-β: ví dụ



α-β pruning

Giải thuật cắt tỉa α-β

function ALPHA-BETA-SEARCH (state) **returns** an action
  input: state, current state in game
  v ← MAX-VALUE (state, -∞,+∞)
  return the action in **succs**(state) with the value v

function MAX-VALUE (state, α, β) **returns** an utility value
  input: state, current state in game
    α: the value of the best alterminate for MAX along the patth to state
    β: the value of the best alterminate for MIN along the patth to state
  if TERMINAL-TEST (state) **then return** the UTILITY (state)
  v ← -∞
  for each s in **succs**(state) do
    v ← MAX(v, MIN-VALUE(s, α, β))
    if v ≥ β **then** return v
    α ← MAX(α, v)

Giải thuật cắt tỉa α-β

function MIN-VALUE (state, α, β) **returns** an utility value
input: state, current state in game
α: the value of the best alterminate for MAX along the patth to state
β: the value of the best alterminate for MIN along the patth to state
if TERMINAL-TEST (state) **then return** the UTILITY (state)
$v \leftarrow +\infty$
for each s in **succs**(state) do
$v \leftarrow$ MIN(v, MIN-VALUE(s, α, β))
if $v \leq \alpha$ **then** return v
$\beta \leftarrow$ MIN(β, v)

Nghĩ trước giới hạn và hàm lượng giá

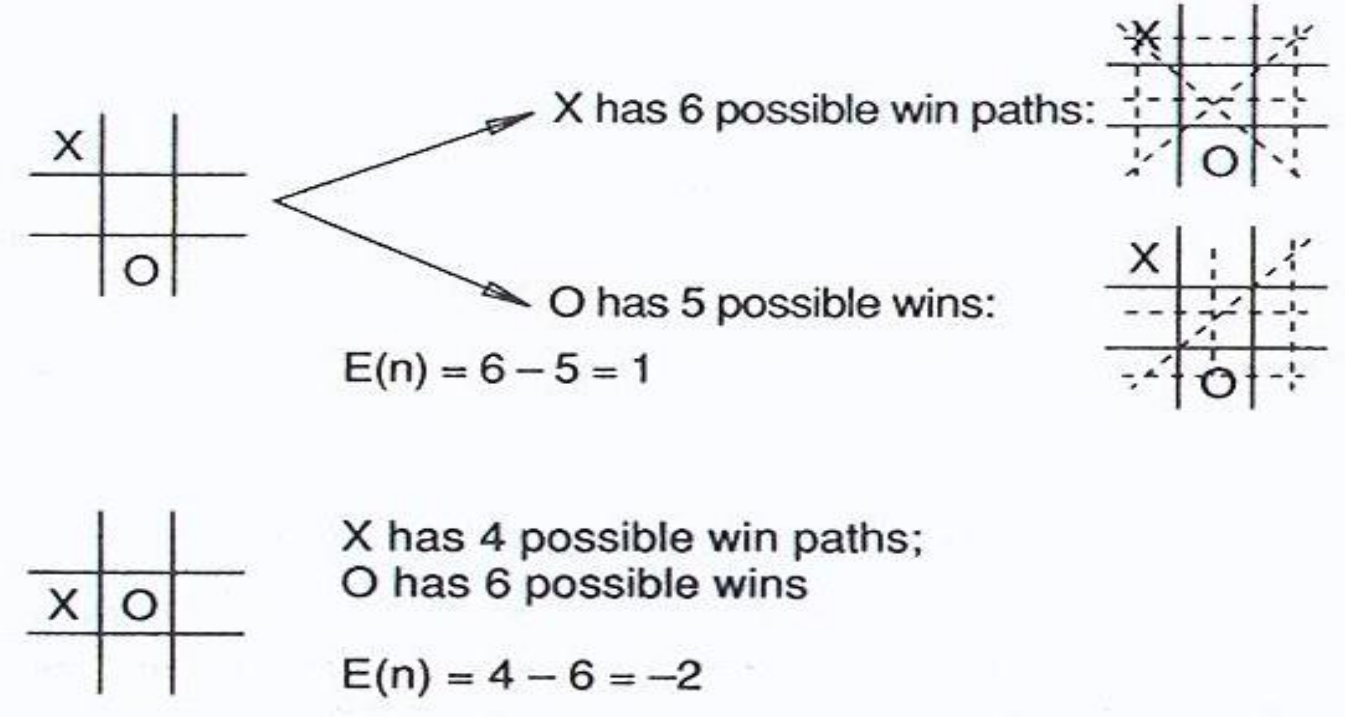
- Các trò chơi thường có độ sâu lớn (≥ 35 đối với cờ vua)
- Trong thời gian thực không thể đi đến trạng thái kết thúc để đánh giá 1 nước đi → nghĩ trước giới hạn số nước
- Cần một hàm lượng giá các trạng thái không kết thúc thay hàm đánh giá lợi ích cho trạng thái kết thúc

Hàm lượng giá

- Đánh giá mức độ thành công của 1 nước đi (thắng, thua, hòa)
- Đánh giá tuyến tính tổng các đặc trưng có được của đối thủ
  - $Eval(s) = w_1f_1(s) + w_2f_2(s) + \ldots + w_nf_n(s)$
    - Trong đó w_i : trọng số gán cho quân thứ i (hậu w=9; ngựa w=3)

      f_i : số quân còn lại

Heuristic trong trò chơi tic-tac-toe



Hàm Heuristic: **E(n) = M(n) – O(n)**

Trong đó: M(n) là *tổng số đường thắng có thể của tôi*

O(n) là *tổng số đường thắng có thể của đối thủ*

E(n) là *trị số đánh giá tổng cộng cho trạng thái n*

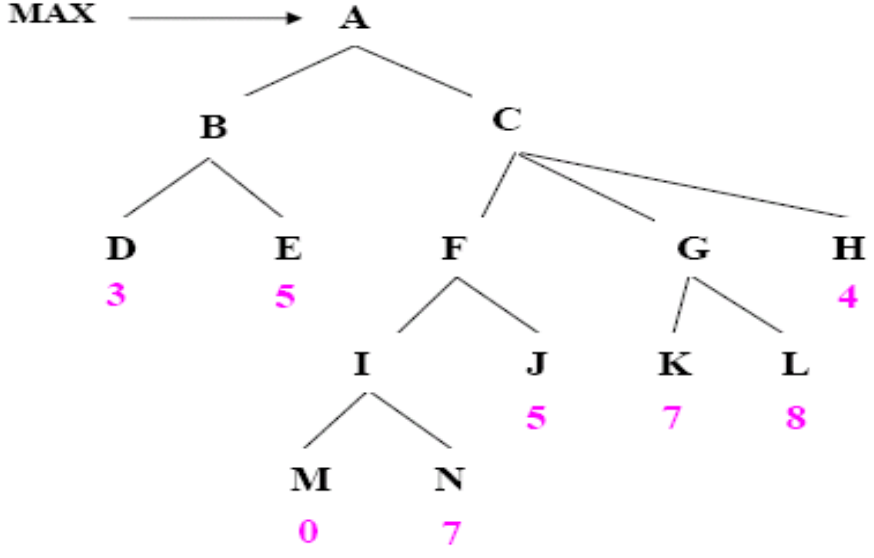
Các link minh họa tìm kiếm

- Giải thuật MiniMax (GS Patrick Henry Winston, MIT)
  - http://www.ai.mit.edu/courses/6.034f/gamepair.html
- Giải thuật MiniMax với cắt tỉa (GS Patrick Henry Winston, MIT)
  - http://www.ai.mit.edu/courses/6.034f/searchpair.html

Bài tập: bài 1 (minimax)

Liệt kê danh sách các nút được duyệt theo tìm kiếm DFS.

- Thực hiện giải thuật Minimax trên cây.



- Sẽ có gì khác biệt nếu như ta dùng giải thuật cắt tỉa alpha – beta để định trị nút gốc cho cây?