CHƯƠNG 4: TÌM KIẾM CÓ ĐỐI THỦ (5 T)

4.1. Cây trò chơi và tìm kiếm trên cây trò chơi (2t)

4.2. Chiến lược MiniMax (1,5t)

4.3. Phương pháp cắt cụt AlphaBeta (1,5t)

4.1. CÂY TRÒ CHƠI VÀ TÌM KIẾM TRÊN CÂY TRÒ CHƠI

- Trong bài, nghiên cứu các trò chơi cho 2 người tham gia như:
  - Cờ vua
  - Cờ ca ro
  - Cờ tướng
- Người chơi là quân trắng, đối thủ là quân đen
- **Mục tiêu:** nghiên cứu giải thuật cho quân trắng đi

4.1. CÂY TRÒ CHƠI VÀ TÌM KIẾM TRÊN CÂY TRÒ CHƠI

- **Một số đặc điểm:**
  - 2 người thay phiên đưa ra các nước đi theo một số luật nào đó
  - Các luật trên là như nhau cho cả 2 người
  - Cả hai người chơi đều biết được thông tin đầy đủ về các tình thế trong trò chơi
  - Trong vấn đề trò chơi, thực chất là tìm kiếm nước đi, một nước tốt sao cho, sau một số nước đi dẫn đến trạng thái kết thúc.

4.1. CÂY TRÒ CHƠI VÀ TÌM KIẾM TRÊN CÂY TRÒ CHƠI

- **Khó khăn:**
  - Trong vấn đề này có đối thủ, nên không biết đối thủ sẽ đi như thế nào
  - Nếu có thể tổng quát thì sẽ rất khó vì không gian tìm kiếm quá rộng.
  - Chỉ tìm được lời giải tối ưu, không tìm được lời giải sấp sỉ

4.1. CÂY TRÒ CHƠI VÀ TÌM KIẾM TRÊN CÂY TRÒ CHƠI

- **Giải pháp:**
  - Trong trò chơi có thể coi như tìm kiếm trong một không gian trạng thái, mỗi trạng thái là một tình thế của trò chơi:
    - **Trạng thái ban đầu:** sự sắp xếp các quân cờ trong lúc bắt đầu của cuộc chơi
    - **Các toán tử:** Các nước đi hợp lệ
    - **Các trạng thái kết thúc:** các tình thế khiến cuộc chơi dừng, thường đã được xác định, có thể thông qua hàm kết quả
    - Có thể biểu diễn không gian trạng thái trên cây trò chơi

4.1. CÂY TRÒ CHƠI VÀ TÌM KIẾM TRÊN CÂY TRÒ CHƠI

- **Cách xây dựng trò chơi:**

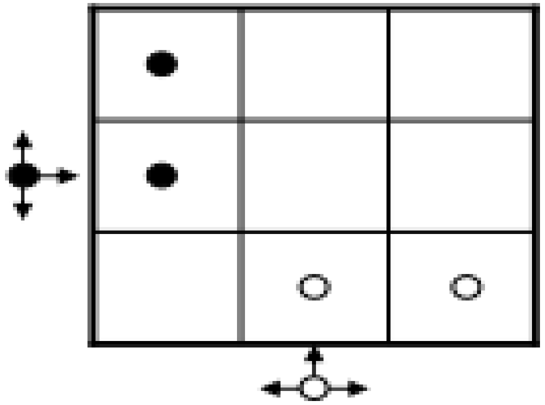
– **Gốc** ứng với trạng thái đầu
– **Đỉnh** ứng với trạng thái mà Trắng (Đen) đưa ra nước đi gọi là đỉnh trắng (Đen)
– **Các đỉnh con** của đỉnh Trắng (Đen) biểu diễn trạng thái u là tất cả các đỉnh biểu diễn trạng thái v, v nhận được từ u do Trắng (Đen) thực hiện nước đi hợp lệ nào đó.
– **Lá** của cây ứng với trạng thái kết thúc.

Nhận xét:

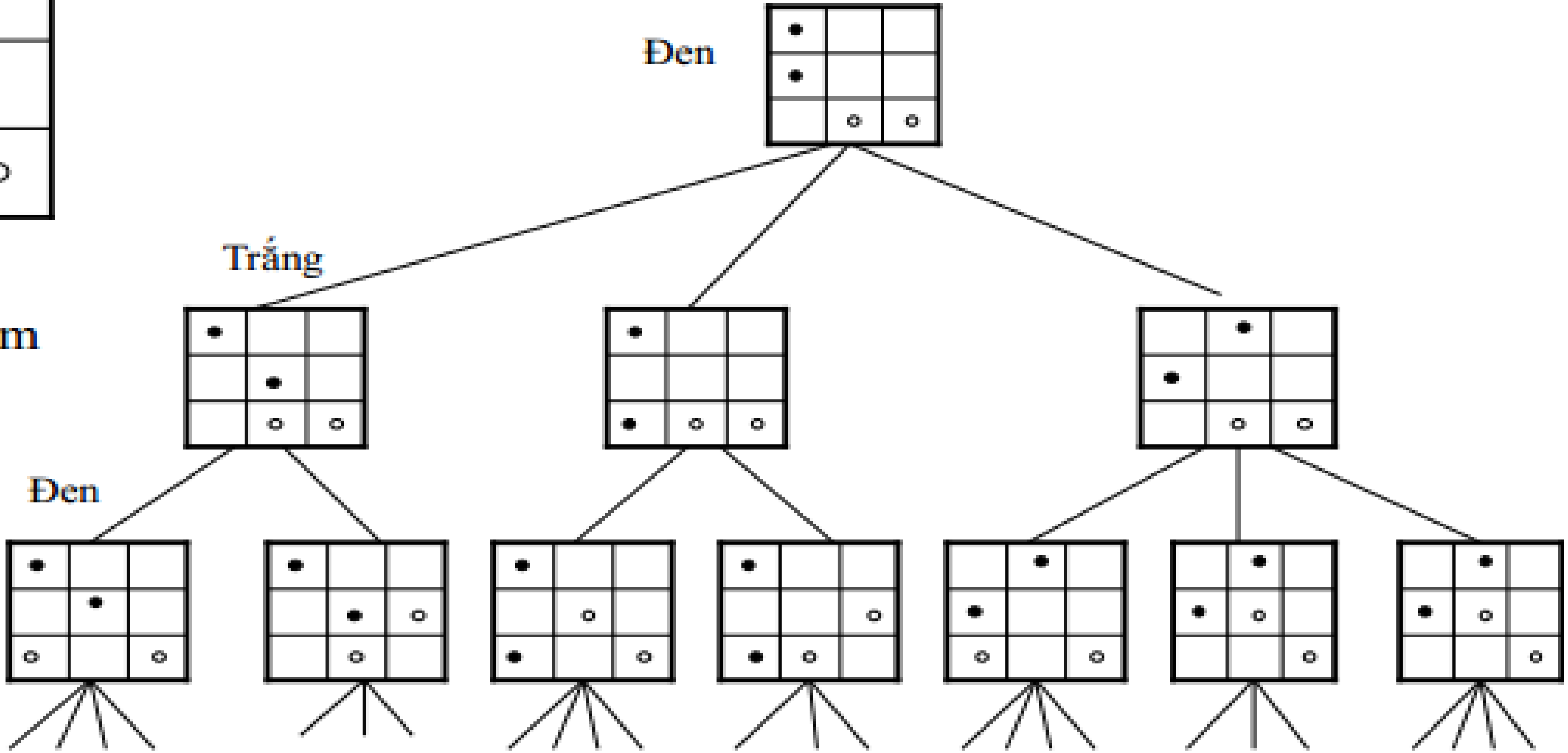
- Độ cao của cây là tổng số nước đi của cả 2 người
- Trên cùng một mức của cây, các đỉnh đều là trắng hoặc đen.
- Các lá của cây ứng với các trạng thái kết thúc.

4.1. CÂY TRÒ CHƠI VÀ TÌM KIẾM TRÊN CÂY TRÒ CHƠI

- **Cách xây dựng trò chơi:**



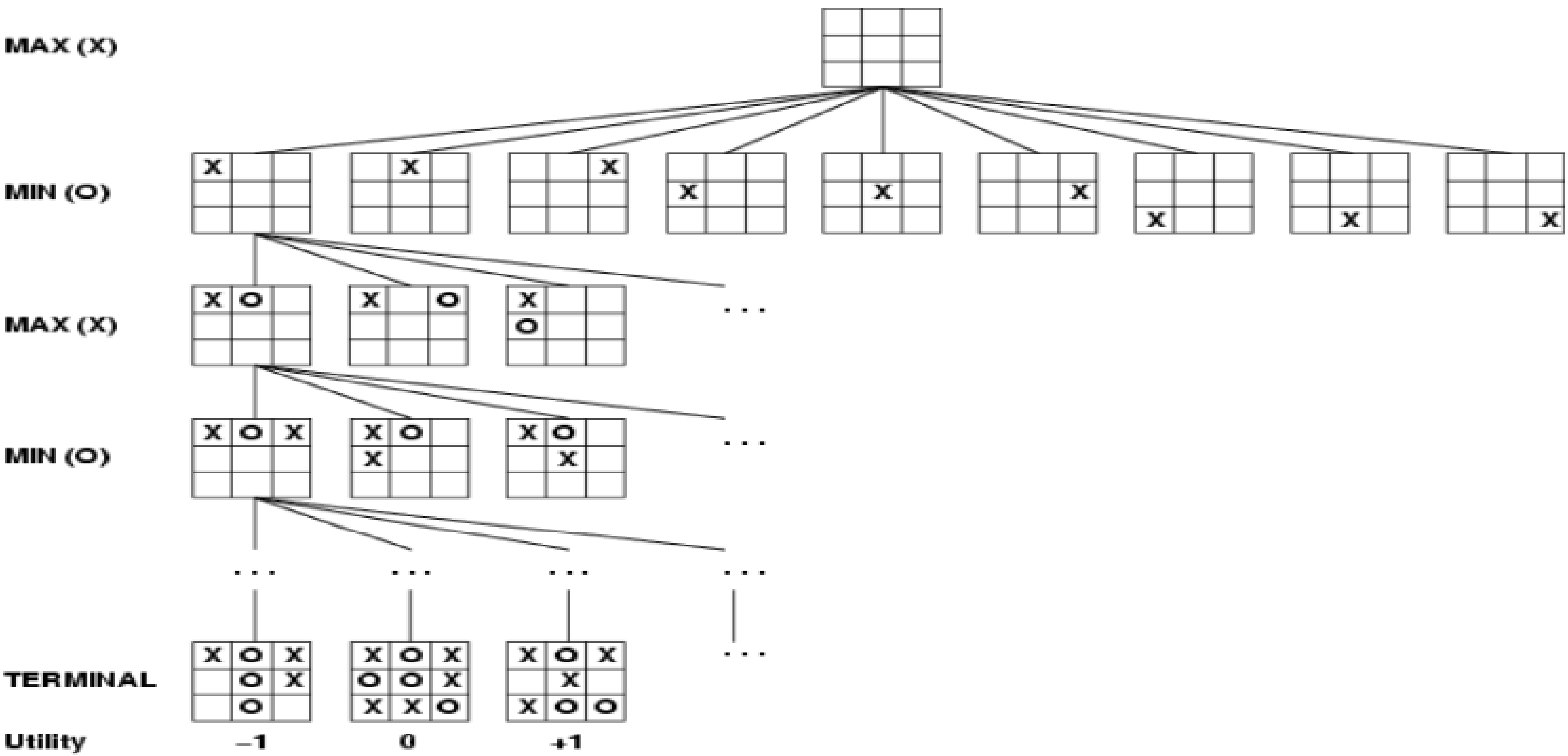
Trò chơi Dodgem



Cây trò chơi Dodgem với Đen đi trước

4.1. CÂY TRÒ CHƠI VÀ TÌM KIẾM TRÊN CÂY TRÒ CHƠI

- **Cách xây dựng trò chơi:**



4.2. CHIẾN LƯỢC MINIMAX

- **Nhận định:** Giả sử đến một thời điểm, đường đi đã đến được đỉnh u:

- Nếu u là đỉnh trắng, thì trắng cần chọn để đi đến một trong các đỉnh đen v là các đỉnh con của u
  nước đi tối ưu cho trắng là nước đi dẫn tới các đỉnh con v là đỉnh tốt nhất cho trắng trong số các đỉnh con. Tương tự như vậy với việc chọn nước đi cho quân đen
- Để chọn nước đi tốt nhất cho trắng tại đỉnh u, ta cần xác định các đỉnh của cây trò chơi có gốc là u .
- Giá trị của các lá được xác định thông qua hàm kết quả
- Đỉnh có giá trị càng lớn (max) càng tốt cho trắng, đỉnh có giá trị càng **nhỏ (min)** càng tốt cho **đen**

4.2. CHIẾN LƯỢC MINIMAX

- **Cách xây dựng trò chơi:**

– Hai đấu thủ trong trò chơi được gọi là **MIN** và **MAX**.

– Mỗi nút lá có giá trị:

• **1** nếu là MAX thắng,

• **0** nếu là MIN thắng.

– Minimax sẽ truyền các giá trị này lên cao dần trên đồ thị, qua các nút cha mẹ kế tiếp theo các luật sau:

• Nếu trạng thái cha mẹ là **MAX**, gán cho nó giá trị **lớn nhất** có trong các trạng thái con.

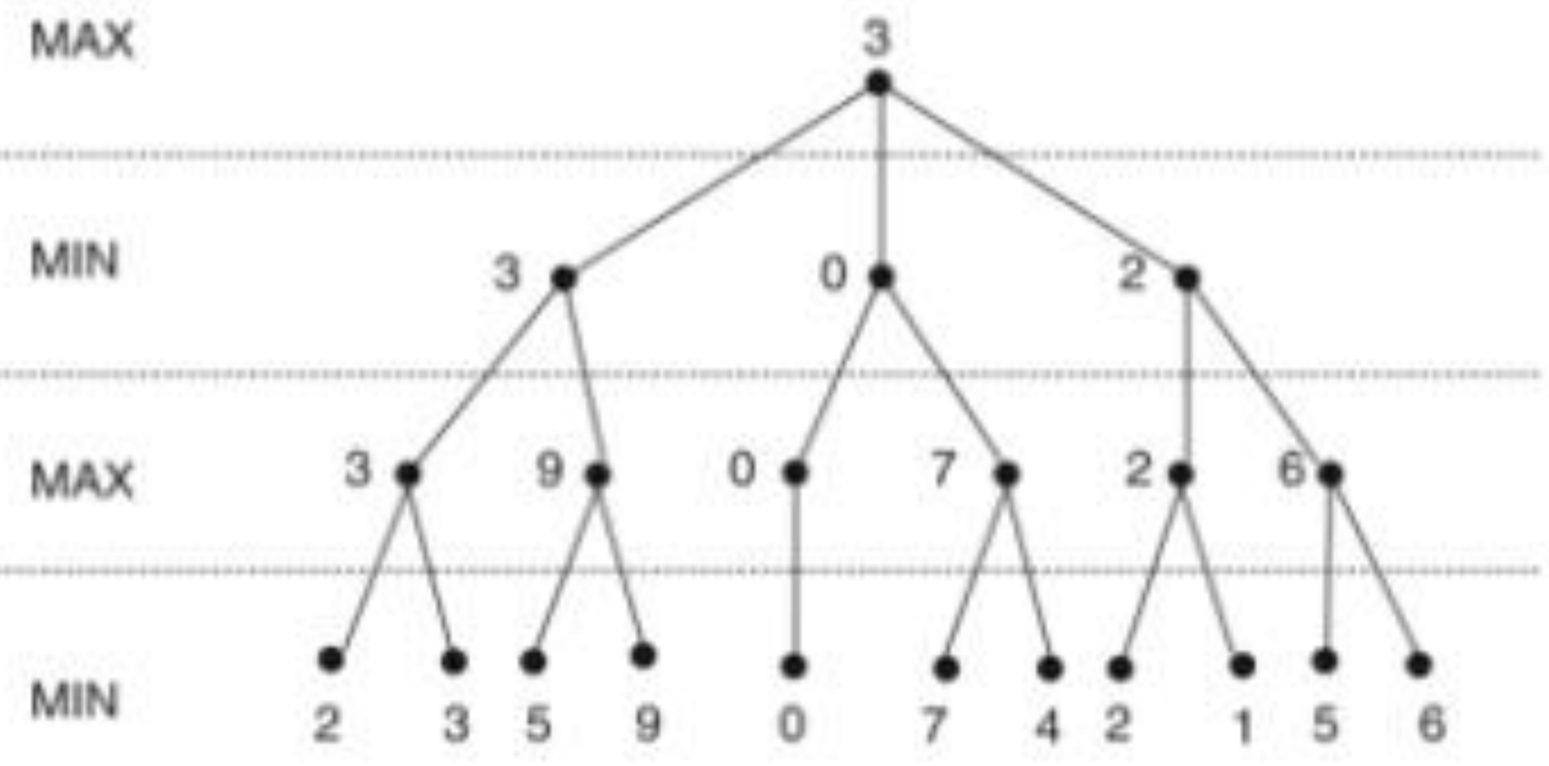
• Nếu trạng thái cha mẹ là **MIN**, gán cho nó giá trị **nhỏ nhất** có trong các trạng thái con.

4.2. CHIẾN LƯỢC MINIMAX

Minimax đối với một KGTT giả định.

 Các nút lá được gán các giá trị ***heuristic***

 Còn giá trị tại các nút trong là các giá trị nhận được dựa trên giải thuật Minimax

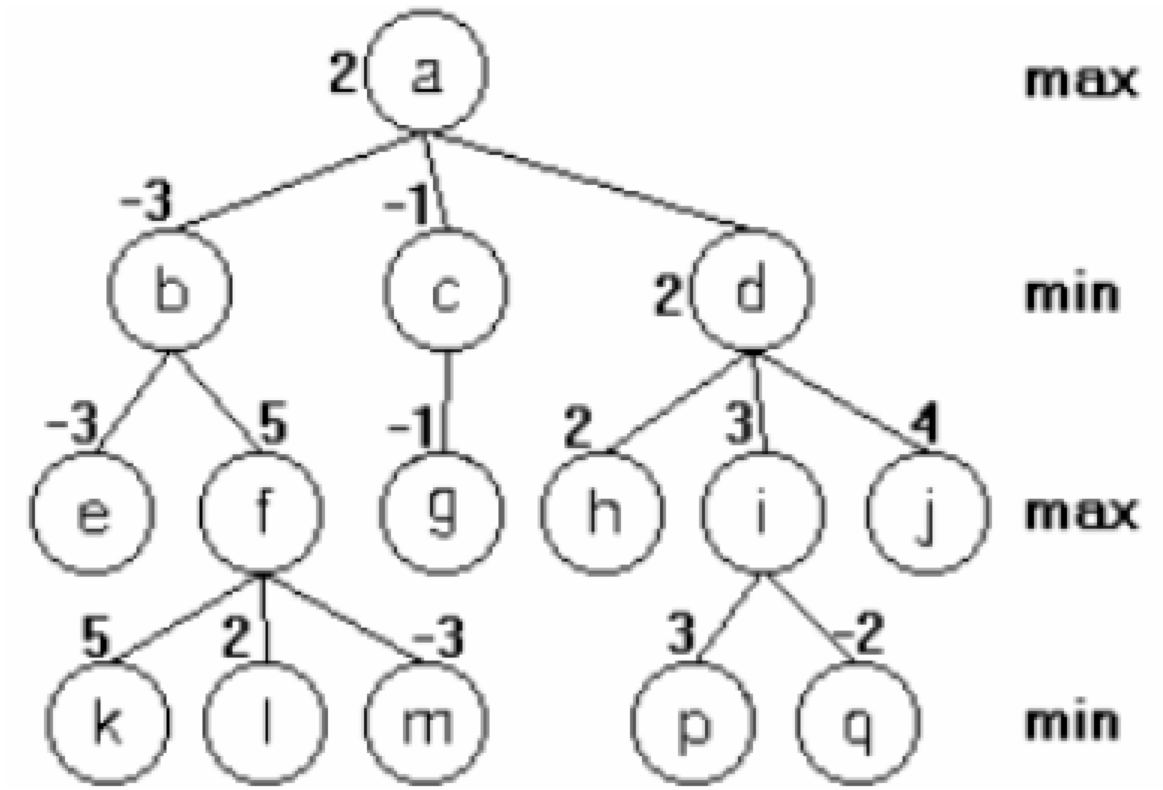


4.2. CHIẾN LƯỢC MINIMAX

Cách tính điểm cho các đỉnh trên cây trò chơi:

- Để xác định giá trị các đỉnh có gốc là u, ta đi từ mức thấp nhất đến u.
- Giả sử xét đỉnh v trên cây, các giá trị các đỉnh con của nó đã xác định.
- Nếu v là đỉnh Trắng (max) , giá của nó được xác định là giá trị lớn nhất trong các giá trị của các đỉnh con.
- Nếu v là đỉnh **đen (min)** , giá của nó được xác định là **giá trị nhỏ nhất** trong các giá trị của các đỉnh con.

4.2. CHIẾN LƯỢC MINIMAX



Gán giá trị cho các đỉnh của cây trò chơi.

Ví dụ:

đỉnh f là đỉnh Trắng, giá của đỉnh f = max(5,2,-3) = 5

đỉnh d là đỉnh Đen, giá của đỉnh d = min(2,3,4) = 2

4.2. CHIẾN LƯỢC MINIMAX

function *MaxVal(u); // hàm gán giá trị max*
begin
if *u là đỉnh kết thúc* **then** *MaxVal(u)* ← *f(u) //f(u) là giá trị của hàm kết cuộc tại đỉnh kết thúc u*
else *MaxVal(u)* ← *max{MinVal(v) | v là đỉnh con của u}*
end;

function *MinVal(u); // hàm gán giá trị min*
begin
if *u là đỉnh kết thúc* **then** *MinVal(u)* ← *f(u) //f(u) là giá trị của hàm kết cuộc tại đỉnh kết thúc u*
else *MinVal(u)* ← *min{MaxVal(v) | v là đỉnh con của u}*
end;

Thủ tục minimax:

procedure *Minimax(u, v);*
begin
val $\leftarrow$ -∞;
for *mỗi w là đỉnh con của u* **do**
if *val <= MinVal(w)* **then**
{val $\leftarrow$ *MinVal(w); v* $\leftarrow$ *w}*
end;

trong đoạn chương trình trên, chọn nước đi cho trắng tại u, v là biến lưu lại trạng thái mà trắng đã chọn đi tới từ u

HÀM ĐÁNH GIÁ

Hàm đánh giá eval cho mỗi đỉnh u là đánh giá "mức độ lợi thế" của trạng thái u.

Giá trị của eval(u):

- Là số dương càng lớn thì trạng thái u càng có lợi cho Trắng
- Là số dương càng nhỏ thì trạng thái u càng có lợi cho Đen,
- eval(u)=0 thì trạng thái u không có lợi cho đối thủ nào
- eval(u)=+∝ thì u là trạng thái thắng cuộc cho Trắng
- eval(u)=-∝ thì u là trạng thái thắng cuộc cho Đen

HÀM ĐÁNH GIÁ- VÍ DỤ 1

Hàm đánh giá cho cờ vua. Mỗi loại quân được gán một giá trị số phù hợp với "sức mạnh" của nó. Chẳng hạn:

- Quân tốt Trắng (Đen) được gán giá trị 1 (-1).
- Mã hoặc tượng Trắng (Đen) được gán giá trị 3 (-3)
- Xe Trắng (Đen) được gán giá trị 5 (-5)
- Hậu Trắng (Đen) được gán giá trị 9 (-9).

Hàm đánh giá của một trạng thái được tính bằng cách lấy tổng giá trị của tất cả các quân cờ trong trạng thái đó. Hàm đánh giá này được gọi là hàm tuyến tính có trọng số, vì có thể biểu diễn dưới dạng:

$$s_1w_1 + s_2w_2 + \ldots + s_nw_n$$

Trong đó, wi là giá trị của quân cờ loại i, si là số quân loại đó.

HÀM ĐÁNH GIÁ- VÍ DỤ 2

Bây giờ ta đưa ra một cách đánh giá các trạng thái trong trò chơi Dodgem. Mỗi quân Trắng ở một vị trí trên bàn cờ được cho một giá trị tương ứng trong bảng bên trái hình. Còn mỗi quân Đen ở một vị trí sẽ được cho một giá trị tương ứng trong bảng bên phải hình.

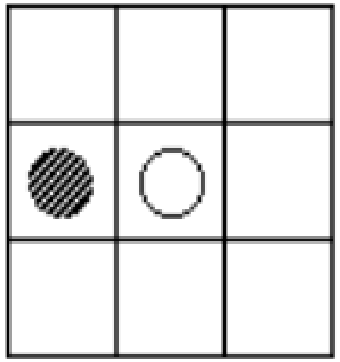
Ngoài ra, nếu quân Trắng cản trực tiếp một quân đen, nó được thêm 40 điểm, nếu cản gián tiếp nó được thêm 30. Tương tự, nếu quân Đen cản trực tiếp quân Trắng nó được thêm -40 điểm, còn cản gián tiếp nó được thêm -30 điểm.

30	35	40
15	20	25
0	5	10

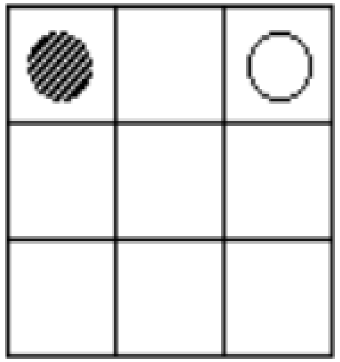
Giá trị quân Trắng.

-10	-25	-40
-5	-20	-35
0	-15	-30

Giá trị quân Đen.



Trắng cản trực tiếp Đen được thêm 40 điểm.

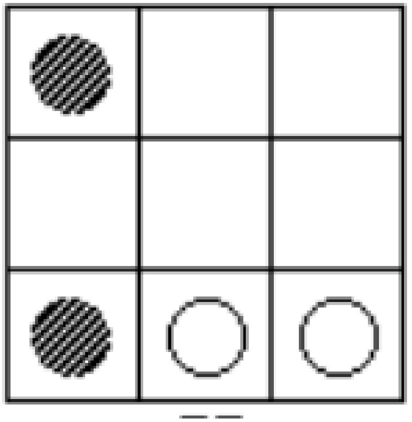


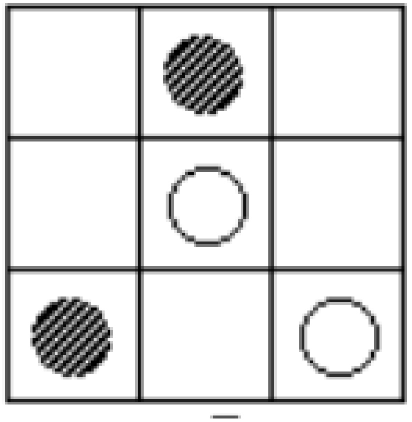
Trắng cản gián tiếp Đen được thêm 30 điểm.

HÀM ĐÁNH GIÁ- VÍ DỤ 2

Giá trị hàm đánh giá trong hình 1:75= (-10+0+5+10)+(40+30)

Giá trị hàm đánh giá trong hình 2:-5= (-25+0+20+10)+(-40+30)





30	35	40
15	20	25
0	5	10

Giá trị quân Trắng.

-10	-25	-40
-5	-20	-35
0	-15	-30

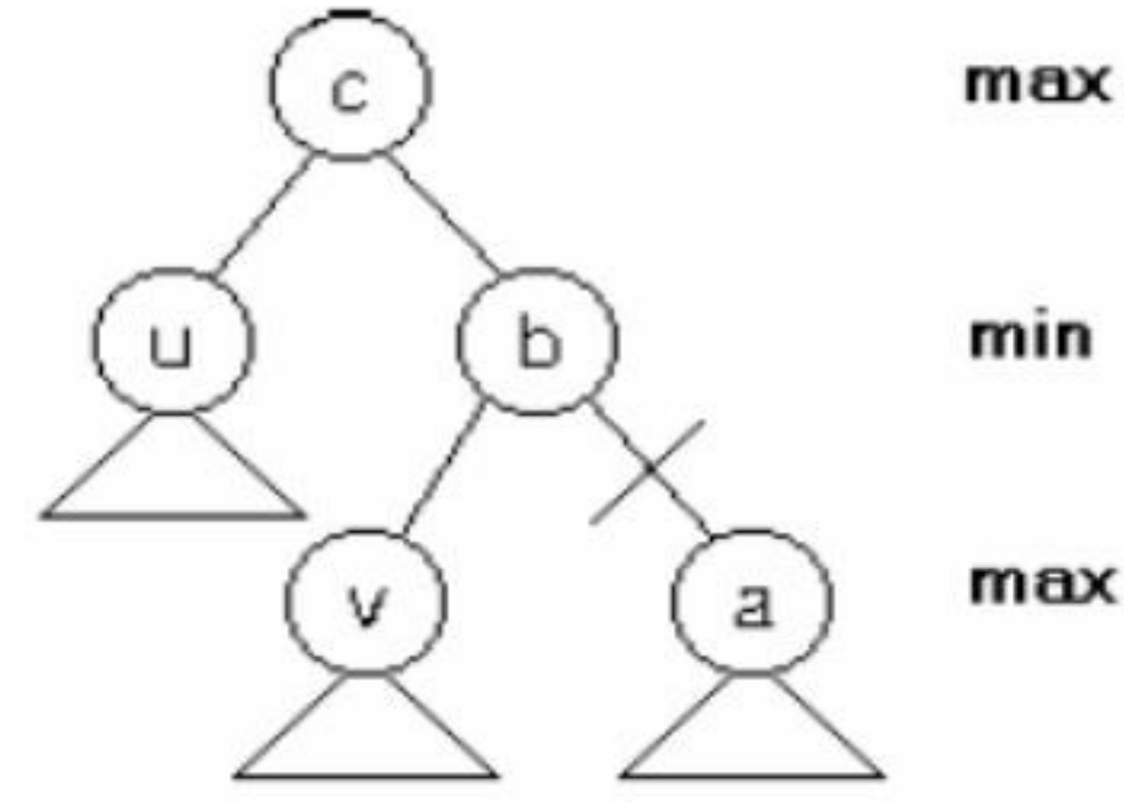
Giá trị quân Đen.

4.2. PHƯƠNG PHÁP CẮT CỤT ALPHA- BETA

Vấn đề: Giải thuật tìm kiếm MINIMAX vấp phải vấn đề bùng nổ (mức hàm mũ) các khả năng nước đi cần phải xét → không phù hợp với nhiều bài toán trò chơi thực tế .

- Chúng ta có thể cắt tỉa (bỏ đi – không xét đến) một số nhánh tìm kiếm trong cây biểu diễn trò chơi
- **Ý tưởng:** Nếu một nhánh tìm kiếm nào đó không thể cải thiện đối với giá trị (hàm tiện ích) mà chúng ta đã có, thì không cần xét đến nhánh tìm kiếm đó nữa.
- Việc cắt tỉa các nhánh tìm kiếm ("tồi") không ảnh hưởng đến kết quả cuối cùng

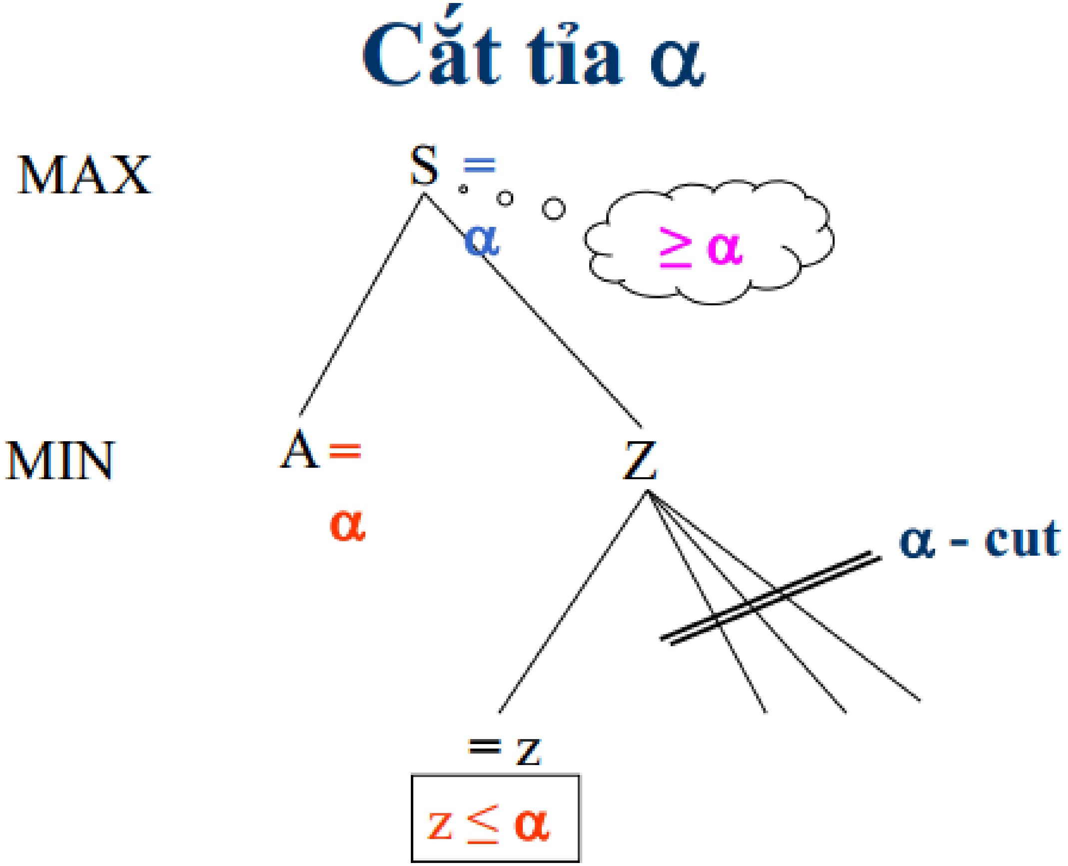
4.2. PHƯƠNG PHÁP CẮT CỤT ALPHA- BETA



Cắt tỉa gốc a nếu eval(u)>eval(v)

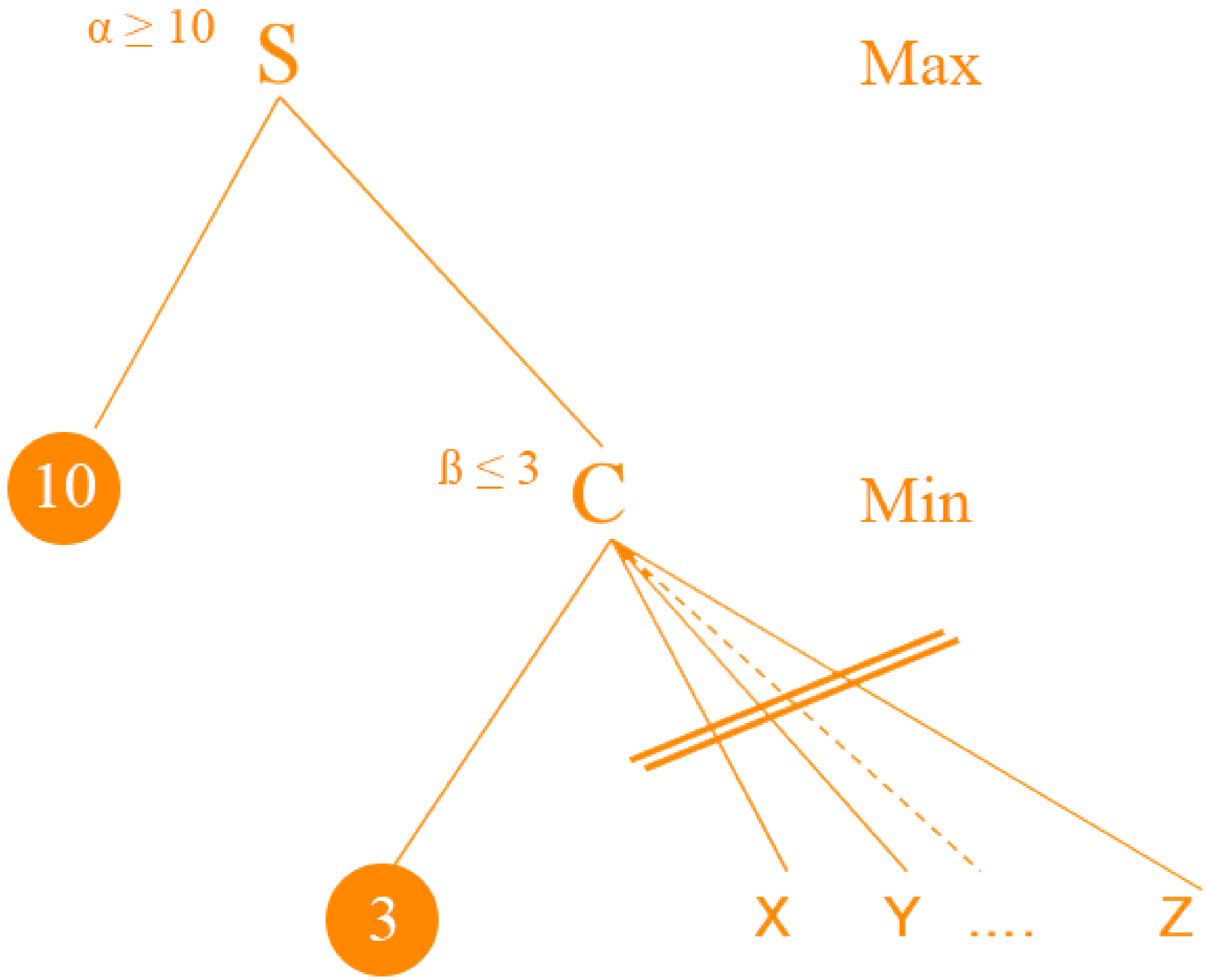
Giả sử quá trình tìm kiếm đi xuống đỉnh trắng **a**, đỉnh **a** có đỉnh cùng cấp **v** đã xét.
Giả sử **a** có cha là **b, b** có đỉnh cùng cấp **u** đã xét.
Cha của **b** là **c**
Khi đó giá của **c** ít nhất là **u** và giá của **b** nhiều nhất là **v.**
Nếu **Eval(u)>eval(v),** ta không cần đi xuống đỉnh **a** nữa mà không ảnh hưởng đến giá của **c.**
Lập luận tương tự với đỉnh <u>**a**</u> là đen với đánh giá **eval(u)<eval(v)**

4.2. PHƯƠNG PHÁP CẮT CỤT ALPHA- BETA

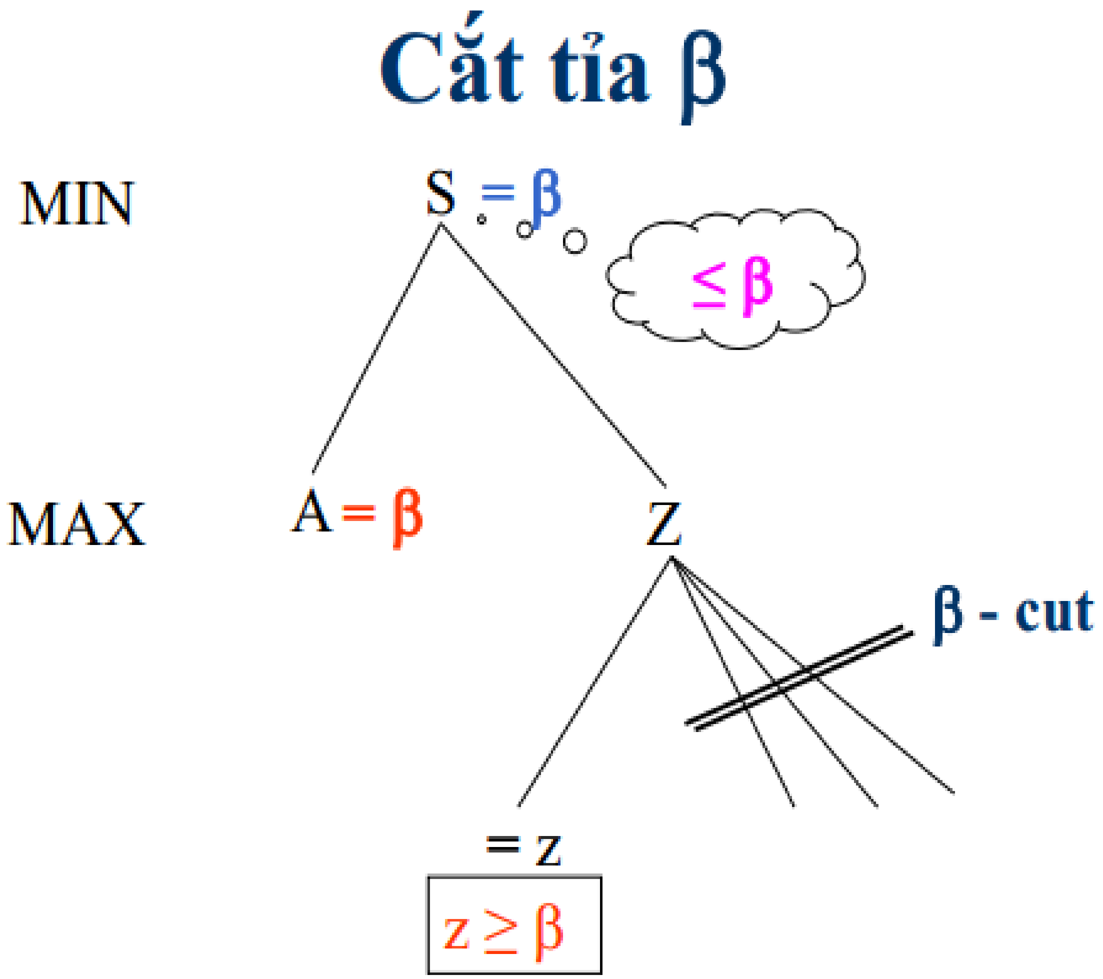


Giả sử quá trình tìm kiếm đi xuống đỉnh trắng **a**, đỉnh **a** có đỉnh cùng cấp **v** đã xét.
Giả sử **a** có cha là **b, b** có đỉnh cùng cấp **u** đã xét.
Cha của **b** là **c**
Khi đó giá của **c** ít nhất là **u** và giá của **b** nhiều nhất là **v.**
Nếu **Eval(u)>eval(v),** ta không cần đi xuống đỉnh **a** nữa mà không ảnh hưởng đến giá của **c.**
Lập luận tương tự với đỉnh **a** là đen với đánh giá **eval(u)<eval(v)**

4.2. PHƯƠNG PHÁP CẮT CỤT ALPHA- BETA

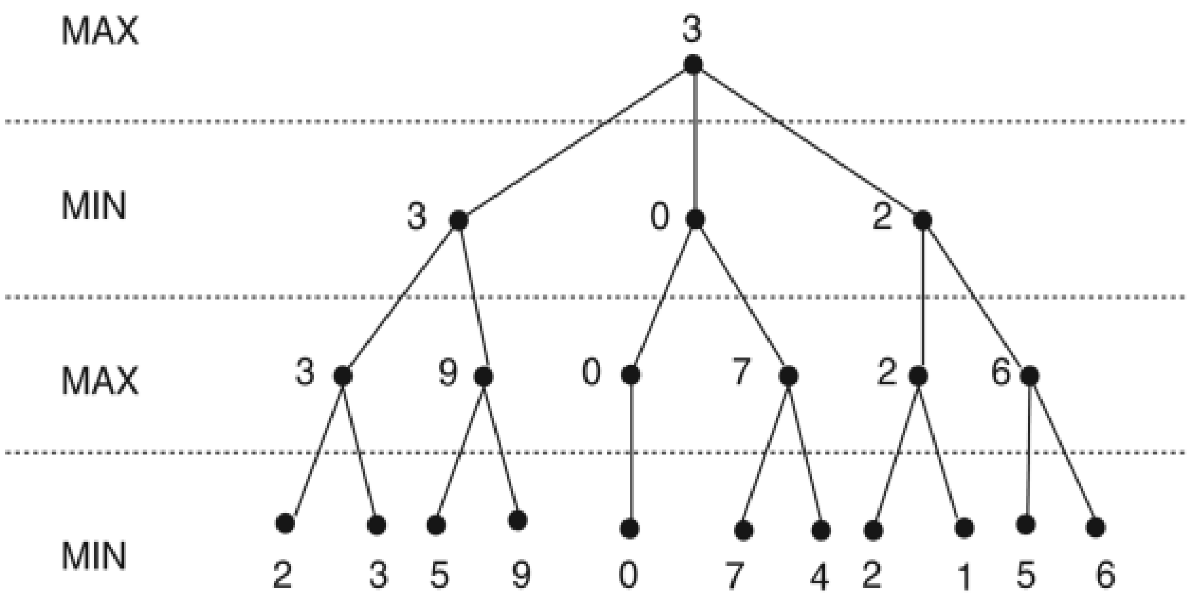


4.2. PHƯƠNG PHÁP CẮT CỤT ALPHA- BETA



Giả sử quá trình tìm kiếm đi xuống đỉnh trắng **a**, đỉnh **a** có đỉnh cùng cấp **v** đã xét.
Giả sử **a** có cha là **b, b** có đỉnh cùng cấp **u** đã xét.
Cha của **b** là **c**
Khi đó giá của **c** ít nhất là **u** và giá của **b** nhiều nhất là **v.**
Nếu **Eval(u)>eval(v),** ta không cần đi xuống đỉnh **a** nữa mà không ảnh hưởng đến giá của **c.**
Lập luận tương tự với đỉnh **a** là đen với đánh giá **eval(u)<eval(v)**

4.2. PHƯƠNG PHÁP CẮT CỤT ALPHA- BETA



- **Giải thuật cắt tỉa** Alpha-Beta
  - Tìm kiếm theo kiểu depth-first.
  - Nút MAX có 1 giá trị α(luôn tăng)
  - Nút MIN có 1 giá trị β (luôn giảm)

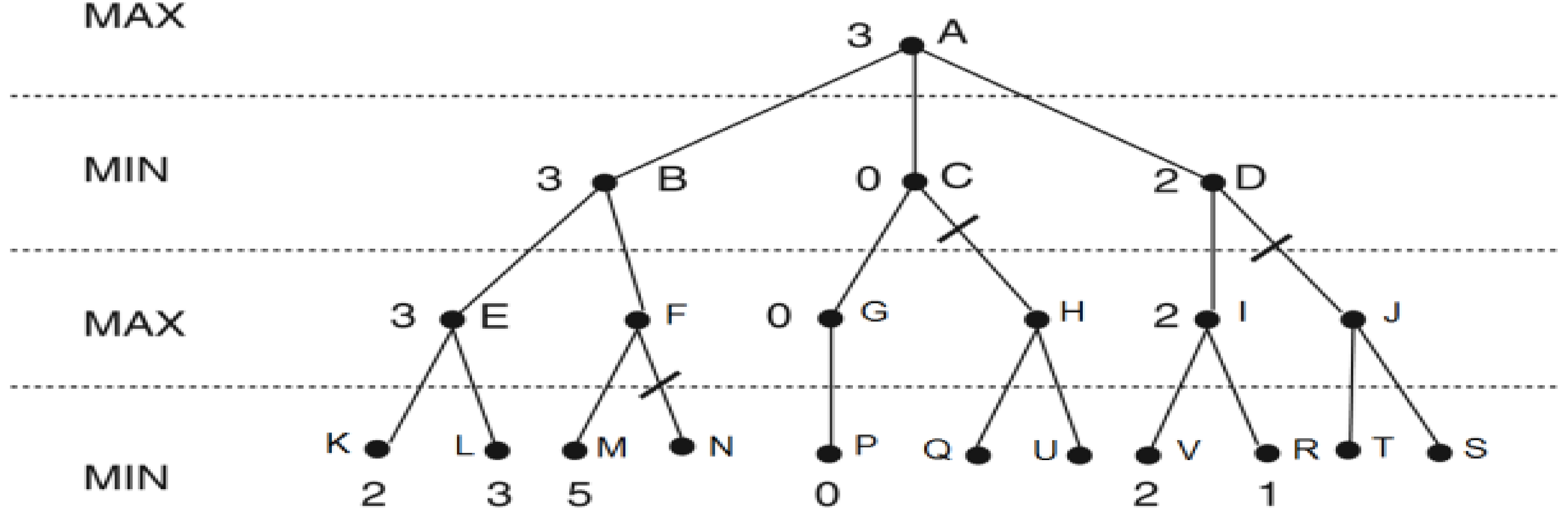
Tìm kiếm có thể kết thúc dưới bất kỳ:

– Nút MIN nào có $\beta \leq \alpha$ của bất kỳ nút cha MAX nào.

– Nút MAX nào có $\alpha \geq \beta$ của bất kỳ nút cha MIN nào.

- Giải thuật cắt tỉa α - β thể hiện *mối quan hệ giữa các nút ở lớp n và n+2*, mà tại đó toàn bộ cây có gốc tại lớp n+1 có thể cắt bỏ.

4.2. PHƯƠNG PHÁP CẮT CỤT ALPHA- BETA



- Xét K(2) và L(3), khi đó E có giá trị 3 = max(2,3).
- Vì M=5, nên ít nhất F=5, do đó, không cần xét nhánh N, có thể kết luận B=3 (cắt bỏ beta).
- Tương tự, xét P(0), suy ra G=0. Do chọn min, suy ra C nhiều nhất =0. Vì A chọn max, nên không cần xét H.
- Tương tự, D nhiều nhất bằng 2, mà chọn A theo max, B>I, nên không cần xét J.

Các hàm trong chiến lược Alpha-beta

- Hàm sử dụng α để ghi giá trị lớn nhất trong các giá trị của đỉnh con đã đánh giá của một đỉnh trắng, β ghi giá trị nhỏ nhất trong các đỉnh con của một đỉnh đen.
- Hàm MaxValue(u, α, β) tính giá của đỉnh Trắng u.
- Hàm MinValue(u, α, β) tính giá của đỉnh Đen u.

Hàm gán giá trị max:

```
Function MaxValue(u, α, β);
Begin
    If u là lá của cây hạn chế hoặc là đỉnh kết thúc
    then MaxValue ← eval(u)
    Else
        for mỗi đỉnh v là con của u do
            begin
          α ← max {α, MinValue(v, α, β)};
          if α > β then exit;
            end;
    MaxValue ← α;
End;
```

Hàm gán giá trị min:

```
Function MinValue(u, α, β);
Begin
   If u là lá của cây hạn chế hoặc là đỉnh kết thúc
   then MinValue ← eval(u)
   Else
      for mỗi đỉnh v là con của u do
          begin
       β ← min {β, MaxValue(v, α, β)};
       if α > β then exit;
          end;
   MinValue ← β;
End;
```

Thủ tục Alpha-Beta: (tìm nước đi cho quân Trắng, v là đỉnh cần tới)

```
Procedure Alpha_Beta(u, v);
Begin
   α ← -∞;
   β ← ∞;
   for mỗi đỉnh w là đỉnh con của u do
   If α<=MinValue(w, α, β) then
     begin
        α=MinValue(w, α, β);
        v = w;
     end;
End;
```