

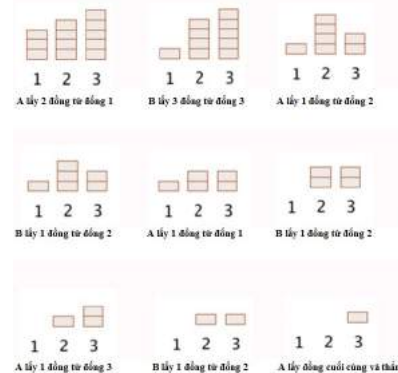
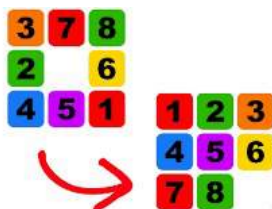
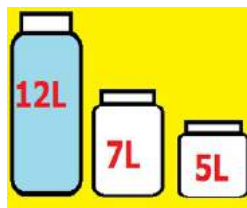
Nội dung

- Biểu diễn bài toán trong KGTT
- Tìm kiếm mù (uninformed search)
- Tìm kiếm heuristic (informed search)
- **Cây trò chơi, cắt tỉa alpha -beta**

142

Sự khác biệt giữa các bài toán?

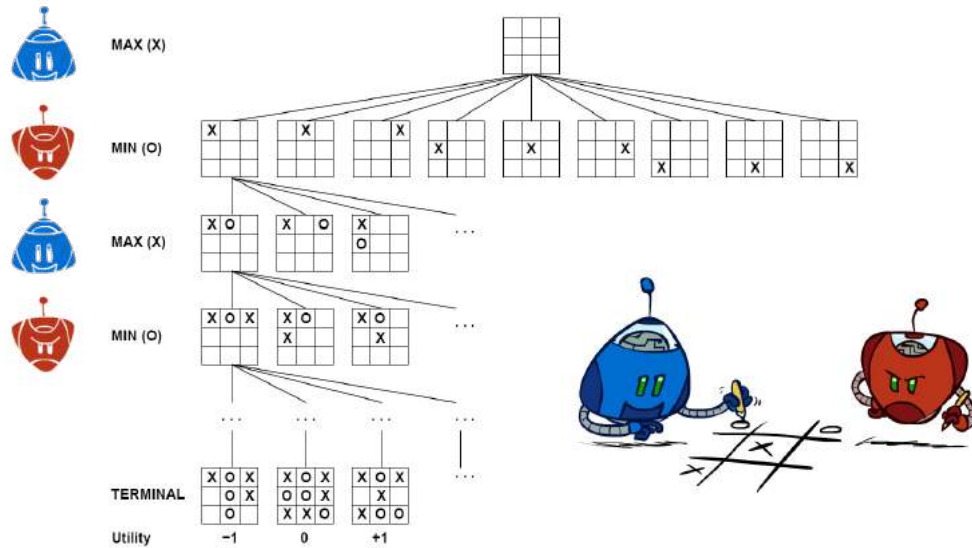
Giả sử có 3 đồng, mỗi đồng lần lượt có 3,4,5 đồng xu



- Có hai người chơi, khi đến lượt, người chơi có thể lấy một số lượng tùy ý đồng xu từ một đồng duy nhất.
- Họ phải lấy ít nhất 1 đồng xu và họ không được lấy các đồng xu từ đồng khác.
- Người thắng là người lấy đồng xu cuối cùng, nghĩa là không còn đồng xu nào sau nước đi của người đó

143

Cây trò chơi Tic-Tac-Toe



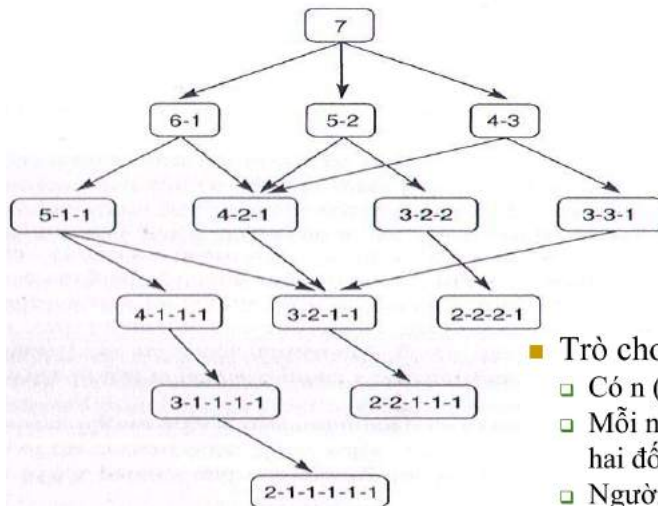
144

Cây 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)

145

Không gian trạng thái trò chơi 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

146

Ứng dụng Heuristic trong các trò chơi

- Sử dụng không gian trạng thái để giải quyết bài toán
 - Tìm kiếm mù
 - Tìm kiếm heuristic – có thông tin bổ sung
 - Sử dụng heuristic cho trò chơi
- Có 2 người tham gia vào quá trình sinh trạng thái
- Bạn tạo ra trạng thái này, đối thủ của bạn sẽ tạo ra trạng thái kế tiếp với mong muốn đánh bại bạn

147

Ứng dụng Heuristic trong các trò chơi

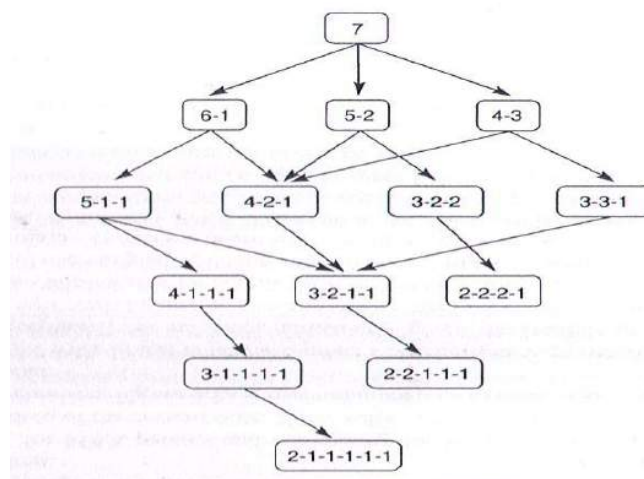
■ Trò chơi Nim:

- Một số token (đồng xu, mảnh gỗ...) được đặt giữa 2 đối thủ.
- Ở mỗi lượt đi, người chơi phải chia các token **thành 2 phần (không rỗng) với số lượng khác nhau**. VD: 6 token có thể được chia thành 5 - 1, 4 - 2 (trường hợp 3 - 3 là không hợp lệ)
- Khi các token không thể được chia một cách hợp lệ ở lượt chơi kế tiếp, người chơi thuộc về lượt đi đó sẽ thua cuộc.

148

Giải thuật minimax

■ Không gian trạng thái trò chơi Nim



149

Giải thuật minimax

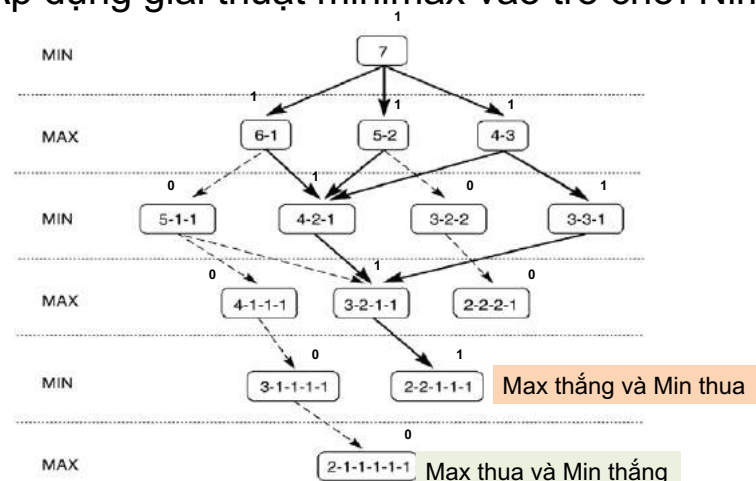
■ Giải thuật minimax:

- Một đấu thủ trong trò chơi được gọi là **MIN** và đấu thủ còn lại là **MAX**. Max đại diện cho người chơi luôn muốn chiến thắng, Min đại diện cho người chơi cố gắng cho người max giành số điểm càng thấp càng tốt
- Giá trị của nút lá:
 - 1 nếu là MAX thắng,
 - 0 nếu là MIN thắng (MAX thua)
- Minimax sẽ truyền các giá trị này lên cao dần trên đồ thị, qua các nút cha kế tiếp theo các luật sau:
 - Nếu trạng thái cha là **MAX**, gán cho nó giá trị **lớn nhất** có trong các trạng thái con.
 - Nếu trạng thái cha là **MIN**, gán cho nó giá trị **nhỏ nhất** có trong các trạng thái con.

150

Giải thuật minimax

■ Áp dụng giải thuật minimax vào trò chơi Nim



151

Giải thuật minimax

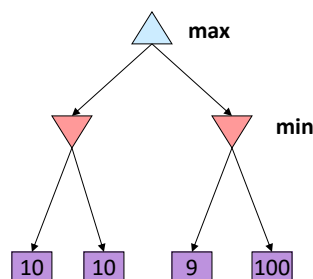
■ Minimax đến độ sâu lớp cố định

- Đối với các trò chơi phức tạp, đồ thị KGTT có khả năng không được triển khai đến các nút lá
- KGTT chỉ có thể được triển khai đến một số mức xác định (tùy vào tiềm năng về thời gian và bộ nhớ) => tính trước n nước đi
- Vì các nút lá của đồ thị con này không phải là trạng thái kết thúc của trò chơi => không xác định được các giá trị thắng - thua (1 hoặc 0)
- Cần sử dụng một hàm đánh giá Heuristic nào đó
- Giá trị nút lá là các giá trị Heuristic đạt được sau n nước đi kể từ nút xuất phát (tùy vào hàm đánh giá Heuristic cụ thể)
- Các giá trị này sẽ được truyền ngược về nút gốc tương tự như trong trò chơi Nim, và chỉ là giá trị của trạng thái tốt nhất có thể.

152

Minimax với độ sâu lớp cố định

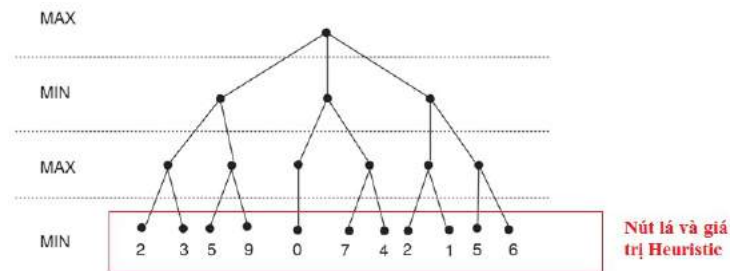
- Các **nút lá** được gán các giá trị **heuristic**
- Còn giá trị tại các nút trong là các giá trị nhận được dựa trên giải thuật Minimax



153

Minimax với độ sâu lớp cố định

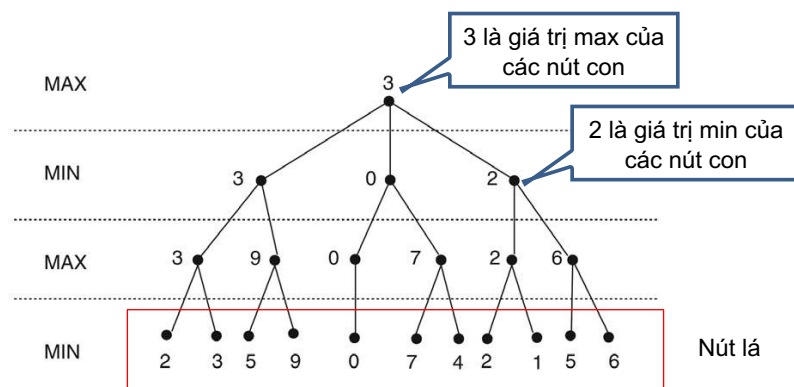
- Ví dụ cho KGTT giả định và giá trị Heuristic của các nút lá. Hãy tính giá trị của các nút còn lại



154

Giải thuật minimax

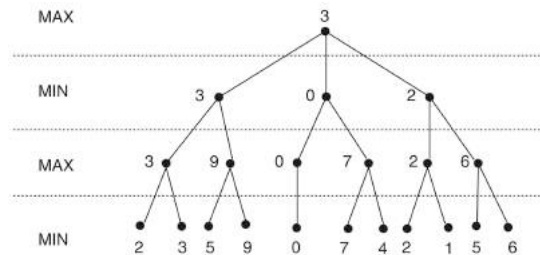
- Ví dụ:



155

Minimax với độ sâu lớp cố định

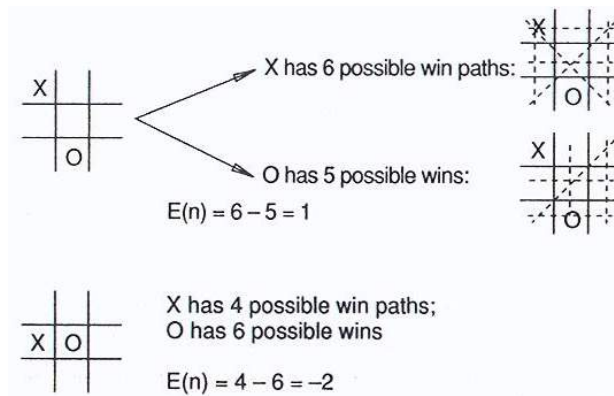
- Minimax đối với một KGTT giả định.



- 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

156

Heuristic trong trò chơi tic-tac-toe



Hàm Heuristic: $E(n) = M(n) - O(n)$

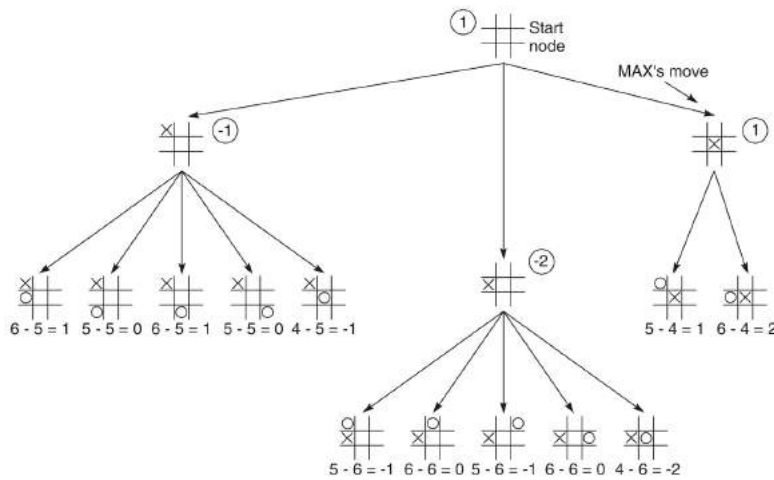
Trong đó: $M(n)$ là tổng số đường thắng có thể của tôi

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

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

157

Minimax 2 lớp được áp dụng vào nước đi mở đầu trong tic-tac-toe



Trích từ Nilsson (1971).

Hàm Heuristic:

$$\mathbf{E}(\mathbf{n}) = \mathbf{M}(\mathbf{n}) - \mathbf{O}(\mathbf{n})$$

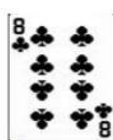
Trong đó: $M(n)$ là tổng số đường thẳng có thể của tôi

$O(n)$ là tổng số đường thẳng có thể của đối thủ

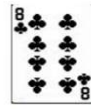
$E(n)$ là trị số đánh giá
tổng cộng cho trạng thái n

Bài toán bắt lá bài

- Mỗi người chơi chọn 1 lá bài trong 4 lá bài cho trước
- Với 4 lá bài được cho sẵn, mỗi người sẽ chọn 2 lá bài
- Cộng điểm 2 lá bài đã chọn, nếu người chơi nào có tổng điểm là chẵn và cao nhất sẽ thắng



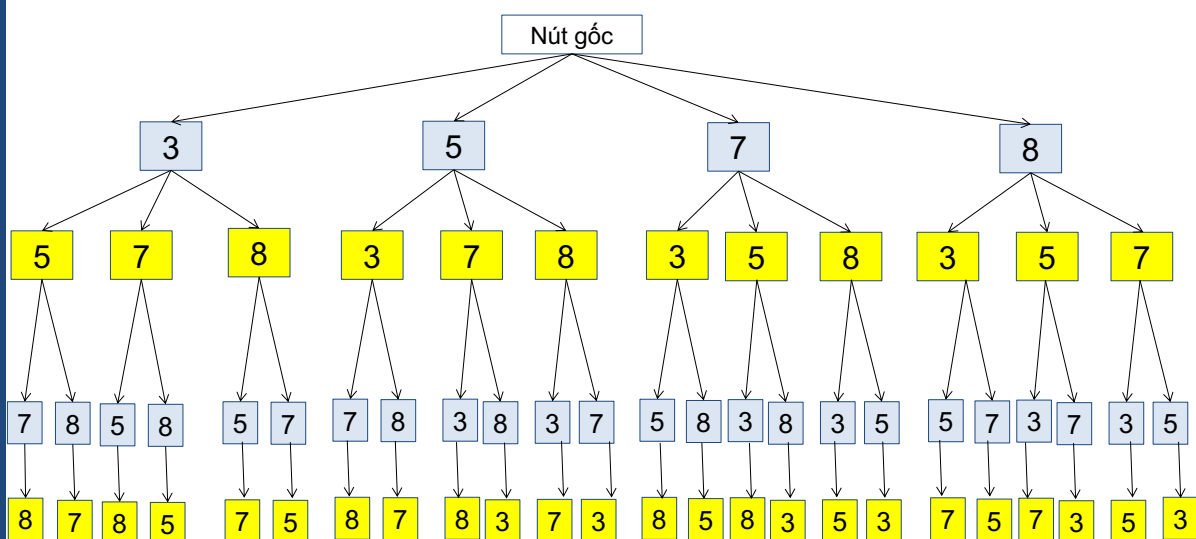
Bài toán bắt lá bài



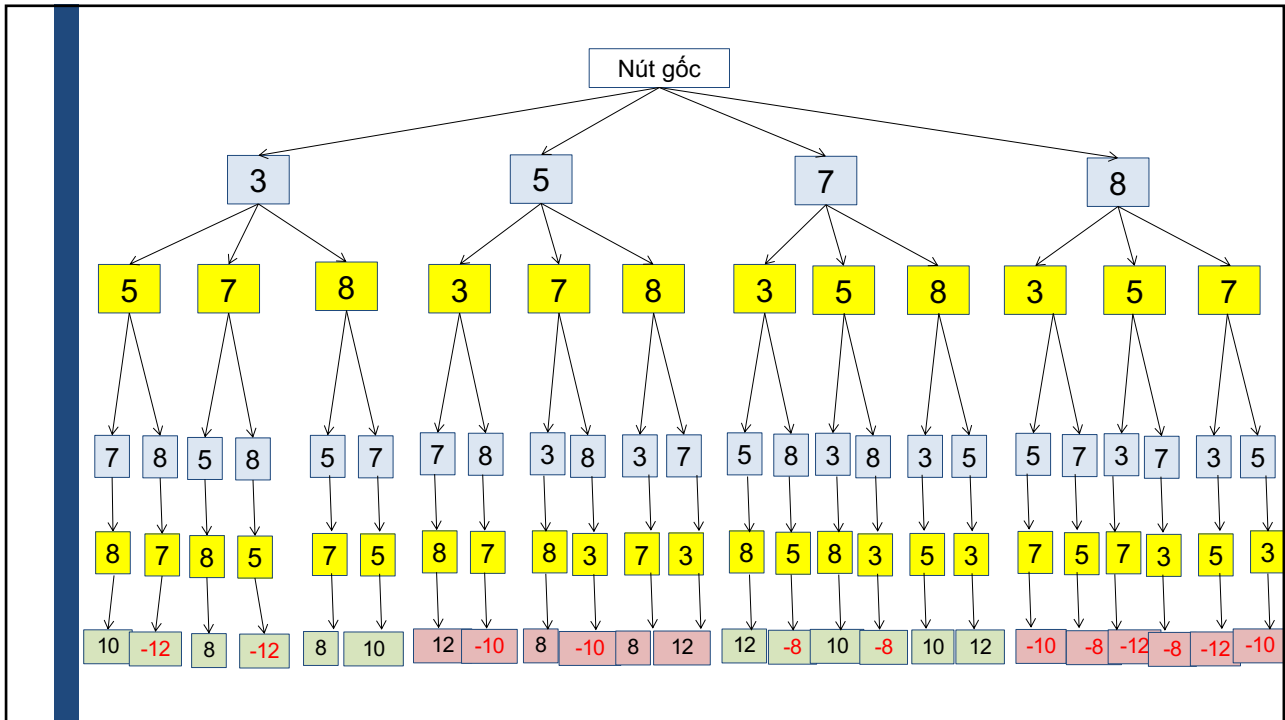
- Để có thể thắng trong trò chơi trên
 - Vẽ toàn bộ không gian trạng thái
 - Gắn điểm số vào mỗi trạng thái kết thúc
 - Sử dụng giải thuật Minimax để cập nhật điểm số từ nút lá đến nút gốc
 - Chọn đường đi có điểm số lớn nhất

160

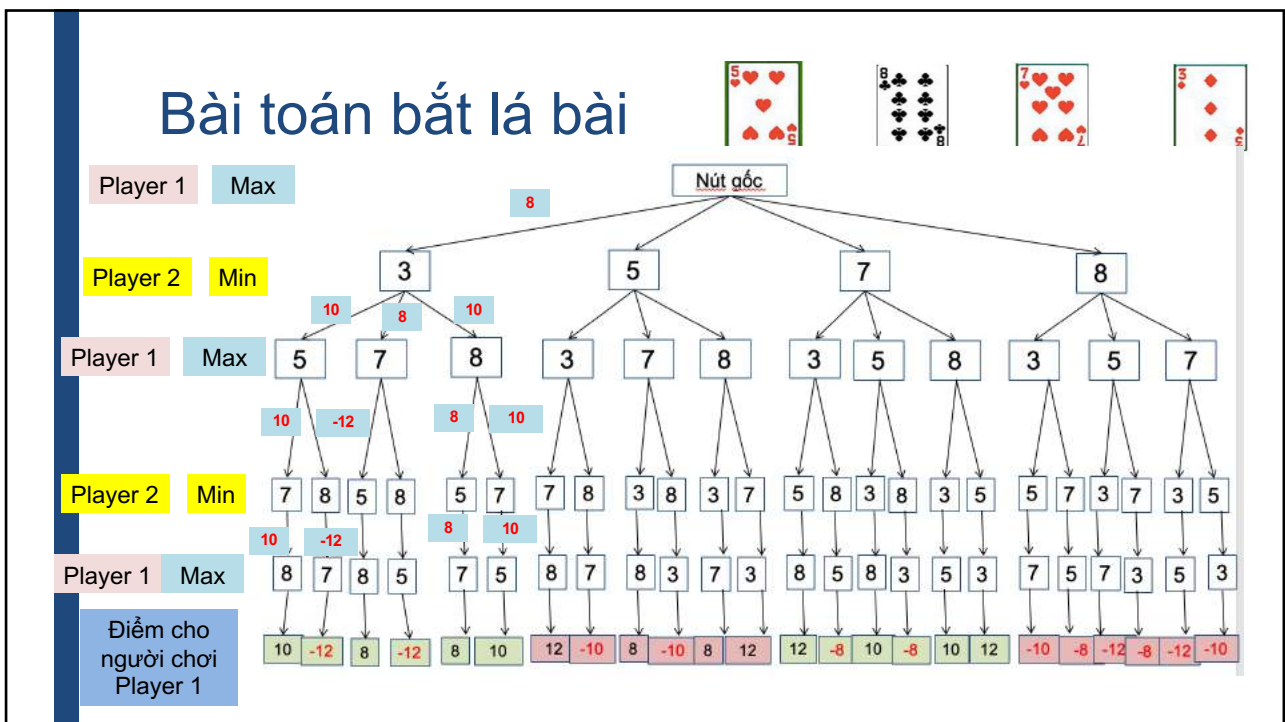
Bài toán bắt lá bài



161



162



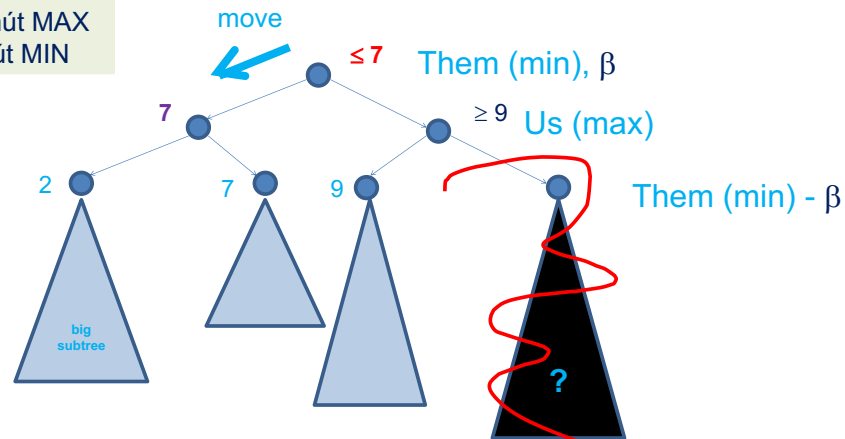
163

165



β -cut

α gắn với các nút MAX
 β gắn với các nút MIN



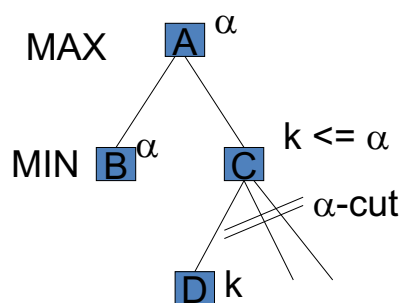
Nút MIN có $\beta \leq \alpha$ của nút cha MAX bất kỳ (α -cut)
Nút MAX có $\alpha \geq \beta$ của nút cha MIN bất kỳ (β -cut)

166

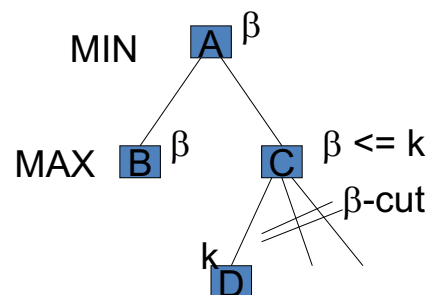
166

Thủ tục cắt tỉa alpha – beta

■ α -cut



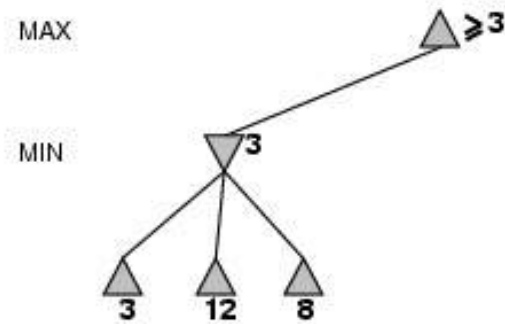
β -cut



Nút MIN có $\beta \leq \alpha$ của nút cha MAX bất kỳ (α -cut)
Nút MAX có $\alpha \geq \beta$ của nút cha MIN bất kỳ (β -cut)

167

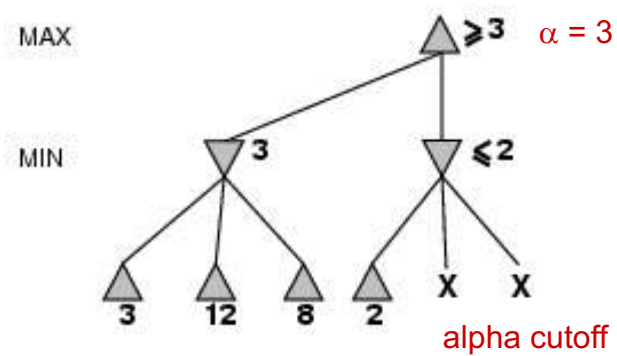
α - β pruning example



168

168

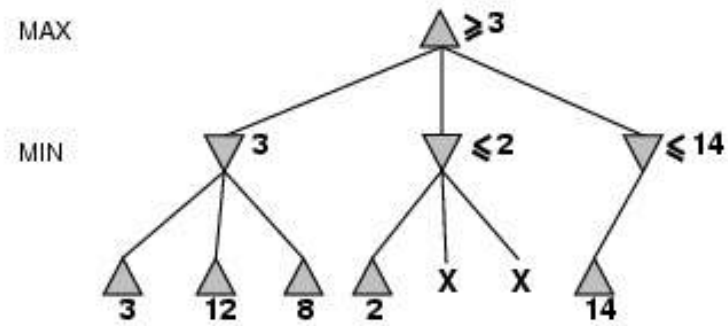
α - β pruning example



169

169

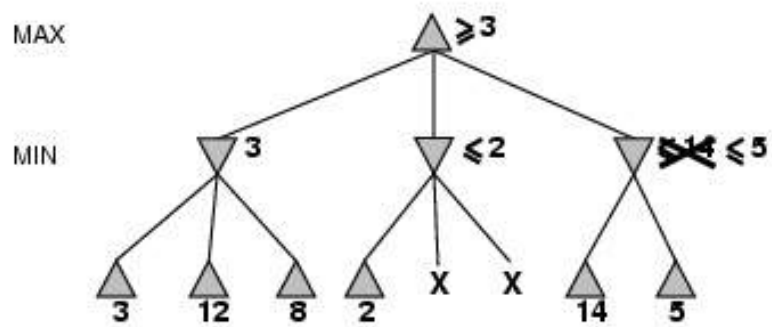
α - β pruning example



170

170

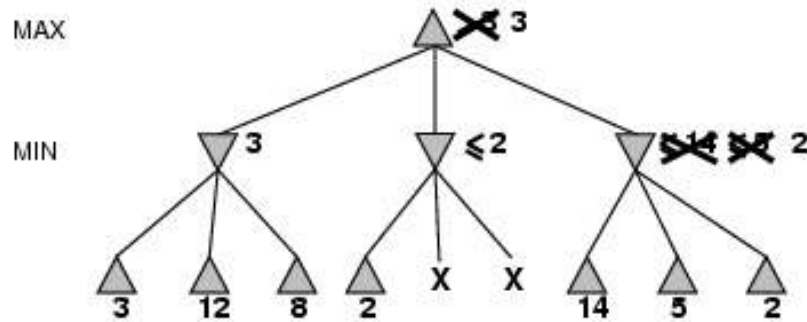
α - β pruning example



171

171

α - β pruning example

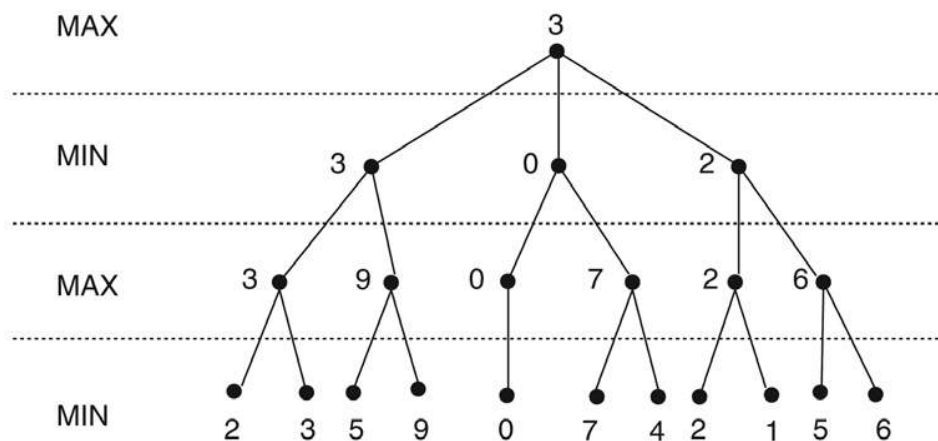


172

172

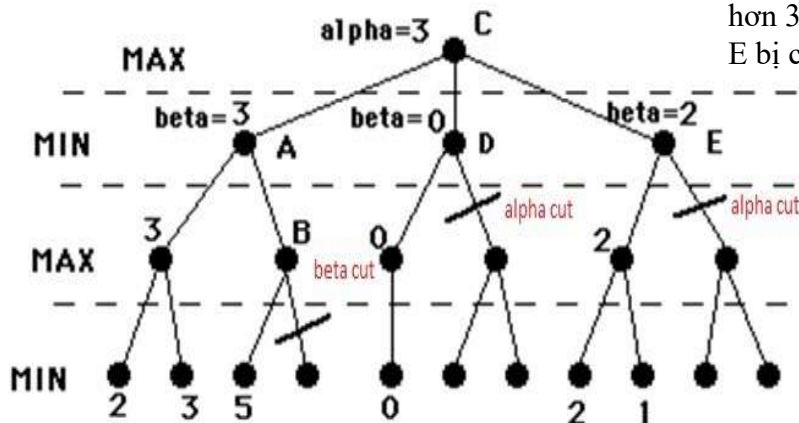
Nút MIN có $\beta \leq \alpha$ của nút cha MAX bất kỳ (α -cut)
 Nút MAX có $\alpha \geq \beta$ của nút cha MIN bất kỳ (β -cut)

■ Minimax đối với một KGTT giả định:



173

Thủ tục cắt tỉa alpha – beta



A có $\beta = 3$ (Trị nút A sẽ không lớn hơn 3)
 B bị cắt tỉa β , vì $5 > 3$
 C có $\alpha = 3$ (Trị nút C sẽ không nhỏ hơn 3)
 D bị cắt tỉa α , vì $0 < 3$
 E bị cắt tỉa α , vì $2 < 3$ Trị nút C là 3

174

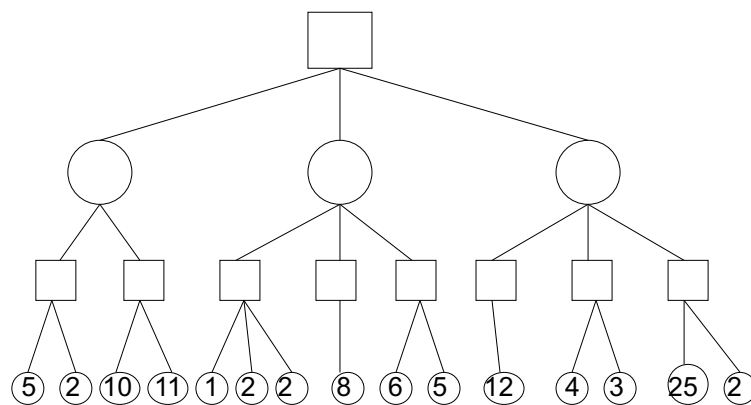
Alpha-Beta Pruning

max

min

max

eval



Nút MIN có $\beta \leq \alpha$ của nút cha MAX bất kỳ (α -cut)
 Nút MAX có $\alpha \geq \beta$ của nút cha MIN bất kỳ (β -cut)

175

175

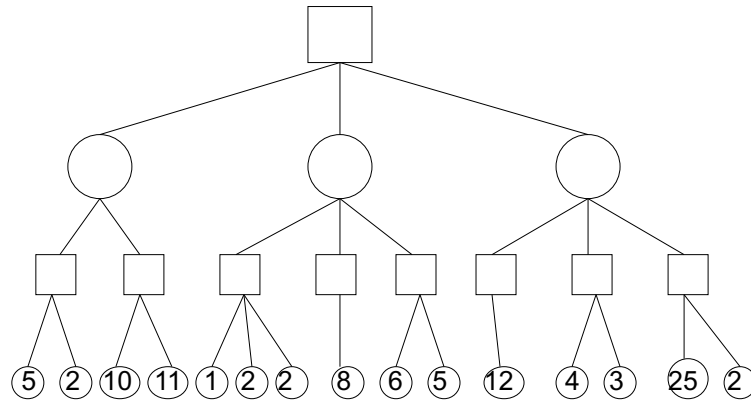
Alpha-Beta Pruning

max

min

max

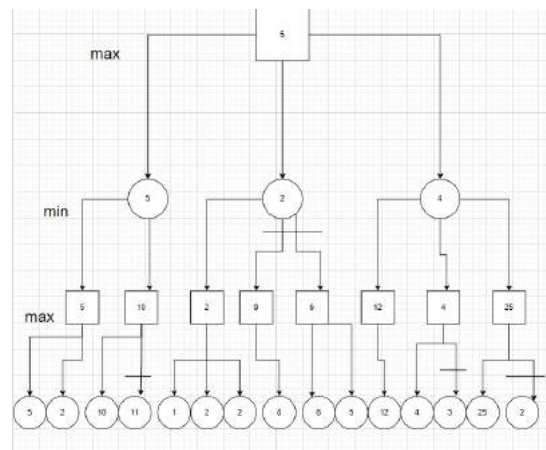
eval



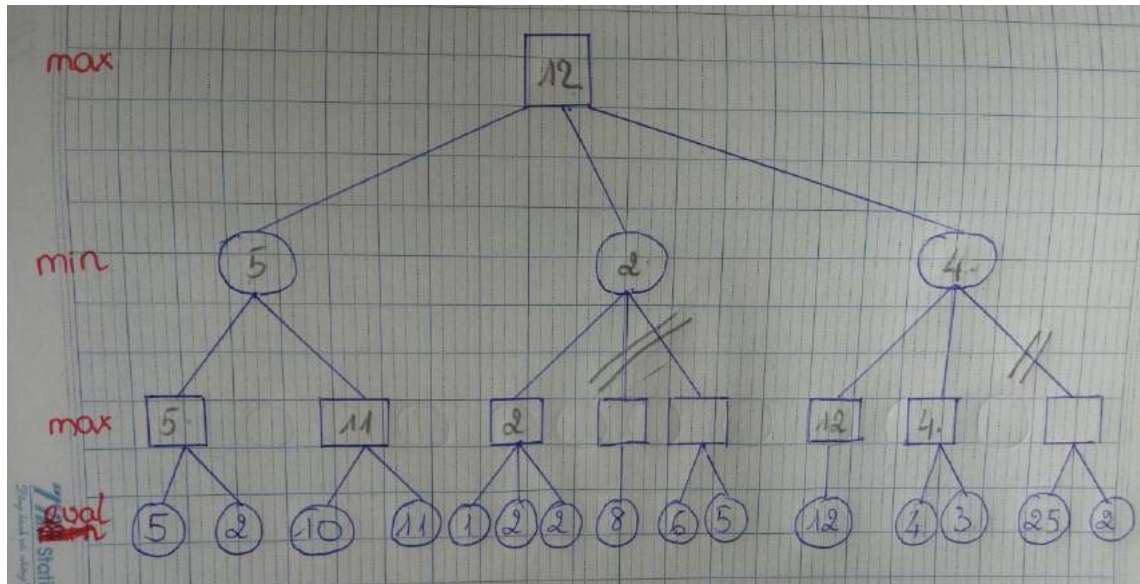
Nút MIN có $\beta \leq \alpha$ của nút cha MAX bất kỳ (α -cut)
 Nút MAX có $\alpha \geq \beta$ của nút cha MIN bất kỳ (β -cut)

180

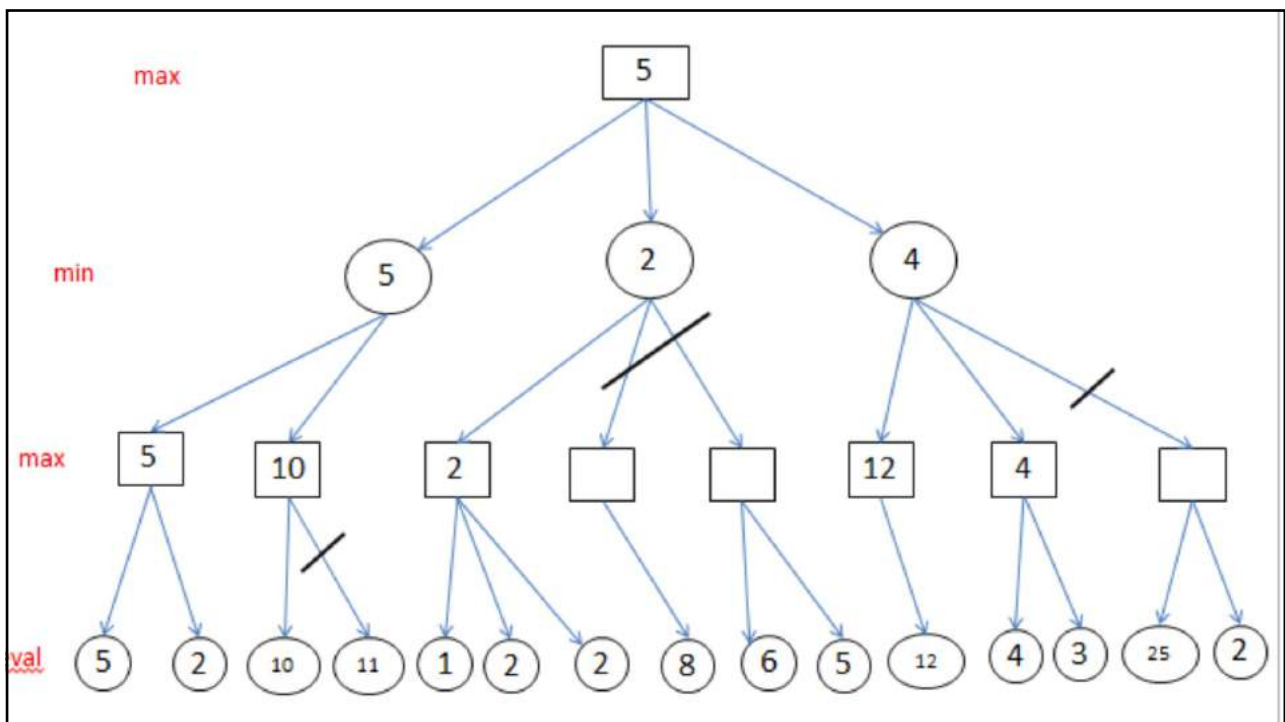
180



181

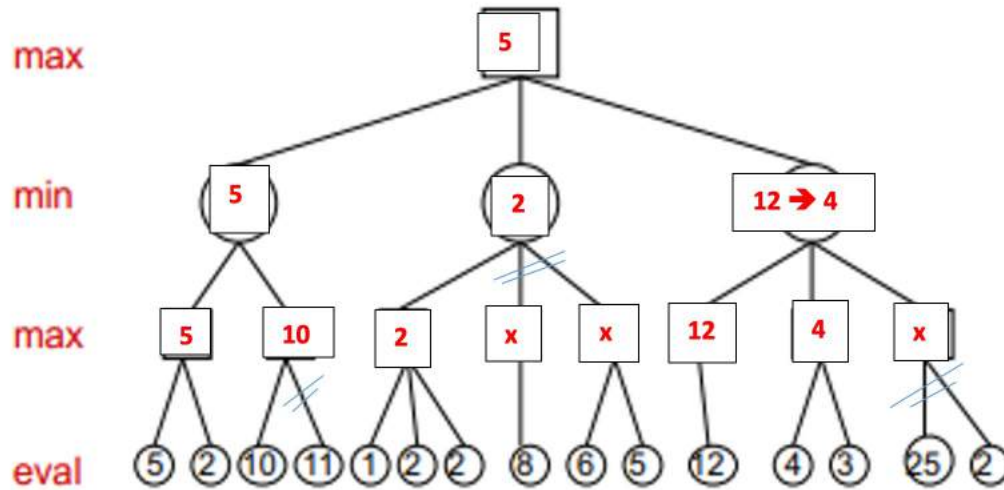


182



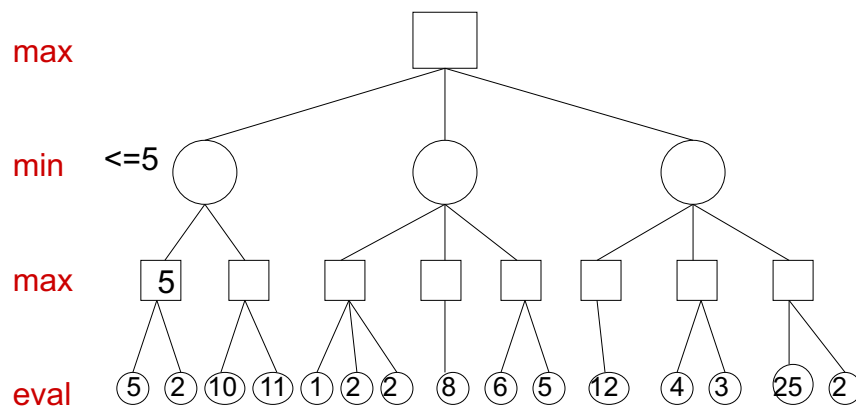
183

Alpha-Beta Pruning



184

Alpha-Beta Pruning

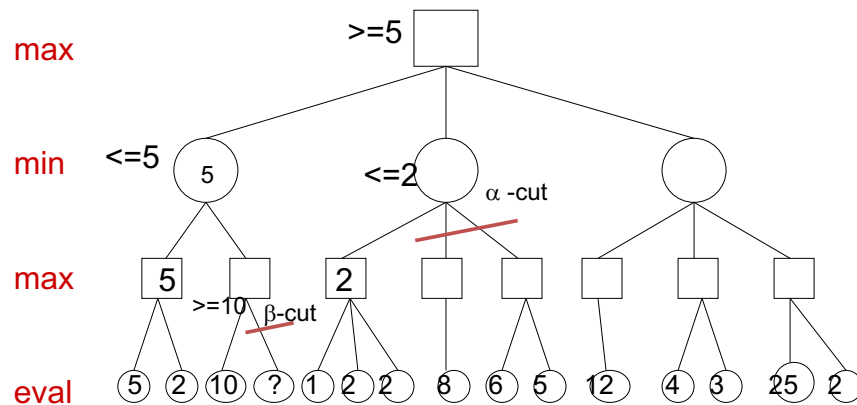


Nút MIN có $\beta \leq \alpha$ của nút cha MAX bất kỳ (α -cut)
 Nút MAX có $\alpha \geq \beta$ của nút cha MIN bất kỳ (β -cut)

185

185

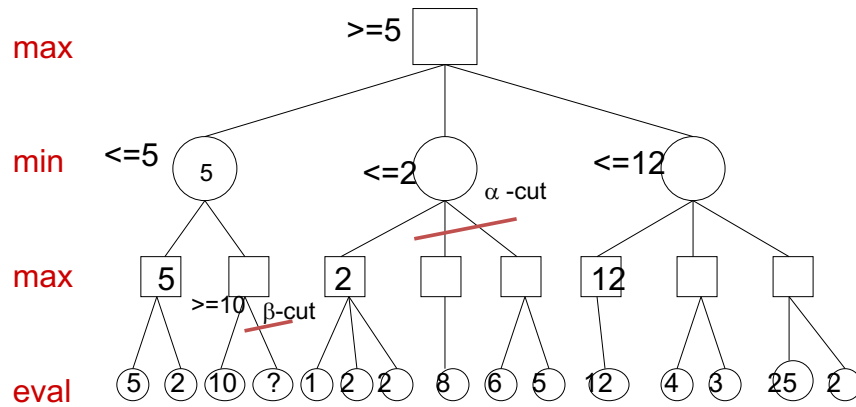
Alpha-Beta Pruning



Nút MIN có $\beta \leq \alpha$ của nút cha MAX bất kỳ (α -cut)
 Nút MAX có $\alpha \geq \beta$ của nút cha MIN bất kỳ (β -cut)

187

Alpha-Beta Pruning

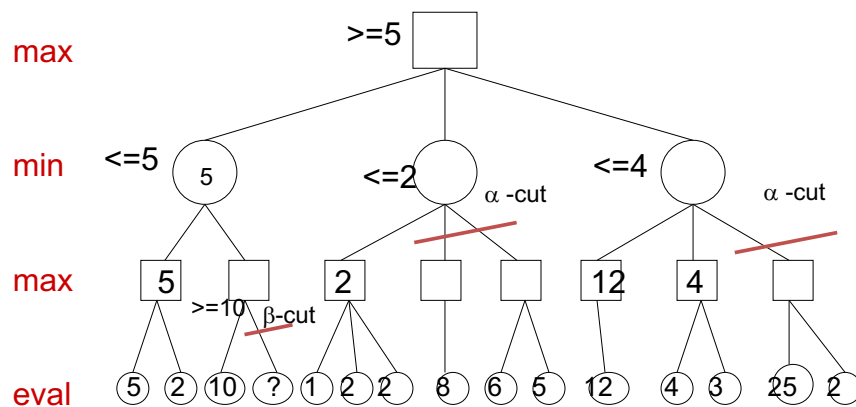


Nút MIN có $\beta \leq \alpha$ của nút cha MAX bất kỳ (α -cut)
 Nút MAX có $\alpha \geq \beta$ của nút cha MIN bất kỳ (β -cut)

188

188

Alpha-Beta Pruning



Nút MIN có $\beta \leq \alpha$ của nút cha MAX bất kỳ (α -cut)
 Nút MAX có $\alpha \geq \beta$ của nút cha MIN bất kỳ (β -cut)

189

189

Bài toán thỏa mãn ràng buộc

190

190

Một số bài toán thỏa mãn ràng buộc thực tế

- Xét bản đồ các bang của nước Úc như hình bên. Cần tô màu các bang với ba màu red, green, blue sao cho hai bang cạnh nhau được tô màu khác nhau



				8			4
	8	4		1	6		
			5			1	
1		3	8			9	
6		8				4	3
		2			9	5	1
		7			2		
			7	8		2	6
2			3				

S E N D
M O R E
M O N E Y

191

191

Một số bài toán thoả thực tế

- Các bài toán giao nhiệm vụ
 - Ví dụ: Giáo viên nào dạy lớp nào?
- Các bài toán lập thời khóa (gian) biểu
 - Ví dụ: Lớp học nào được dạy vào thời gian nào và ở đâu?
- Các bài toán lập lịch vận tải (giao hàng) của các công ty
- Các bài toán lập lịch sản xuất của các nhà máy

192

192

Bài toán thoả mãn ràng buộc (Constraint satisfaction problem)

- Bài toán thoả mãn ràng buộc (CSP): sử dụng phương pháp biểu diễn có cấu trúc để biểu diễn các trạng thái, mỗi trạng thái là một tập biến, mỗi biến có một giá trị
- Bài toán được giải quyết khi mỗi biến đều được gán trị thoả mãn tất cả ràng buộc
- Một số bài toán được giải quyết nhanh chóng khi mô hình hóa về CSP trong khi không giải quyết được bằng phương pháp tìm kiếm trong không gian trạng thái

193

193

Bài toán thỏa mãn ràng buộc

- Một bài toán thỏa mãn ràng buộc gồm ba thành phần: X , D , và C trong đó:
 - X : tập hợp các biến $\{X_1, X_2, \dots, X_n\}$
 - D : tập hợp các miền giá trị $\{D_1, D_2, \dots, D_n\}$, $D_i = \{v_1, v_2, \dots, v_k\}$ gán vào biến X_i tương ứng
 - C : tập hợp các ràng buộc, C_i là một cặp $\langle \text{scope}, \text{rel} \rangle$ với scope là một bộ biến tham gia vào quan hệ rel (biểu diễn như một danh sách các bộ giá trị tường minh thỏa mãn ràng buộc, hoặc dưới dạng trừu tượng phép toán)
- Bài toán CSP được giải quyết khi tất cả các biến đều được gán trị hợp lệ

194

194

Bài toán thỏa mãn ràng buộc

- Giải quyết bài toán CSP bằng cách sử dụng lan truyền ràng buộc
 - Dùng các ràng buộc để giảm số giá trị hợp lệ cho một biến
 - Kết quả các giá trị này lại có thể làm giảm các giá trị hợp lệ cho biến khác...
- Ý tưởng chính của lan truyền ràng buộc là tính nhất quán cục bộ:
 - Mỗi biến là một nút trong đồ thị
 - Mỗi ràng buộc nhị phân (trên 2 biến) như một cung thì quá trình ép buộc tính nhất quán cục bộ trong mỗi phần của đồ thị tạo ra các giá trị bất hợp lệ cần phải loại bỏ

195

195

Ví dụ: Bài toán tô màu bản đồ

- Xét bản đồ các bang của nước Úc như hình bên. Cần tô màu các bang với ba màu red, green, blue sao cho hai bang cạnh nhau được tô màu khác nhau
- **Variables** WA, NT, Q, NSW, V, SA, T
- **Domains** $D_i = \{\text{red, green, blue}\}$
- **Ràng buộc:** các vùng lân cận có màu khác nhau
- e.g., $WA \neq NT$, or (WA, NT) in $\{(\text{red, green}), (\text{red, blue}), (\text{green, red}), (\text{green, blue}), (\text{blue, red}), (\text{blue, green})\}$

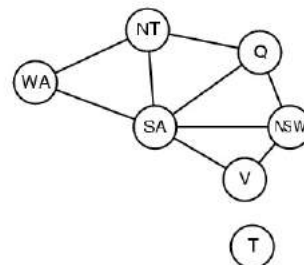


196

196

Ví dụ: Bài toán tô màu bản đồ

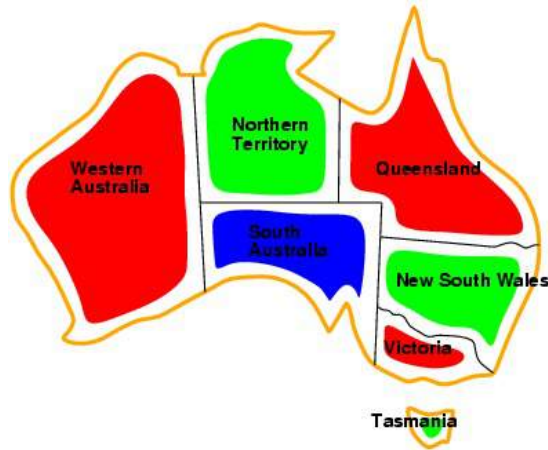
- **Đồ thị ràng buộc:** các nút của đồ thị tương ứng với các biến, các cung là các ràng buộc



197

197

Ví dụ: Bài toán tô màu bản đồ



- **Giải pháp:** WA = red, NT = green, Q = red, NSW = green, V = red, SA = blue, T = green

198

198

Thỏa mãn ràng buộc (Constraint satisfaction)

- *Trạng thái khởi đầu chứa các ràng buộc được cho trong mô tả vấn đề*
- *Một trạng thái đích là trạng thái đã bị ràng buộc « đủ », « đủ » được định nghĩa tùy thuộc vấn đề. Ví dụ, trong câu đố « mật mã số học » « đủ » có nghĩa là mỗi chữ được gán một chữ số duy nhất.*
- Thỏa mãn ràng buộc là quá trình hai bước:
 - *Đầu tiên, các ràng buộc được phát hiện và được lan truyền xa như có thể.*
 - *Sau đó, nếu vẫn chưa có lời giải, bắt đầu tìm kiếm. Đoán một sự kiện, thêm vào ràng buộc mới, lan truyền ràng buộc ...*

199

199

Thỏa mãn ràng buộc (Constraint satisfaction)

1. Lan truyền các ràng buộc sẵn có: Đặt OPEN = tập chứa các đối tượng cần phải gán trị (trong lời giải). Tiến hành các bước sau đến tận khi gặp mâu thuẫn hoặc OPEN rỗng:
 - a) Chọn một đối tượng X trong OPEN. Tăng cường tập các ràng buộc trên X
 - b) Nếu tập này khác với tập đã được gán cho X trong lần kiểm tra trước hoặc X lần đầu tiên được kiểm tra, thêm vào OPEN tất cả các đối tượng chia sẻ các ràng buộc với X
 - c) Xóa X khỏi OPEN
2. Nếu hợp các ràng buộc được phát hiện ở trên xác định lời giải, thông báo lời giải và thoát
3. Nếu hợp các ràng buộc được phát hiện ở trên xác định một mâu thuẫn, thông báo thất bại

200

200

Thỏa mãn ràng buộc (Constraint satisfaction)

4. Nếu 2. và 3. không xảy ra, Lặp lại đến tận khi tìm thấy một lời giải hoặc tất cả các lời giải có thể bị loại bỏ:
 - a) Chọn một đối tượng chưa được xác định giá trị, chọn một phương pháp tăng cường ràng buộc trên đối tượng
 - b) Gọi đệ quy thỏa mãn ràng buộc với tập hiện hành các ràng buộc được tăng cường thêm qua bước a)

201

201

Thỏa mãn ràng buộc (Constraint satisfaction)

- Nhiều vấn đề AI có thể được xem như vấn đề thỏa mãn ràng buộc:
Đích là phát hiện trạng thái thỏa mãn một tập các ràng buộc đã cho.

VD: Số học mật mã (CryptArithmetic)

$$\begin{array}{r} \text{S E N D} \\ + \text{M O R E} \\ \hline \text{M O N E Y} \end{array}$$

Ràng buộc:

- Mỗi chữ tương ứng với một chữ số 0 .. 9
 - Các chữ khác nhau tương ứng với các chữ số khác nhau
 - Các chữ số làm thỏa mãn phép cộng
- Sự thỏa mãn ràng buộc thường làm giảm khối lượng tìm kiếm
 - Thỏa mãn ràng buộc là một thủ tục tìm kiếm hoạt động trong không gian các tập ràng buộc:

202

202

Thỏa mãn ràng buộc (Constraint satisfaction)

$$\begin{array}{r} \text{S E N D} \\ + \text{M O R E} \\ \hline \text{M O N E Y} \end{array}$$

Số học mật mã (CryptArithmetic)

Ràng buộc:

- Mỗi chữ tương ứng với một chữ số 0 .. 9
- Các chữ khác nhau tương ứng với các chữ số khác nhau
- Các chữ số làm thỏa mãn phép cộng

Ký hiệu các số mang của cột i là C_i ($i = 1, 2, 3, 4$)

$C_i = 0$ hoặc 1

203

203

Thỏa mãn ràng buộc (Constraint satisfaction)

- Các quy tắc để lan truyền ràng buộc sinh ra các ràng buộc sau:

- $M = 1$ ($S + M + C_3 \leq 19$)
 - $S = 8$ hoặc 9 ($S + M + C_3 > 9, C_3 \leq 1, M = 1$)
 - $O = 0$ ($S + M + C_3 = 10 + O$, với $M = 1, S \leq 9, C_3 \leq 1$ nên $O \leq 1$)
 - $S = 9, C_3 = 0$ (vì nếu $C_3 = 1$ thì $E = 9$ và $N = 0$ mà $O = 0$)
 - $N = E + 1$ (vì $N = E + O + C_2 = E + C_2$ nên $N = E + 1$)
 - $C_2 = 1$
 - $C_1 = 1$ và $R = 8$ (nếu $C_1 = 0 \Rightarrow R = 9$)
 - D từ 7 và E từ 5 (vì $D + E \geq 12$)
- Không có thêm ràng buộc !

S	E	N	D
M	O	R	E
M	O	N	E
Y			

204

204

Thỏa mãn ràng buộc (Constraint satisfaction)

- Để tiến triển thực hiện « phương pháp đoán »:

$E = 5;$

.....

- Kết quả:

$S = 9, E = 5, N = 6, D = 7, M = 1, O = 0, R = 8, Y = 2$

9	5	6	7
+	1	0	8
1	0	6	5

205

205

Cần xếp lịch dạy của 3 giảng viên cho 5 lớp học có lịch gần nhau, bao gồm:

- Lớp 1: Phân tích thiết kế thuật toán, lịch học 8:00 - 09:00AM
- Lớp 2: Nhập môn AI, lịch học 8:30 - 09:30AM
- Lớp 3: Cơ sở dữ liệu, lịch học 9:00 - 10:00AM
- Lớp 4: Hệ điều hành, lịch học 9:00 - 10:00AM
- Lớp 5: Nguyên lý máy học, lịch học 9:30 - 10:30AM

Biết rằng giảng viên chỉ có thể dạy 1 lớp tại một thời điểm và thông tin về giảng viên lần lượt là:

- GV A có thể dạy lớp 3 và 4
- GV B có thể dạy lớp 2, 3, 4, và 5
- GV C có thể dạy lớp 1, 2, 3, 4, 5

Hãy giải quyết vấn đề trên sử dụng giải thuật thỏa mãn ràng buộc với gợi ý các biến lần lượt là các lớp học, xác định miền giá trị cho tập biến và các ràng buộc cần thiết

206

Sử dụng 3 màu: R (red), G(green), B(blue) để tô màu bản đồ bên dưới với yêu cầu các vùng lân cận phải sử dụng các màu khác nhau



1. Hãy xây dựng đồ thị ràng buộc tương ứng với bản đồ đã cho
2. Giả sử gán AZ là R, xác định 1 giải pháp tô màu có thể cho các đỉnh còn lại

207