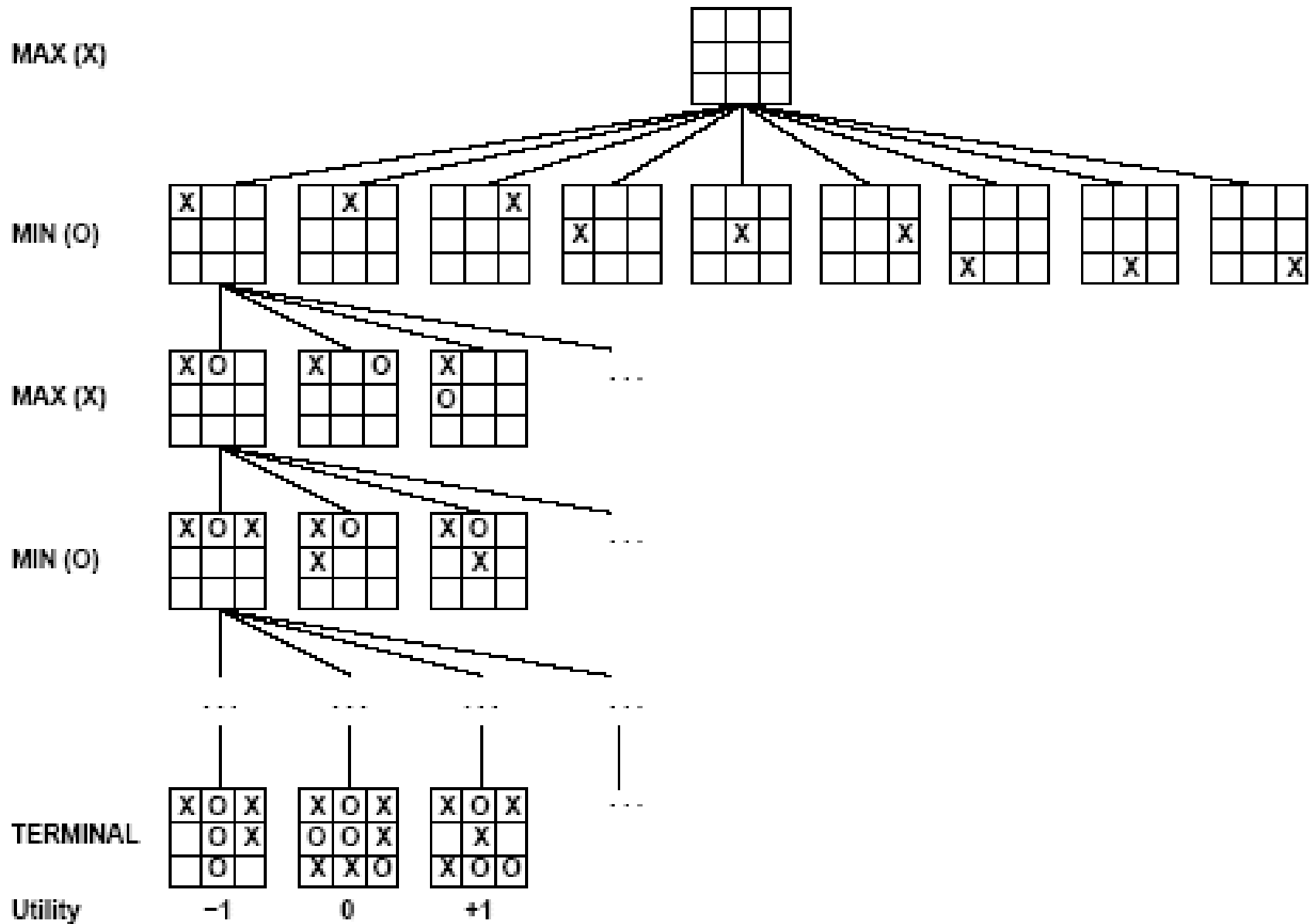


- Adversário imprevisível
 - a solução é uma **estratégia** que especifica uma jogada para cada resposta possível do adversário
- Limitações de tempo
 - na impossibilidade de encontrar a solução ótima, é necessário aproximar
- História
 - Computador considera possíveis linhas de jogo (Babbage, 1846)
 - Algoritmo para jogada perfeita (Zermelo, 1912; Von neumann, 1944)
 - Horizonte finito, avaliação aproximada (Zuse, 1945; Wiener, 1948, Shannon, 1950)
 - Primeiro programa de xadrez (Turing, 1951)
 - Aprendizagem (machine learning) para aumentar a qualidade da avaliação (Samuel, 1952-1957)
 - Corte (pruning) para permitir pesquisa a maior profundidade (McCarthy, 1956)

Tipos de jogos

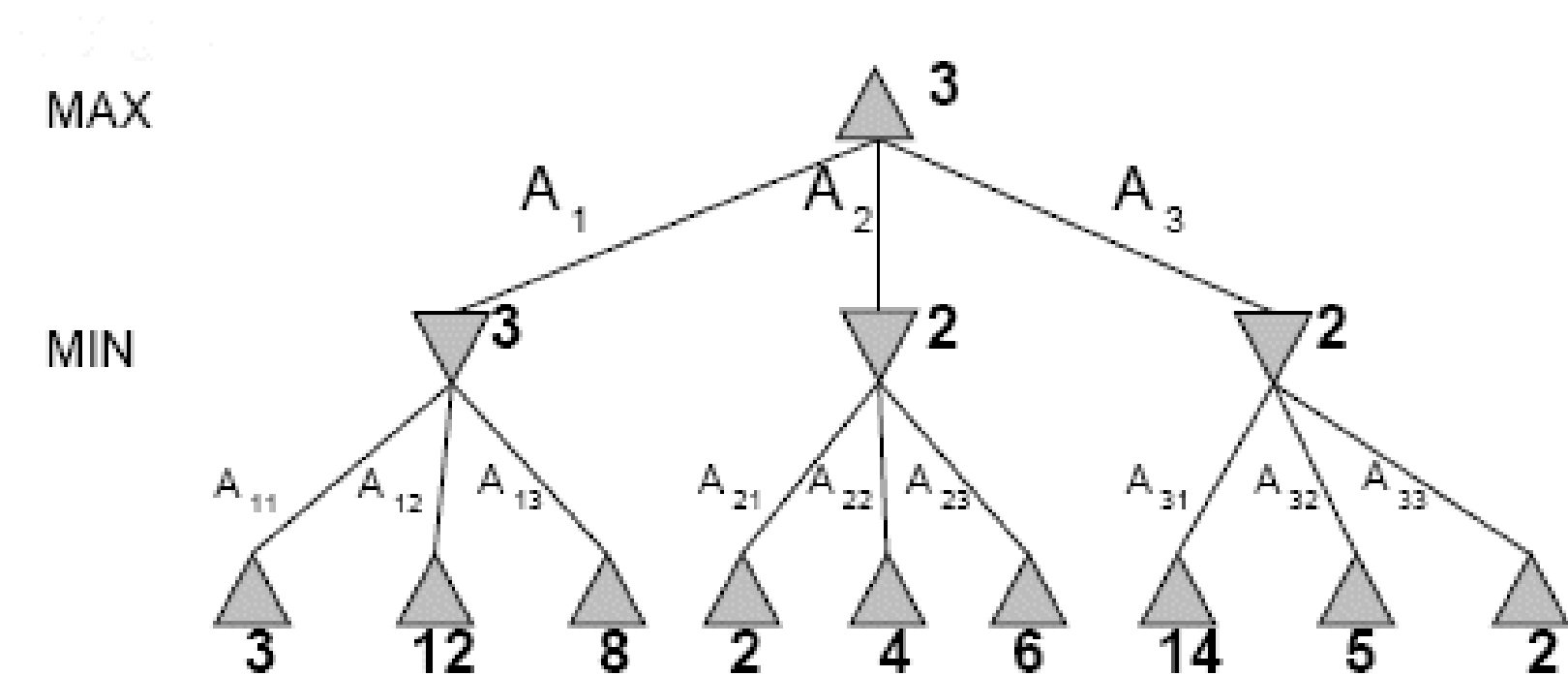
	determinísticos	de azar
informação perfeita	xadrez, damas, go, jogo do galo	gamão, monopólio
informação imperfeita		bridge, poker, bisca, scrabble, guerra

Jogo do galo



- Estratégia perfeita para jogos **determinísticos** com **informação perfeita**
 - Assume que os jogadores usam uma estratégia racional
- Ideia:
 - escolher a jogada com maior valor **minimax**
- Valor: o nosso ganho
 - Quando somos nós a jogar: escolhe-se a jogada que proporciona o valor mais elevado (**MAX**)
 - Quando é o adversário a jogar, assume-se que vai jogar bem (do ponto de vista dele), ou seja, vai escolher a jogada que acarreta valor mais baixo para nós (**MIN**)

- Exemplo: jogo de duas jogadas:



função DECISÃO-MINIMAX(*estado*) retorna *ação*
 $v \leftarrow \text{VALOR-MAX}(\textit{estado})$ é máximo
 retorna *ação com valor v*

função VALOR-MAX(*estado*) retorna um valor de utilidade
 se TESTE-TERMINAL(*estado*) então
 retorna UTILIDADE(*estado*)
 senão
 retorna o valor mais elevado de VALOR-MIN(SUCESSORES(*estado*))

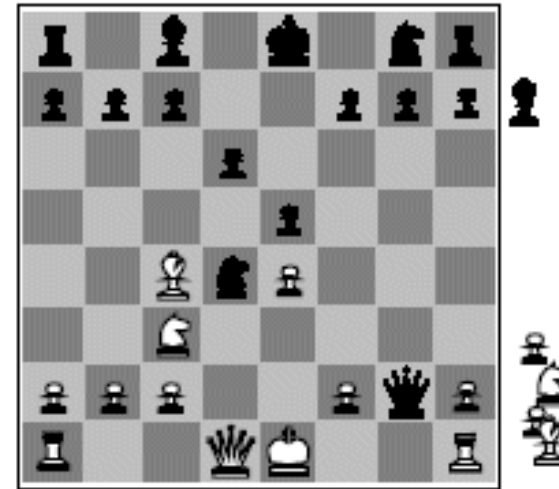
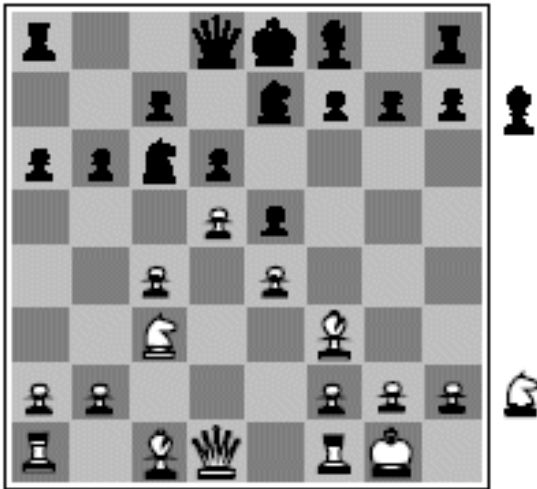
função VALOR-MIN(*estado*) retorna um valor de utilidade
 se TESTE-TERMINAL(*estado*) então
 retorna UTILIDADE(*estado*)
 senão
 retorna o valor mais baixo de VALOR-MAX(SUCESSORES(*estado*))

- Completo ??
 - Só se a árvore é finita
- Ótimo ??
 - Sim, contra um adversário ótimo (racional). Senão?...
- Tempo ??
 - $O(b^m)$
- Espaço ??
 - $O(b \cdot m)$ -- pesquisa em profundidade
- Xadrez:
 - $b = 35, m = 100$
 - uma solução exata é completamente irrealizável
 - tempo = $O(35^{100})$

- Exemplo:
 - 100 segundos para realizar uma jogada
 - exploram-se 10 000 nós por segundo
 - então podem-se explorar 1 000 000 de nós por jogada
- Solução tradicional
 - Corte (cutoff)
 - por exemplo, limitar a profundidade da pesquisa
 - Função de avaliação
 - **estimar** a qualidade da posição (usando por exemplo uma função heurística)

Função de avaliação -- Exemplo

- Qual é a melhor função de avaliação?
 - Como medimos a qualidade de cada uma das posições seguintes?



- Normalmente a função de avaliação é uma função linear pesada de vários parâmetros

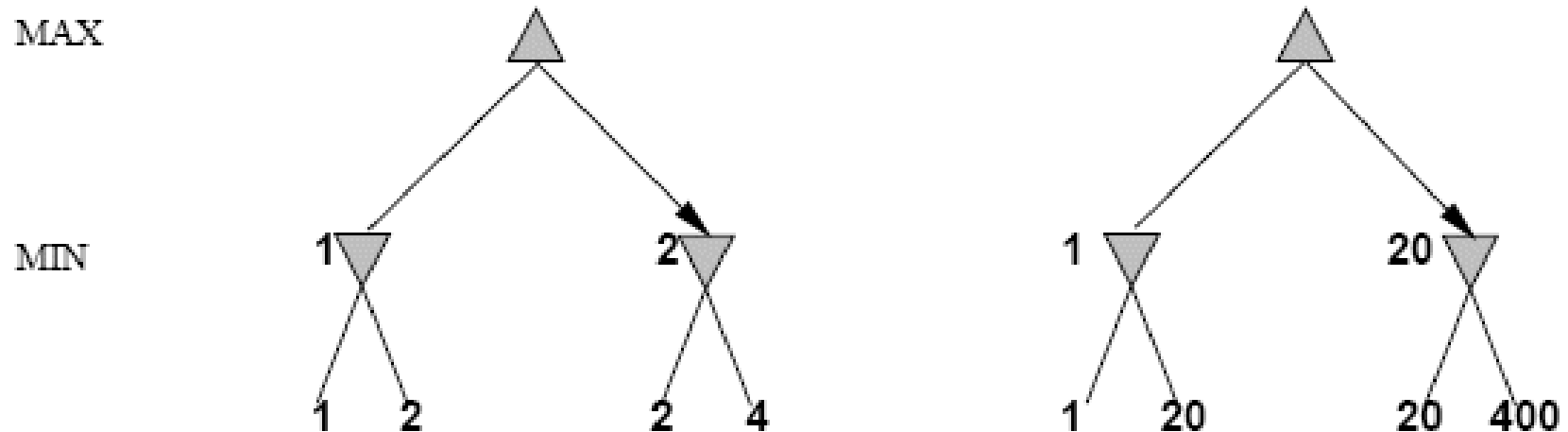
- Exemplo do xadrez:

$$\text{Avaliação}(s) = w_1 f_1(s) + w_2 f_2(s) + \dots + w_n f_n(s)$$

$$w_1 = 9, \text{ e } f_1(s) = (\text{n}^\circ \text{ de rainhas brancas}) - (\text{n}^\circ \text{ de rainhas pretas})$$

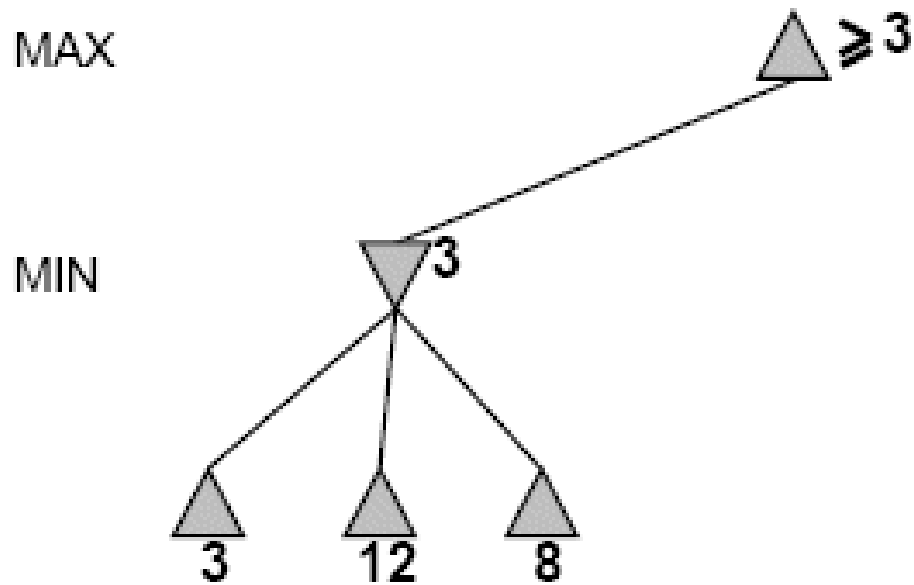
Função de avaliação (cont.)

- Os valores exatos não interessam
- O comportamento mantém-se para qualquer transformação monotónica da função de avaliação

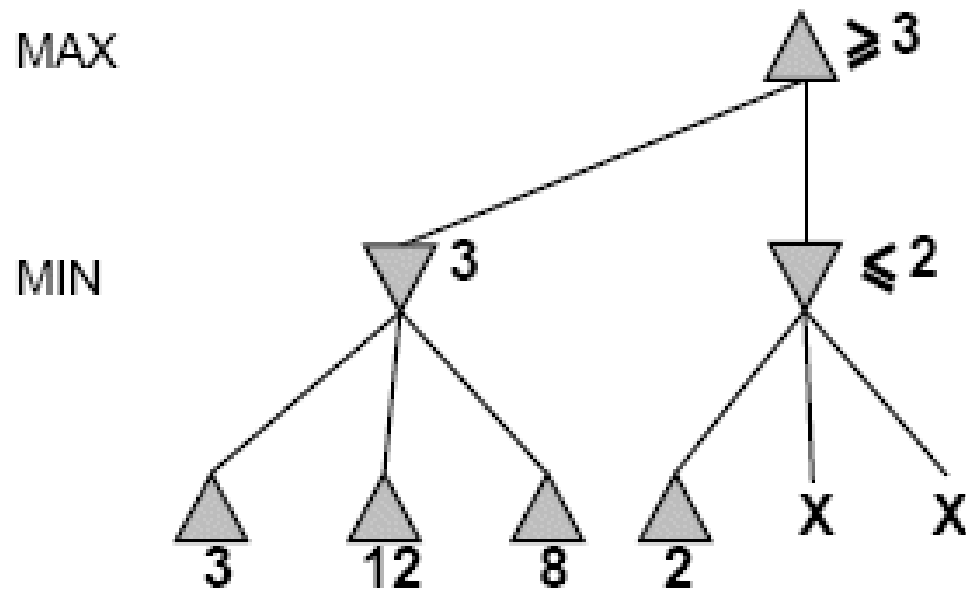


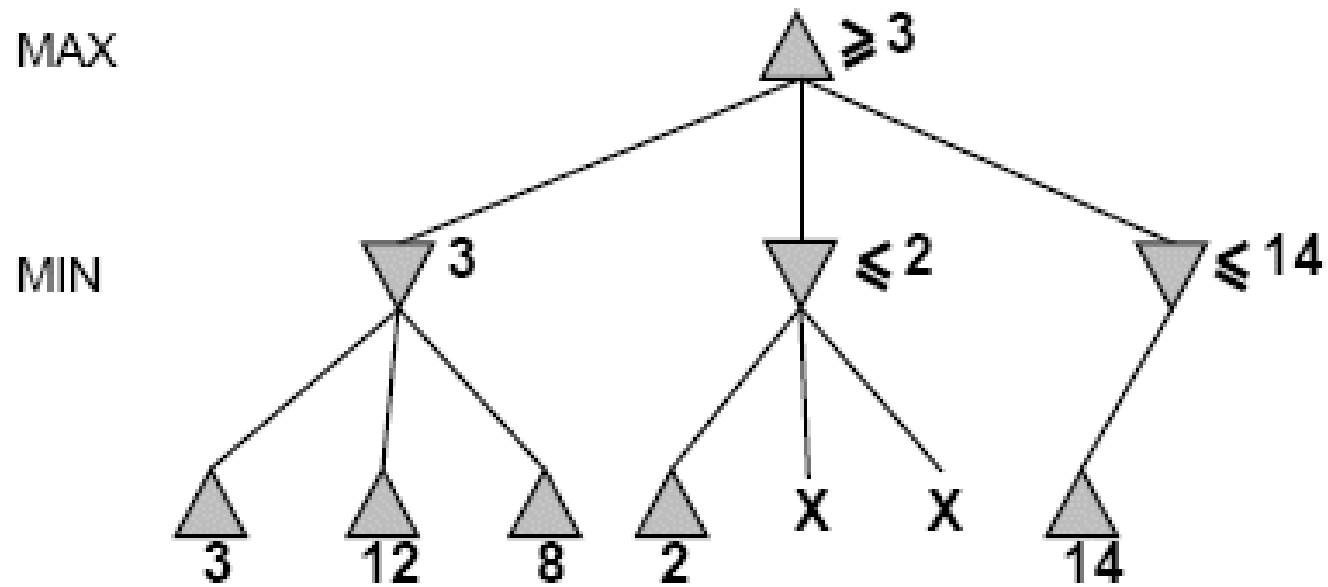
Corte da pesquisa (cutoff)

- MINIMAX-CUTOFF é idêntico a VALOR-MINIMAX exceto
 - TERMINAL? é substituído por CUTOFF?
 - UTILIDADE é substituído por AVALIAÇÃO
- Funciona na prática?
 - Exemplo do xadrez:
 $b^m = 10^6$, $b = 35 \implies m = 4$
 - um jogador que analisa até 4 jogadas futuras não tem hipóteses!
 - $m = 4$: humano principiante
 - $m = 8$: computador típico, humano com nível de mestre
 - $m = 12$: Deep blue, Kasparov



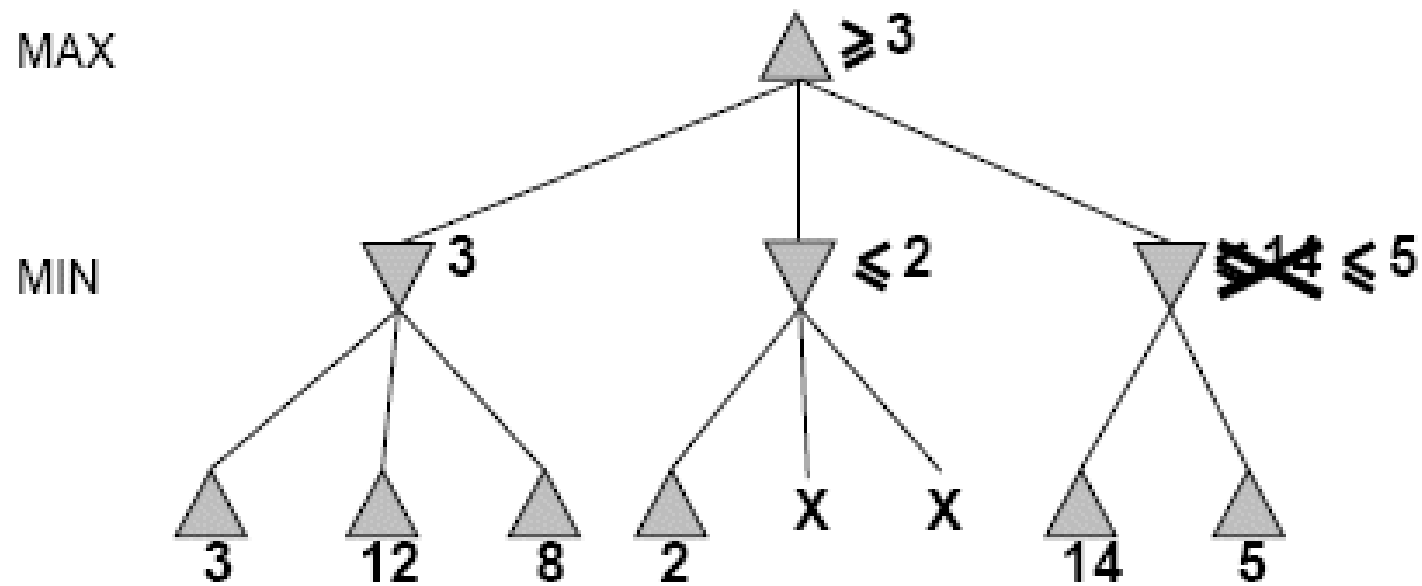
•



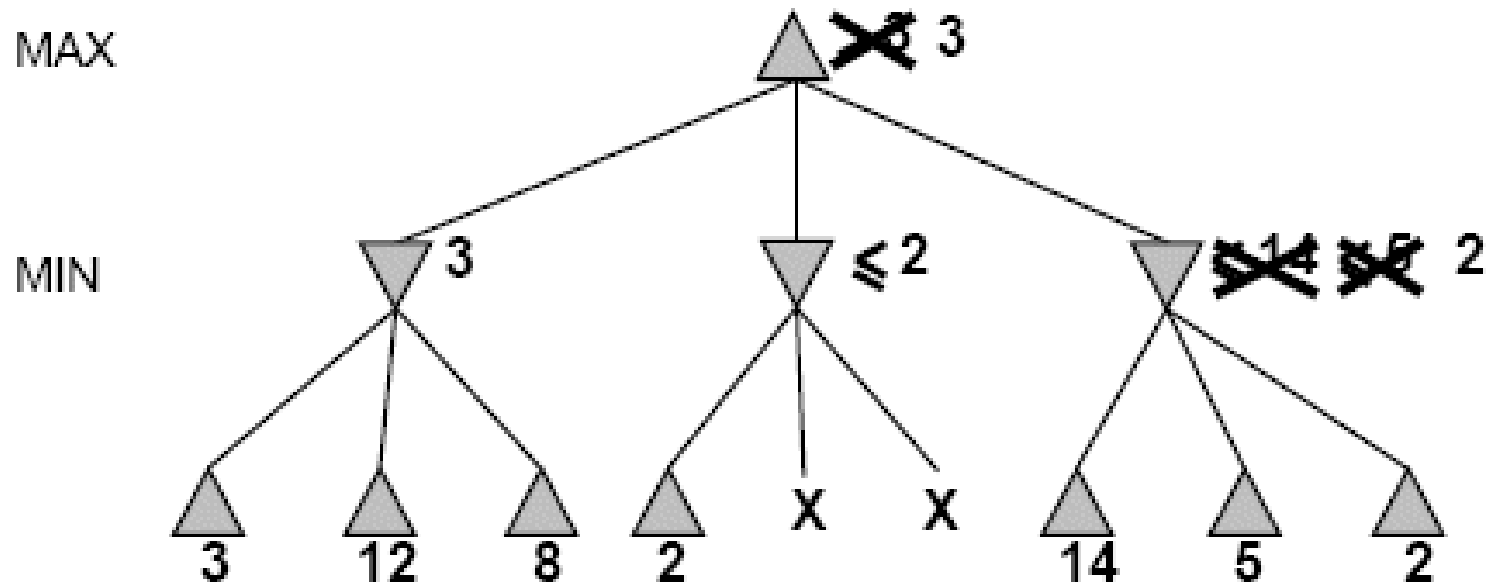


Corte $\alpha - \beta$ (cont.)

•



Corte $\alpha - \beta$ (cont.)



- O corte α - β **não afeta** o resultado final
- Uma boa ordenação dos nós aumenta a eficiência do corte α - β
- Com uma ordenação perfeita, complexidade temporal = $O(b^{m/2})$
 - duplica a profundidade da pesquisa
 - por exemplo, pode passar de 4 para 8 jogadas em xadrez, e passar do nível de principiante ao de mestre

Algoritmo do corte $\alpha - \beta$

função PESQUISA-ALFA-BETA(estado, jogo) retorna ação
 ação, estado \leftarrow a, s em SUCESSORES[jogo](estado)
 tal que VALOR-MIN(s, jogo, $-\infty$, $+\infty$) é máximo
 retorna ação

função VALOR-MAX(estado, jogo, α , β) retorna valor minimax de estado
 se TESTE-CORTE(estado) então retorna AVALIAÇÃO(estado)
 para cada s em SUCESSORES(estado)
 $\alpha \leftarrow \max(\alpha, \text{VALOR-MIN}(s, \text{jogo}, \alpha, \beta))$
 se $\alpha \geq \beta$ então retorna β
 retorna α

função VALOR-MIN(estado, jogo, α , β) retorna valor minimax de estado
 se TESTE-CORTE(estado) então retorna AVALIAÇÃO(estado)
 para cada s em SUCESSORES(estado)
 $\beta \leftarrow \min(\beta, \text{VALOR-MAX}(s, \text{jogo}, \alpha, \beta))$
 se $\beta \leq \alpha$ então retorna α
 retorna β

α : a melhor alternativa para MAX ao longo do caminho

β : a melhor alternativa para MIN ao longo do caminho

- Damas
 - Chinook venceu o campeão humano histórico Marion Tinsley em 1994; usou base de dados de finais (443 748 401 247 posições)
- Xadrez
 - Deep Blue derrotou Kasparov em 1997. Pesquisa 200 milhões de posições por segundo; algumas linhas de pesquisa têm profundidade 40
- Othello
 - Campeões humanos recusam-se a jogar contra computadores, que são demasiados bons
- Go
 - Campeões humanos recusam-se a jogar contra computadores, que são demasiado fracos. No jogo do Go, $b > 300$.