

# Advanced School in Artificial Intelligence

## Strategie di ricerca

Ricerca per giochi

Marco Alberti

[marco.alberti@unife.it](mailto:marco.alberti@unife.it)



*Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021*

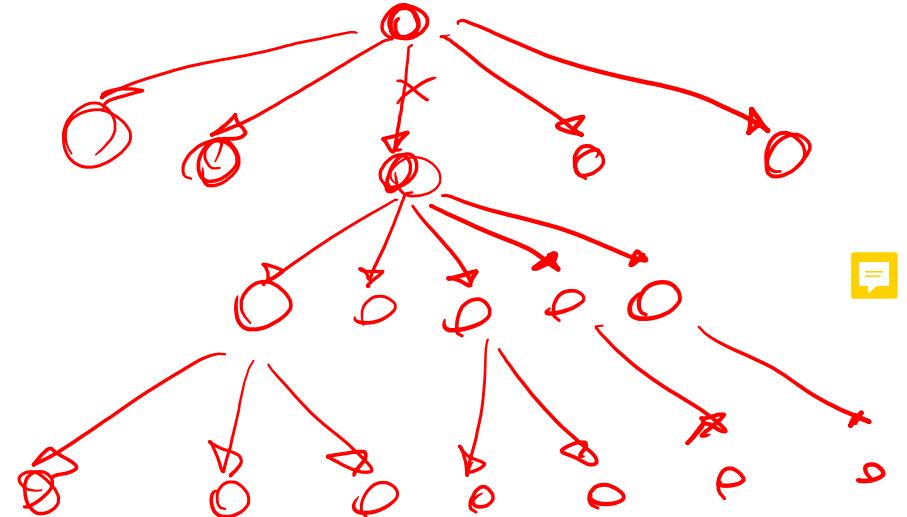


Università  
degli Studi  
di Ferrara

## Algoritmi di ricerca per giochi

Giochi come tris, dama, scacchi, go sono

- deterministici
- a due giocatori
- a turni
- a informazione perfetta
- a somma zero



## Caratterizzazione di giochi

game

- $S_0$ : lo **stato iniziale**, in cui si trova il gioco all'inizio
- To-MOVE( $s$ ): quale giocatore muove nello stato  $s$
- ACTIONS( $s$ ): l'insieme di mosse valide nello stato  $s$
- RESULT( $s,a$ ): il **modello di transizione**, quale stato risulta dalla mossa  $a$  nello stato  $s$
- Is-TERMINAL( $s$ ): **test di terminazione**, vero se e solo se in  $s$  il gioco è terminato
- UTILITY( $s,p$ ): **funzione di utilità** (o funzione obiettivo o payoff), il valore per il giocatore  $p$  dello stato terminale  $s$

STATO

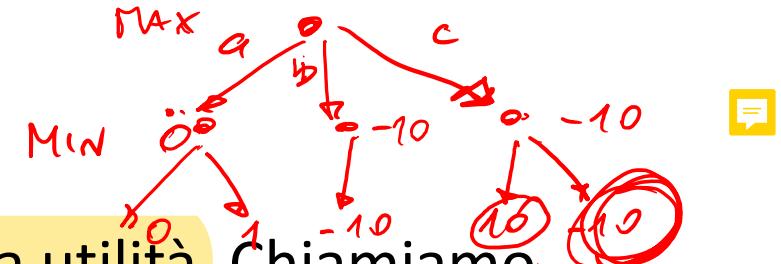
POSIZIONE

giocatore che muove



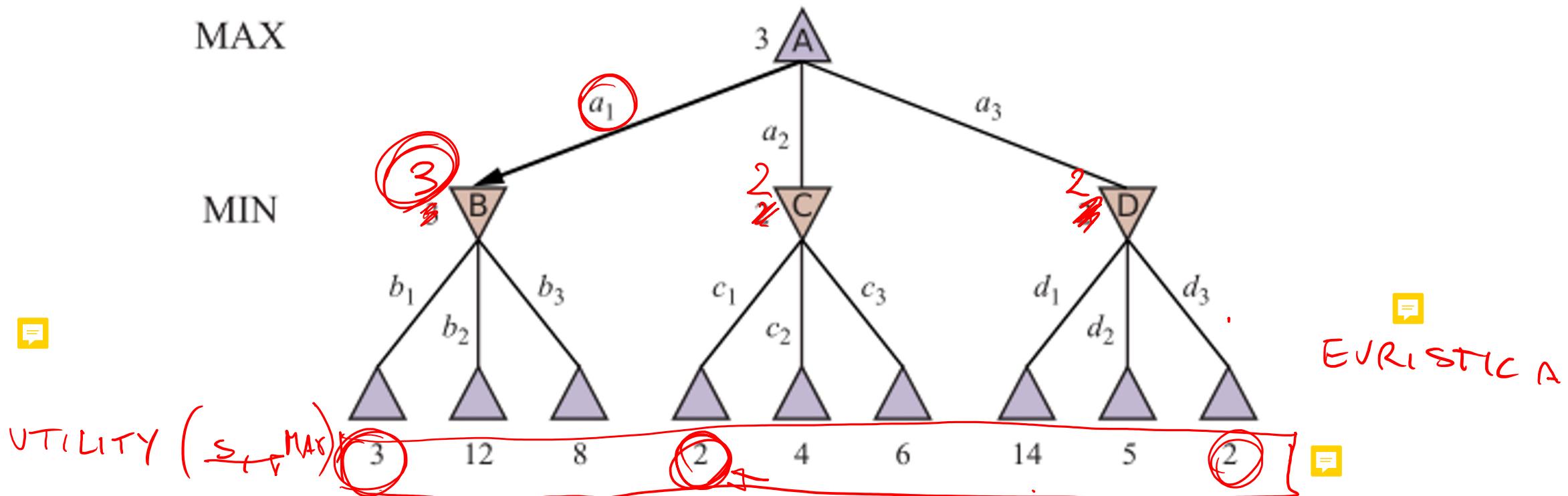
100      0      -100

## Come selezionare la mossa migliore?



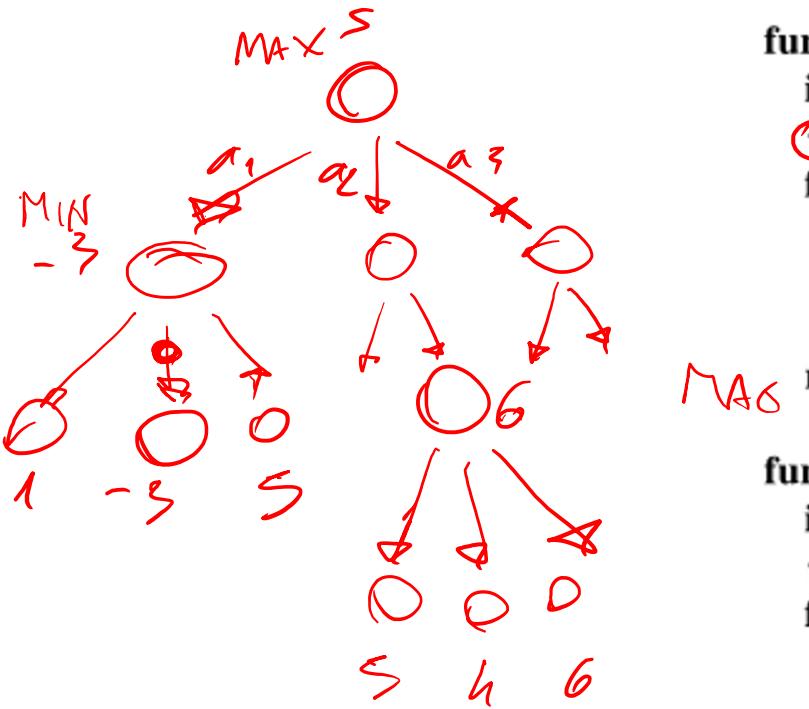
- Il giocatore muoverà cercando di massimizzare la sua utilità. Chiamiamo questo giocatore MAX
- L'avversario cercherà di minimizzare l'utilità di MAX; lo chiamiamo MIN
- A una mossa di MAX, MIN risponderà con la sua mossa che minimizza l'utilità di MAX
- Quindi MAX deve selezionare la mossa per cui è massimo il minimo delle utilità delle risposte di MIN

## Esempio



# Advanced School in Artificial Intelligence

## min-max



```
function MINIMAX-SEARCH(game, state) returns an action
    player  $\leftarrow$  game.TO-MOVE(state)
    value, move  $\leftarrow$  MAX-VALUE(game, state)
    return move
```



```
function MAX-VALUE(game, state) returns a (utility, move) pair
    if game.IS-TERMINAL(state) then return game.UTILITY(state, player), null
    v  $\leftarrow$   $-\infty$ 
    for each a in game.ACTIONS(state) do
        v2, a2  $\leftarrow$  MIN-VALUE(game, game.RESULT(state, a))
        if v2 > v then
            v, move  $\leftarrow$  v2, a
    return v, move
```

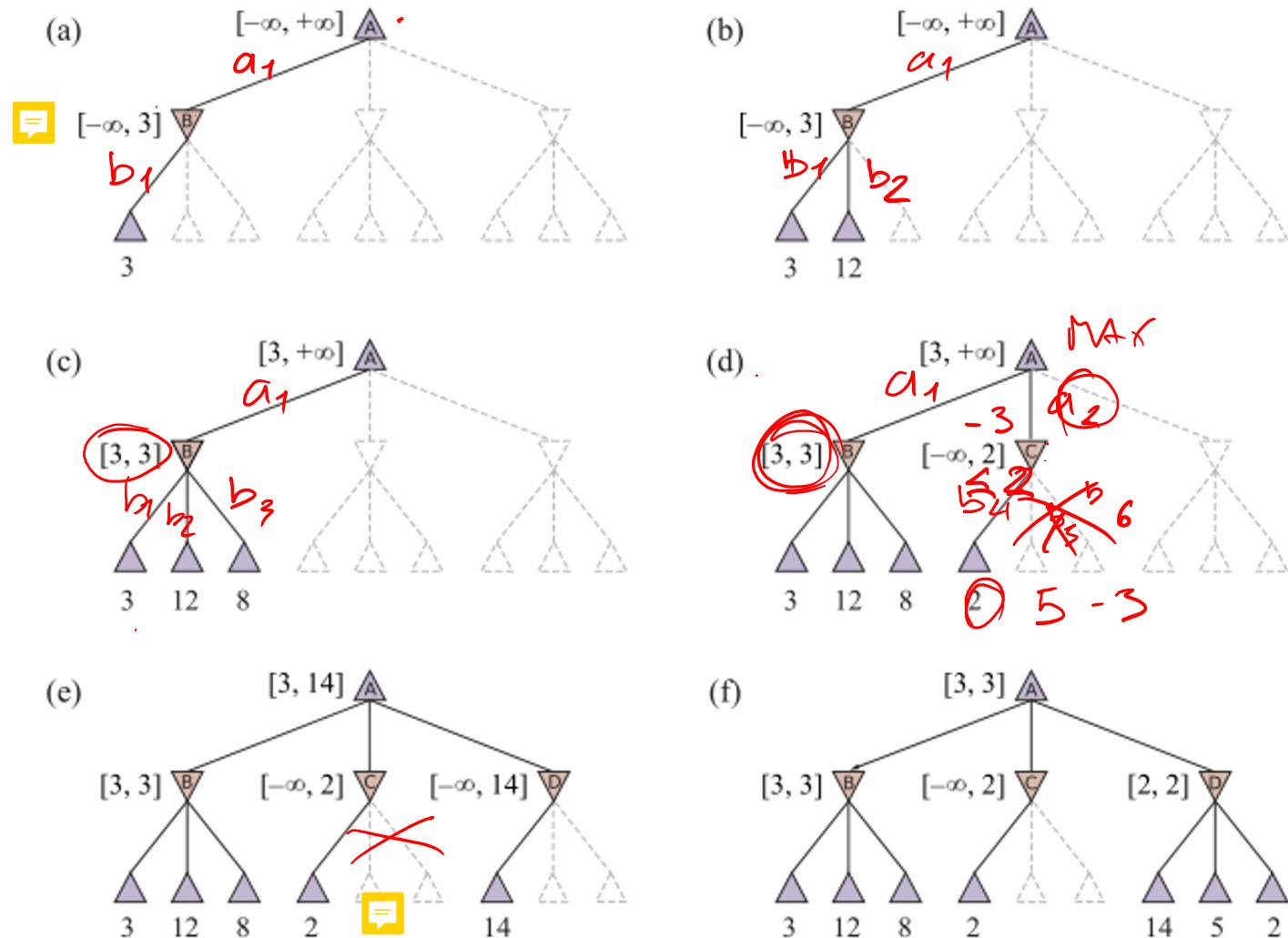


```
function MIN-VALUE(game, state) returns a (utility, move) pair
    if game.IS-TERMINAL(state) then return game.UTILITY(state, player), null
    v  $\leftarrow$   $+\infty$ 
    for each a in game.ACTIONS(state) do
        v2, a2  $\leftarrow$  MAX-VALUE(game, game.RESULT(state, a))
        if v2 < v then
            v, move  $\leftarrow$  v2, a
    return v, move
```



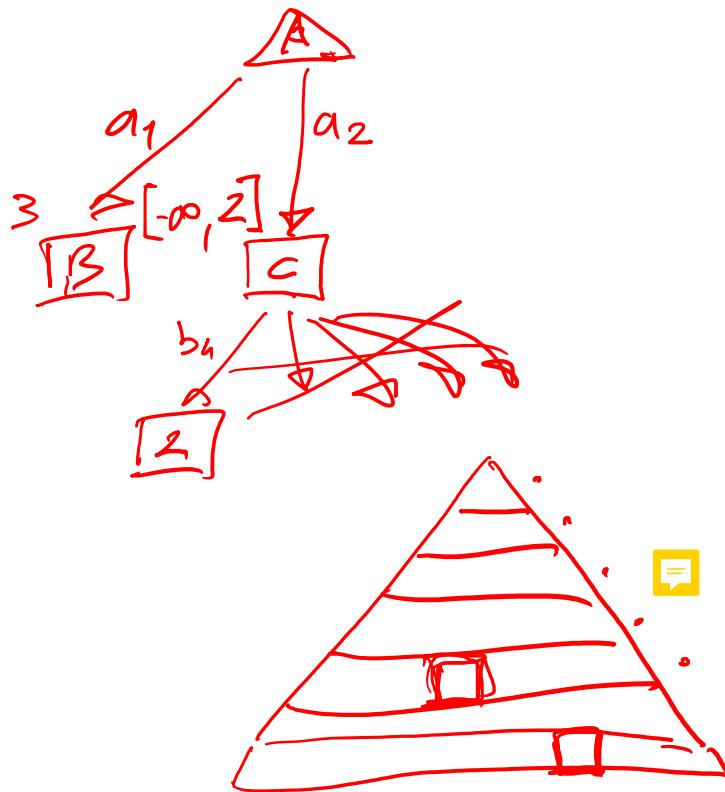
## Tagli alfa-beta

- In certi casi, si può dire con certezza che un nodo avrà un valore peggiore, per il giocatore che lo sta considerando, del migliore dei suoi fratelli
  - In questi casi si può smettere di espandere il nodo, o tagliare i rami rimanenti dell'albero di cui è radice



# Advanced School in Artificial Intelligence

## alfa-beta



```
function ALPHA-BETA-SEARCH(game, state) returns an action
    player ← game.TO-MOVE(state)
    value, move ← MAX-VALUE(game, state, -∞, +∞)
    return move
```

☞

```
function MAX-VALUE(game, state, α, β) returns a (utility, move) pair
    if game.IS-TERMINAL(state) then return game.UTILITY(state, player), null
    v ← -∞
    for each a in game.ACTIONS(state) do
        v₂, a₂ ← MIN-VALUE(game, game.RESULT(state, a), α, β)
        if v₂ > v then
            v, move ← v₂, a
            α ← MAX(α, v)
        if v ≥ β then return v, move
    return v, move
```

TAGLIO ALFA-BETA ☞

**C 3**

```
function MIN-VALUE(game, state, α, β) returns a (utility, move) pair
    if game.IS-TERMINAL(state) then return game.UTILITY(state, player), null
    v ← +∞
    for each a in game.ACTIONS(state) do
        v₂, a₂ ← MAX-VALUE(game, game.RESULT(state, a), α, β)
        if v₂ < v then
            v, move ← v₂, a
            β ← MIN(β, v)
        if v ≤ α then return v, move
    return v, move
```

