

Strategie di ricerca con avversario

Studieremo alcune strategie che tengono conto della presenza di altri agenti che possono manipolare l'ambiente in cui operiamo. Tali agenti sono supposti avere obiettivi che contrastano con quelli che vorremmo conseguire. Le loro azioni non saranno quindi cooperative ma competitive.

Ambienti competitivi

- Ambienti **multi-agente**
- Ogni agente è guidato da **obiettivi**
- Gli obiettivi di agenti diversi sono **conflittuali**, cioè il conseguimento degli obiettivi di un agente impedisce il conseguimento degli obiettivi degli altri agenti
- I problemi di ricerca con avversario sono anche detti **giochi**



I giochi nell'AI e in altre discipline

Marvin Minsky (1968):

*“i giochi non vengono scelti perché sono chiari e semplici,
ma perché ci danno la massima complessità con le minime
strutture iniziali”*

Pungolo Scientifico

- **Matematica**: teoria dei grafi e complessità
- **Computer Science**: AI, database, calcolo parallelo, etc.
- **Economia**: teoria dei giochi, eco. cognitiva/sperim.
- **Psicologia**: fiducia, rischio, etc..

Tipologie di giochi

- Condizioni di scelta
 - Ad informazione “perfetta” : gli stati del gioco sono totalmente espliciti per tutti gli agenti
 - Ad informazione “imperfetta” : gli stati del gioco sono solo solo parzialmente esplicitati
- Effetti della scelta
 - Deterministici: gli stati sono determinati unicamente dalle azioni degli agenti
 - Stocastici: gli stati sono determinati anche da fattori esterni (es: dadi)

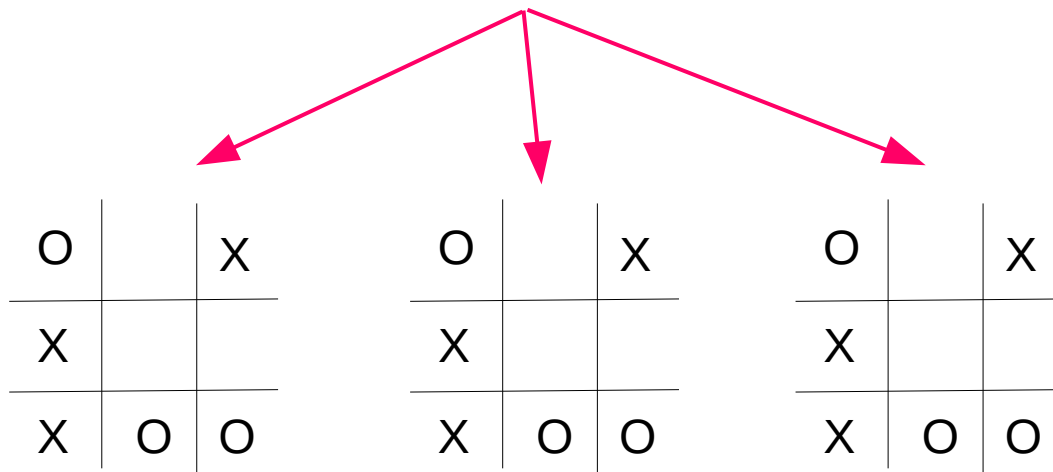
Esempi

	Informazione Perfetta	Informazione Imperfetta
Giochi deterministici	Scacchi, Go, Dama, Otello, Forza4, tris	MasterMind
Giochi stocastici	Backgammon, Monopoli	Scarabeo, Bridge, Poker... (giochi di carte) Risiko

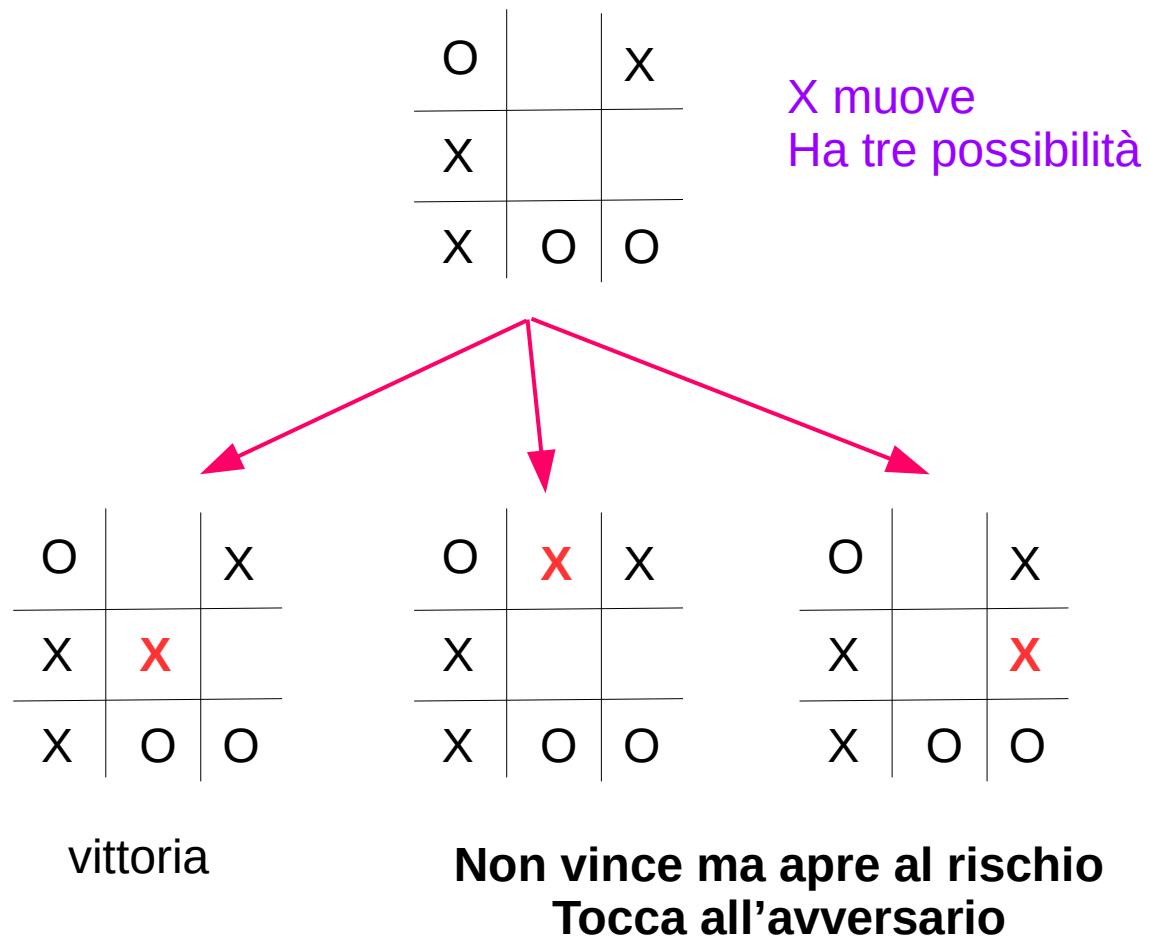
Esempio: gioco del tris

O		X
X		
X	O	O

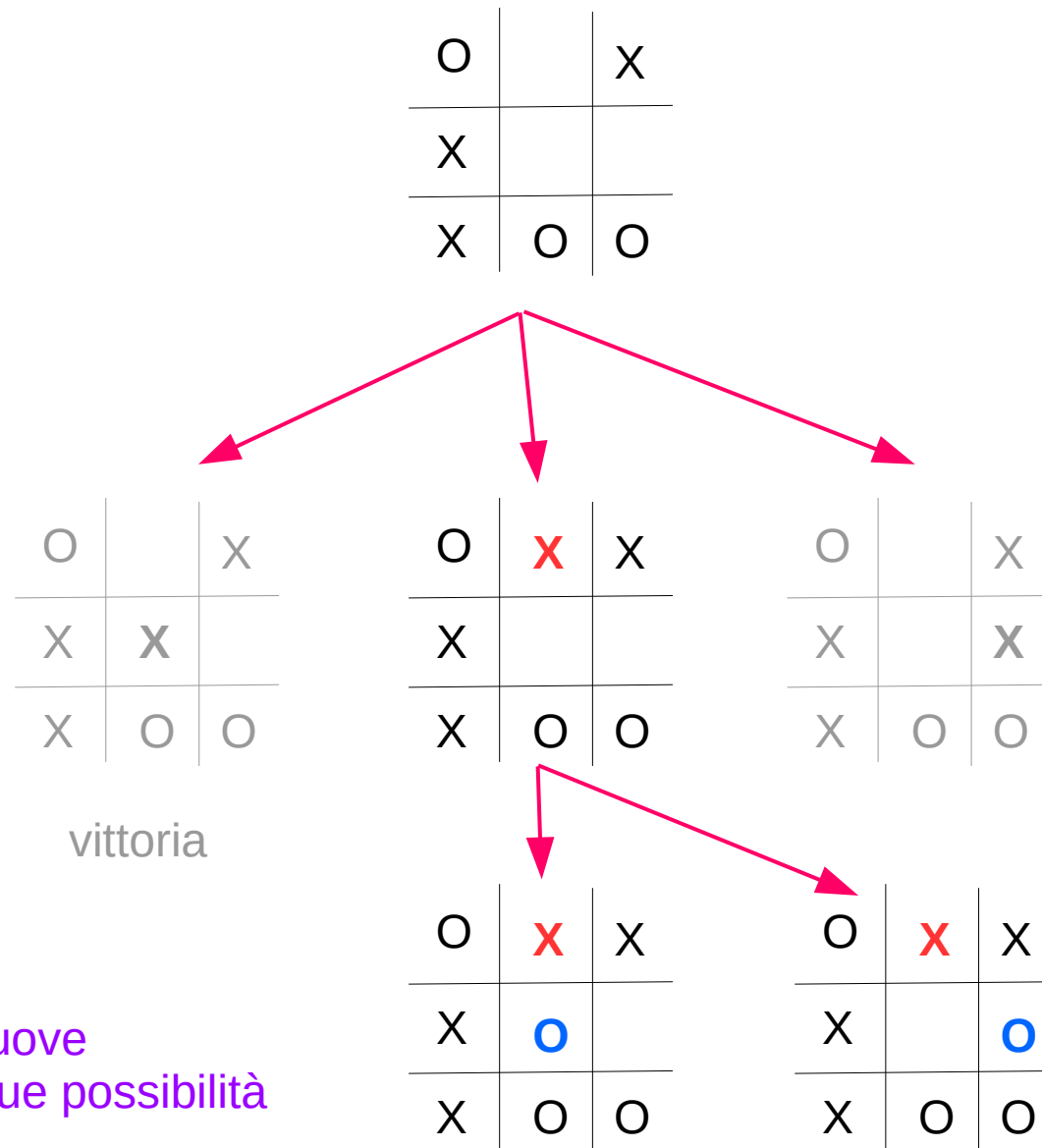
X muove
Ha tre possibilità



Esempio: gioco del tris



Esempio: gioco del tris

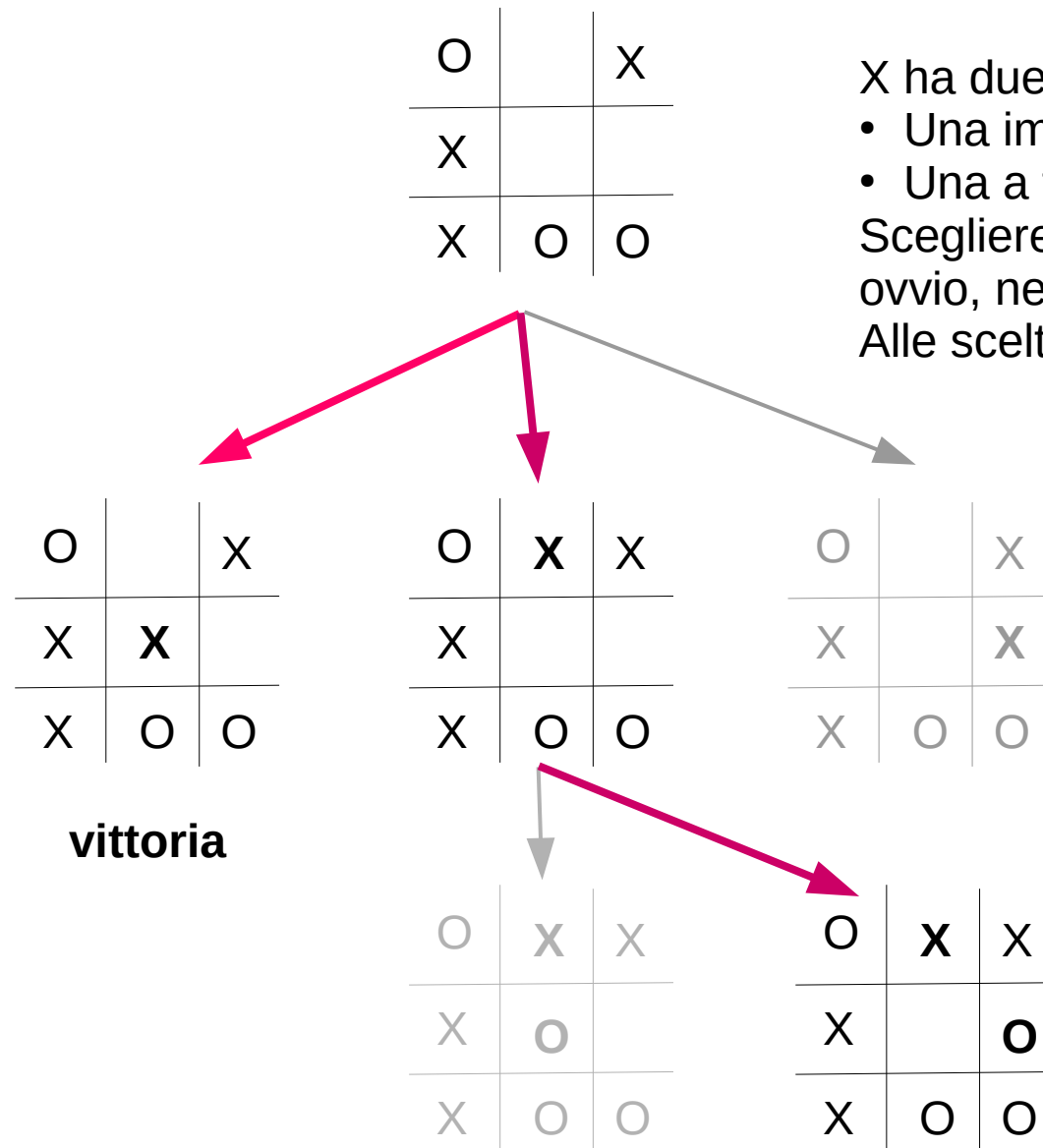


O muove
Ha due possibilità

Vittoria di O

**La vittoria sarà di X
al prossimo turno**

Esempio: due strategie per la vittoria



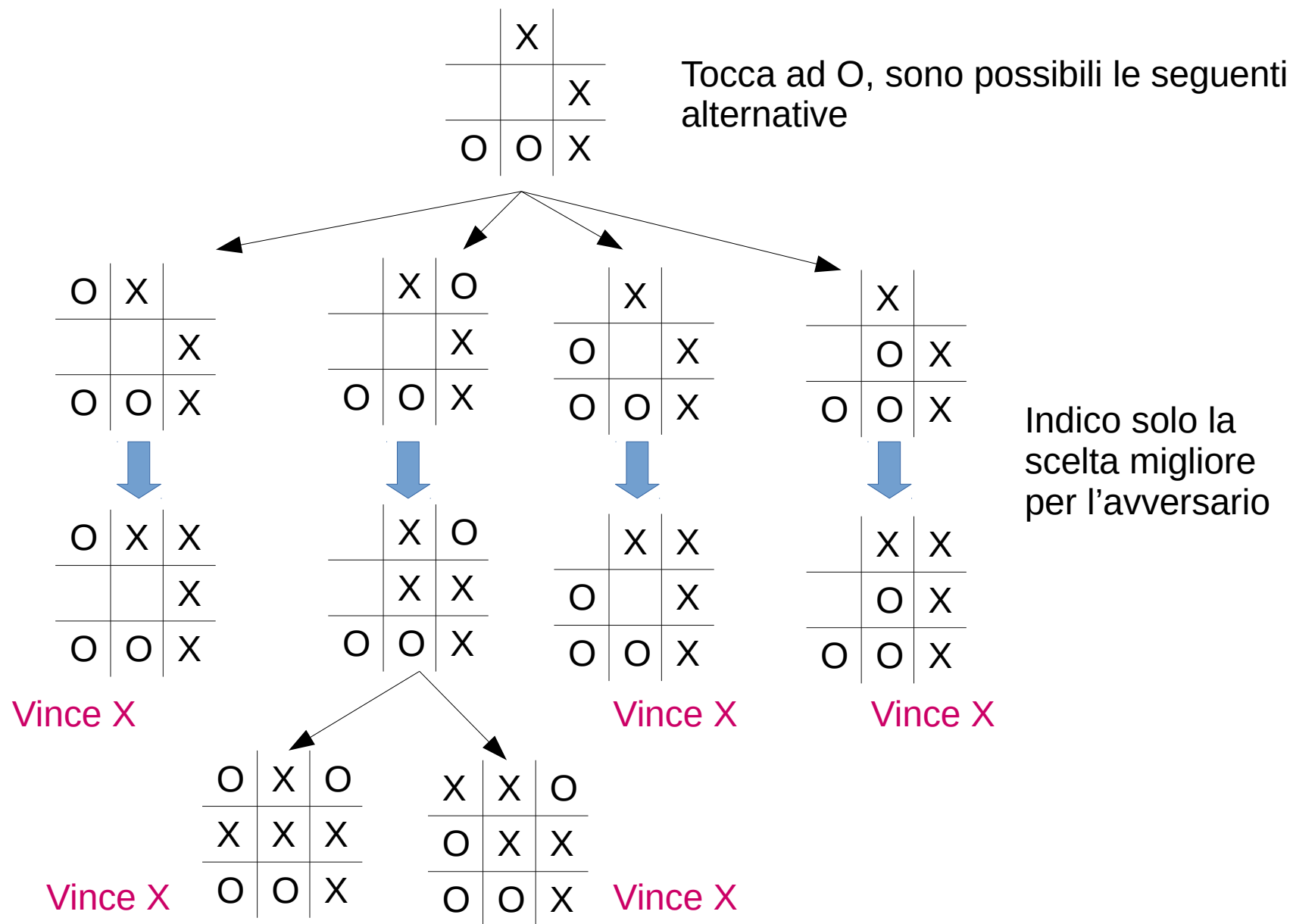
X ha due strategie per la vittoria:

- Una immediata
- Una a tre passi

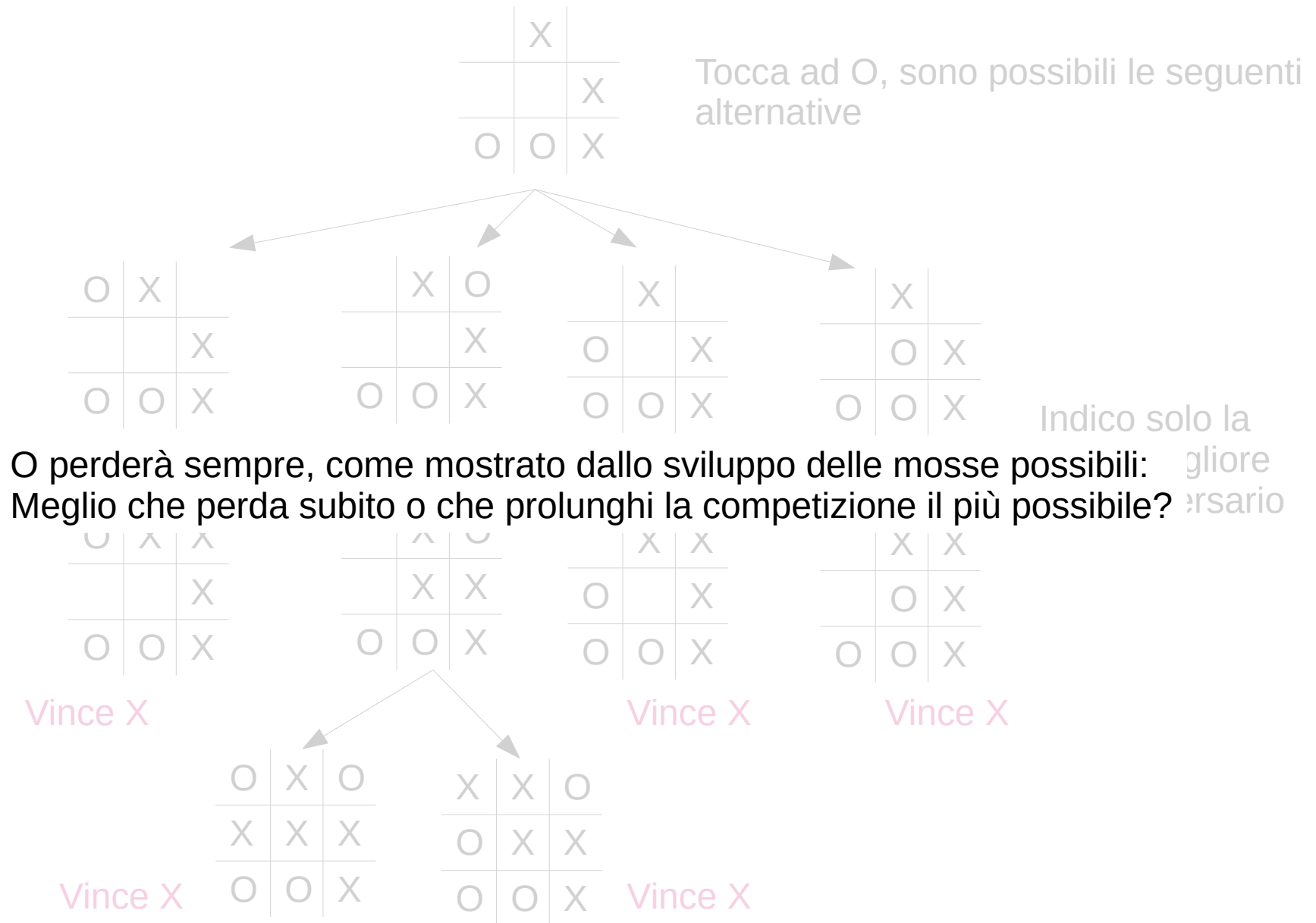
Scegliere quella immediata sembra ovvio, nell'altro caso ci esponiamo
Alle scelte del secondo giocatore

Vittoria al prossimo turno

Esempio: e in caso di perdita?



Esempio: e in caso di perdita?



Differenze con ricerca (non) informata

- L' obiettivo del giocatore è **determinare una strategia** (sequenza di mosse) che porti alla **vittoria**
- $g(x)$ non si usa
- I **nodi terminali** sono categorizzabili in:
vittoria, sconfitta, parità
- **Anche l' avversario muove!** Quindi la scelta del nodo **successore** non è sempre controllabile

Giochi a somma zero

- **Un gioco a somma zero** è un contesto di interazione multiagente in cui la perdita (o il guadagno) di un agente è compensata dal guadagno (perdita) degli altri
- **Esempio: le fette di torta**
quando le fette sono irregolari la parte in più di chi ha la fetta più grande è compensata dalla parte in meno che gli altri non ottengono



Approcci maximax, maximin, minimax regret (nella teoria delle decisioni)

- Approccio maximax (ottimistico)
- Approccio maximin (pessimistico o conservativo)
- Approccio minimax regret (“pentimento” minimax)
- Esempio: occorre scegliere quale investimento fare. È data una tabella dei payoff

alternative	sale	stabile	scende
fondo1	40	45	5
fondo2	70	30	-20
azioni	55	40	-10

Approcci maximax, maximin, minimax regret (teoria delle decisioni)

Andamenti: non sono controllabili,
dipendono da dinamiche esterne

alternative	sale	stabile	scende
fondo1	40	45	5
fondo2	70	30	-20
azioni	55	40	-10

Scelte, se ne può fare
una sola

Payoff: possono essere guadagno o
perdite ma anche riferirsi ad altre misure
(esempio: tempo, risorse). Sono specifici
del problema

Approccio maximax

Guarda i payoff più alti per ogni possibile scelta e fa la scelta che promette di più in assoluto (fondo2 dà 70): **il massimo dei massimi**. E' ottimistica perché non ha garanzie che le dinamiche esterne faranno salire il fondo scelto (nell'esempio).

alternative	sale	stabile	scende
fondo1	40	45	5
fondo2	70	30	-20
azioni	55	40	-10

Approccio maximin

Questo approccio guarda le perdite maggiori legate a ciascuna scelta e poi esegue l'azione che minimizza le perdite (**il massimo dei minimi**). Nell'esempio sceglierà fondo 1 che è l'unica opzione a non andare in perdita.

alternative	sale	stabile	scende
fondo1	40	45	5
fondo2	70	30	-20
azioni	55	40	-10

Approccio minimax regret: regret

Best regret = Best payoff – Real payoff

Se il mercato sale il best payoff è 70. Se abbiamo investito in fondo1 il real regret è 70-40 Cioè 30. Se abbiamo investito in azioni è 15 e se abbiamo investito in fondo2 è 0.



alternative	sale	stabile	scende
fondo1	40	45	5
fondo2	70	30	-20
azioni	55	40	-10

Approccio minimax regret: regret

Best regret = Best payoff – Real payoff

Stesso genere di conti viene fatto per gli altri casi: mercato stabile o in discesa.

alternative	sale	stabile	scende
fondo1	40	45	5
fondo2	70	30	-20
azioni	55	40	-10

Si calcola una **regret table** ...

Approccio minimax regret: regret table

Si calcolano i massimi regret per ogni scelta alternativa e poi si seleziona l'alternativa che porta al regret minimo, nell'esempio acquistare azioni che dà un pentimento di 15.

alternative	sale	stabile	scende
fondo1	30	0	0
fondo2	0	15	25
azioni	15	5	15

Leggendo le righe: fondo1 dà regret massimo pari a 30, fondo2 dà 25, azioni dà 15.

Guadagno e giochi con avversario

- Nel nostro caso le dinamiche dell' ambiente sono costituite dall' avversario:
 - Come non sappiamo come evolverà l' ambiente, non sappiamo quale scelta farà l' avversario
 - Dobbiamo far bastare la *conoscenza dello stato corrente* e delle *mosse a disposizione*

Caratteristiche del gioco

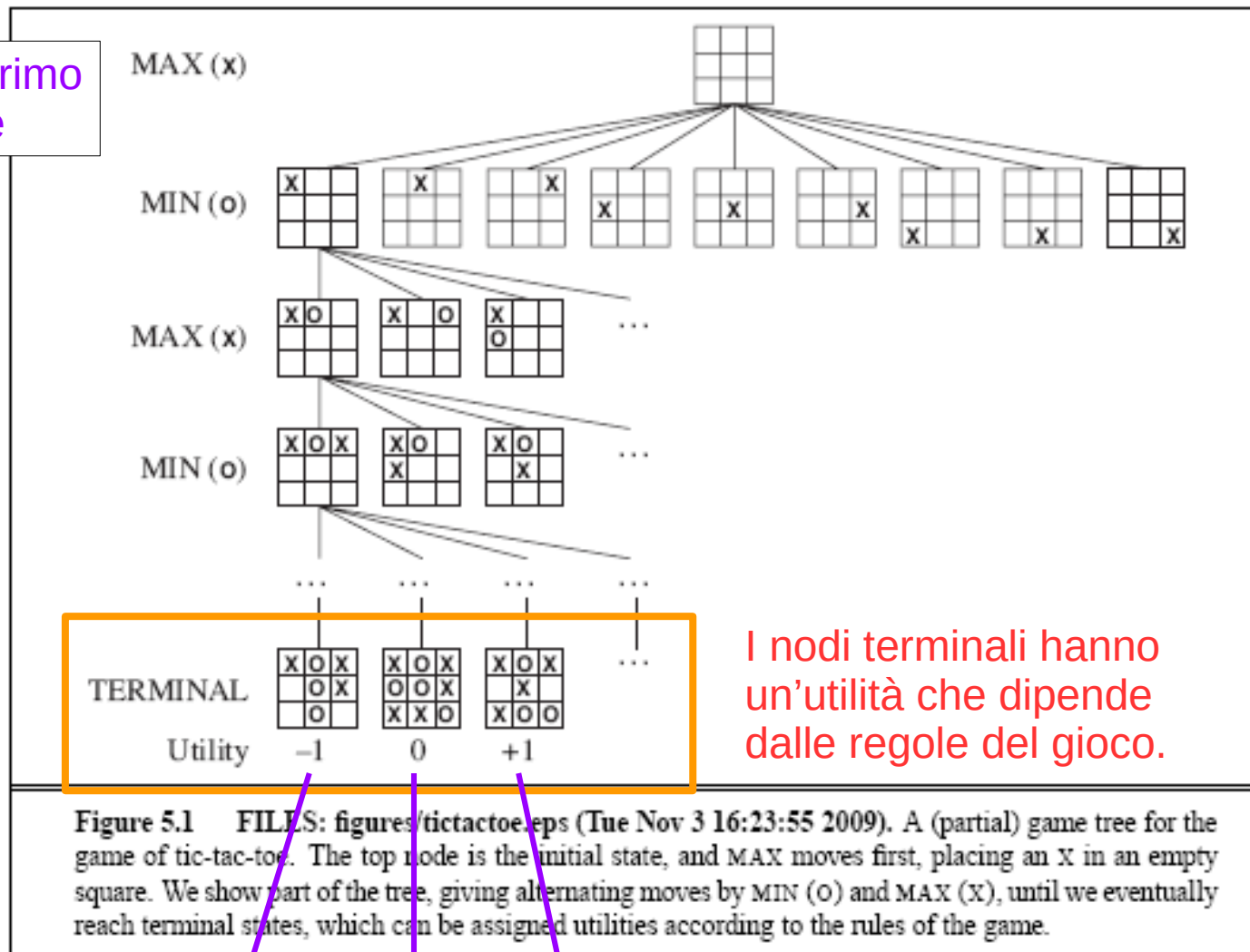
- **Due giocatori**, può essere comodo chiamarli MAX e MIN con **MAX che muove per primo**
- Ciascun giocatore non sa quali mosse farà l'altro ma le mosse possibili sono note e sono calcolabili i **successori** che produrranno una volta applicate a qualche stato
- **Osservabilità:**
 - È **totale** nel caso di giochi con turno: i giocatori conoscono i risultati delle mosse precedenti
 - È **parziale** nel caso di giochi ad azione simultanea: i giocatori non conoscono le mosse che i giocatori eseguono simultaneamente alla loro
- Partendo da uno **stato iniziale** è possibile sviluppare un **albero di possibili evoluzioni (albero di gioco)**, applicando le azioni eseguibili e calcolando così gli stati successivi
- Alcuni stati sono **terminali**, quando uno di essi è raggiunto la partita termina

Caratteristiche del gioco

- I giocatori si avvalgono del calcolo dell' **utilità degli stati**
- I giocatori devono **tener conto dell' avversario** quindi il calcolo dell' utilità comprende una valutazione del punto di vista dell' avversario
- **Giocatori pessimisti**: suppongono che l' avversario faccia sempre la mossa che gli porta il guadagno maggiore
- **Giochi con turno (2-ply)**

Utilità soggettiva nel caso del tris

MAX è il primo a muovere

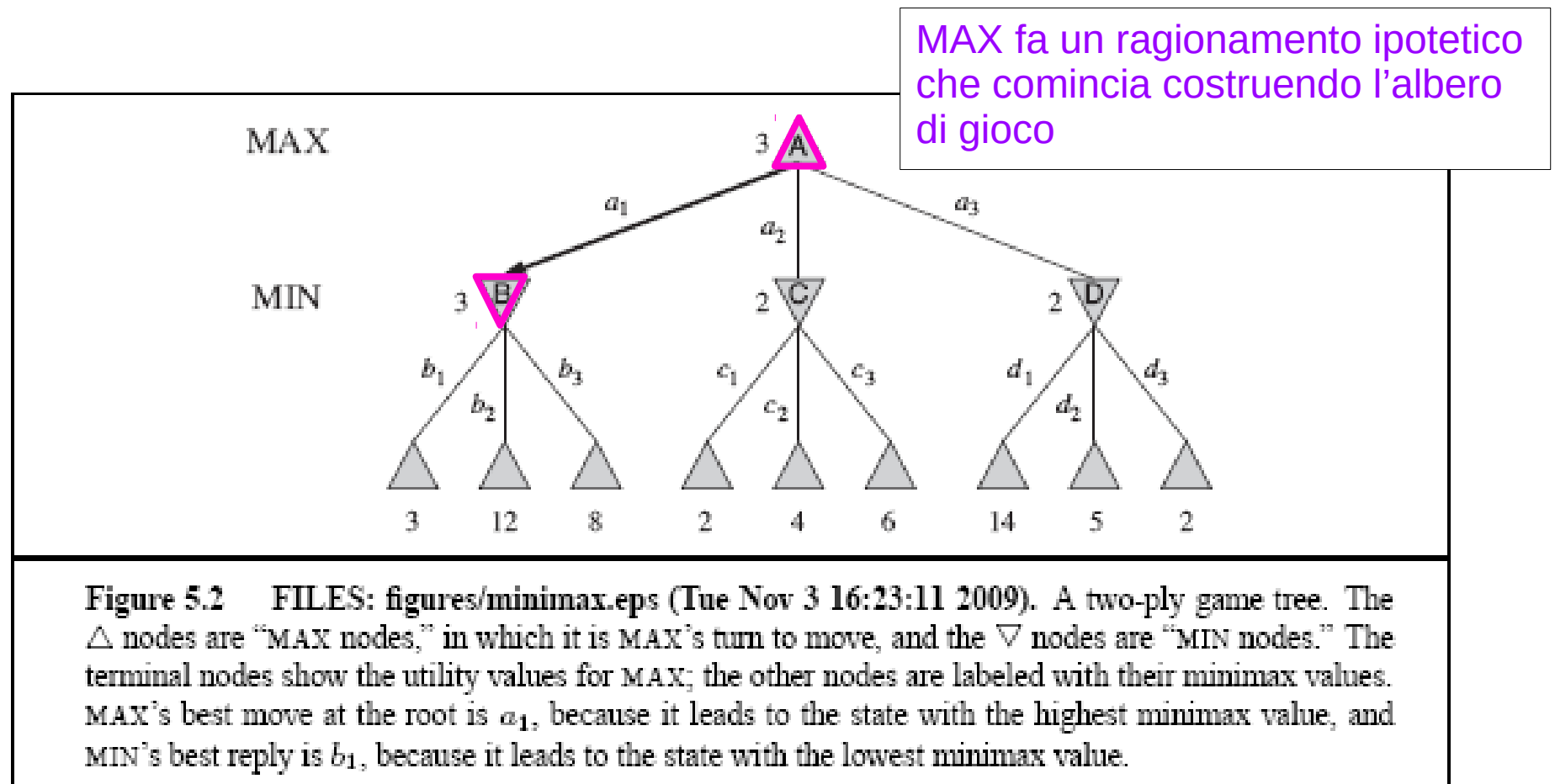


Cattivo, neutro, buono per MAX

Strategia ottima per un agente

- Sequenza di mosse che porta a uno stato terminale corrispondente alla vittoria dell' agente
- L' agente non sa come muoverà l' altro, può solo “immedesimarsi”

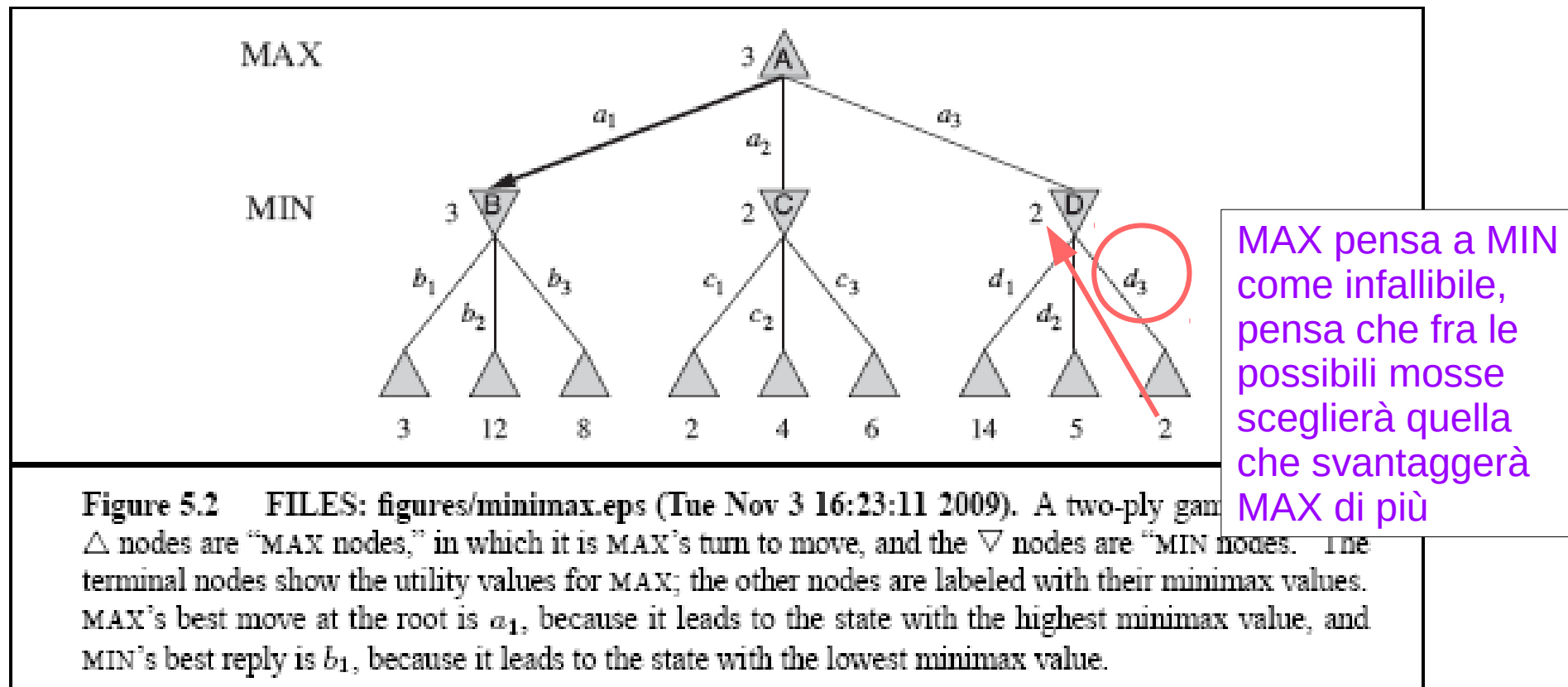
Ricerca strategia ottima, esempio



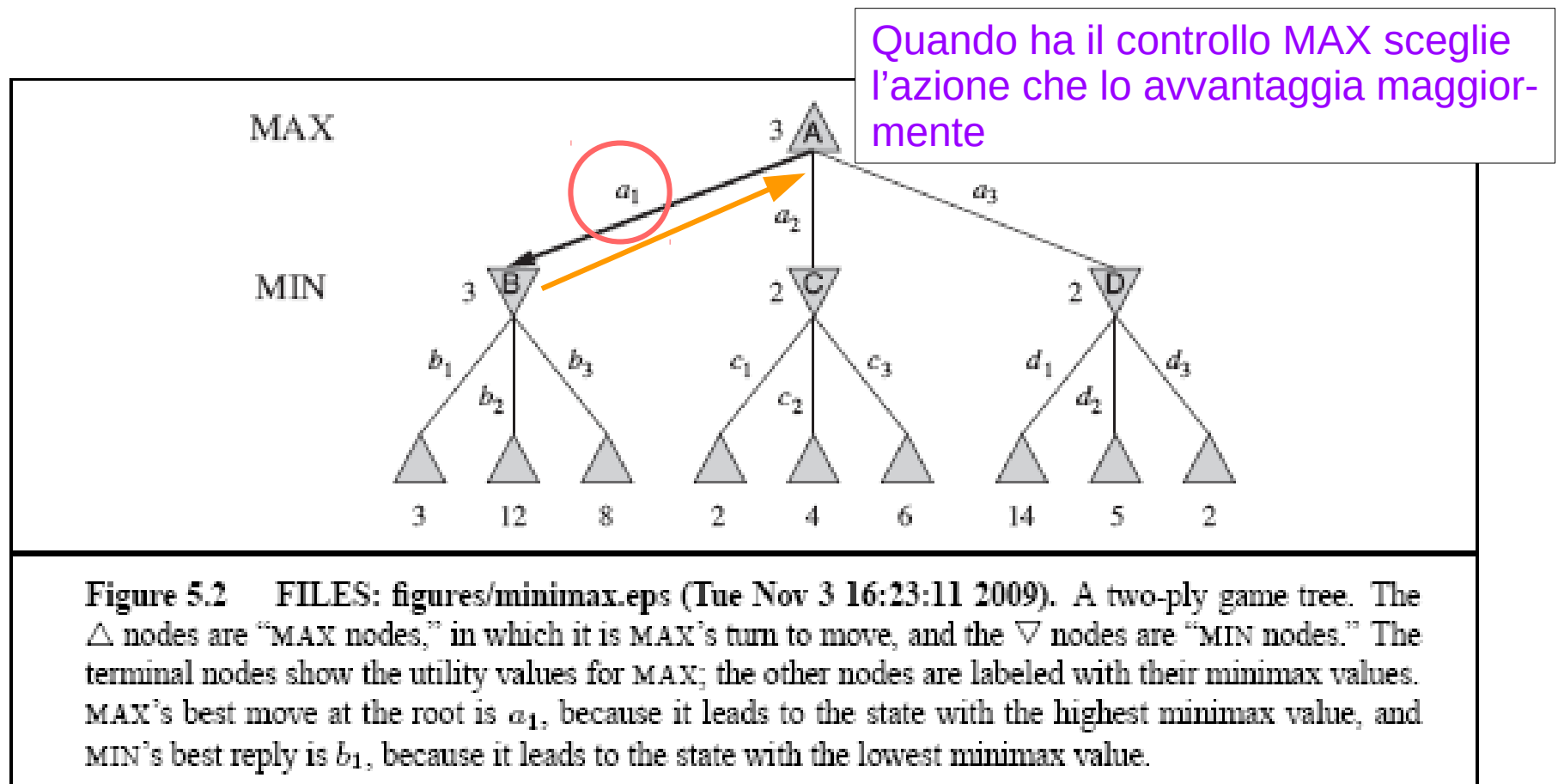
\triangle Nodi MAX

∇ Nodi MIN

Ricerca strategia ottima, esempio



Ricerca strategia ottima, esempio



Valore-minimax(n)

$$\text{Valore-minimax}(n) = \begin{cases} \text{utilità}(n) & \text{se } n \text{ terminale} \\ \max_{s \in \text{succ}(n)} \text{valore-minimax}(s) & \text{se } n \text{ è un nodo MAX} \\ \min_{s \in \text{succ}(n)} \text{valore-minimax}(s) & \text{se } n \text{ è un nodo MIN} \end{cases}$$

È un valore calcolato per ogni nodo. Permette all'agente la scelta della mossa da eseguire. I nodi MAX e MIN rappresentano due tipi di scelte: MAX è una **scelta interna** dell'agente mentre MIN è una **scelta esterna**, cioè tale che l'agente non ha controllo su di essa

Il nome minimax deriva dal fatto che ogni giocatore, minimizzando il guadagno massimo dell'altro minimizza la propria perdita

Nei **giochi a due** l'utilità per un giocatore è uguale a meno l'utilità per l'altro giocatore. Basta ricordare uno di questi valori per avere automaticamente anche l'altro

Algoritmo minimax ricorsivo

function MINIMAX-DECISION(*state*) *returns an action*

inputs: *state*, current state in game

$v \leftarrow \text{MAX-VALUE}(\text{state})$

return the *action* in **SUCCESSORS**(*state*) with value *v*

function MAX-VALUE(*state*) *returns a utility value*

if **TERMINAL-TEST**(*state*) **then return** **UTILITY**(*state*)

$v \leftarrow -\infty$

for *a, s* **in** **SUCCESSORS**(*state*) **do**

$v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s))$

return *v*

function MIN-VALUE(*state*) *returns a utility value*

if **TERMINAL-TEST**(*state*) **then return** **UTILITY**(*state*)

$v \leftarrow \infty$

for *a, s* **in** **SUCCESSORS**(*state*) **do**

$v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s))$

return *v*

Visita completa in profondità dell'albero di gioco

Complessità temporale: **$O(b^m)$**
con *b* = branching factor ed
m = profondità massima

Complessità spaziale: **$O(bm)$**
se i successori sono generati
contemporaneamente

Figure 6.3 An algorithm for calculating minimax decisions. It returns the action corresponding to the best possible move, that is, the move that leads to the outcome with the best utility, under the assumption that the opponent plays to minimize utility. The functions **MAX-VALUE** and **MIN-VALUE** go through the whole game tree, all the way to the leaves, to determine the backed-up value of a state.