

# Alberi di decisione

Alberto Borghese

Università degli Studi di Milano  
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)  
Dipartimento di Scienze dell'Informazione  
[borghese@dsi.unimi.it](mailto:borghese@dsi.unimi.it)



## Sommario



**Alberi di decisione.**

Alberi e teoria dei giochi



## Why agents are important?



**Agente** (software): essere software che svolge servizi per conto di un altro programma, solitamente in modo automatico ed invisibile. Tali software vengono anche detti agenti intelligenti

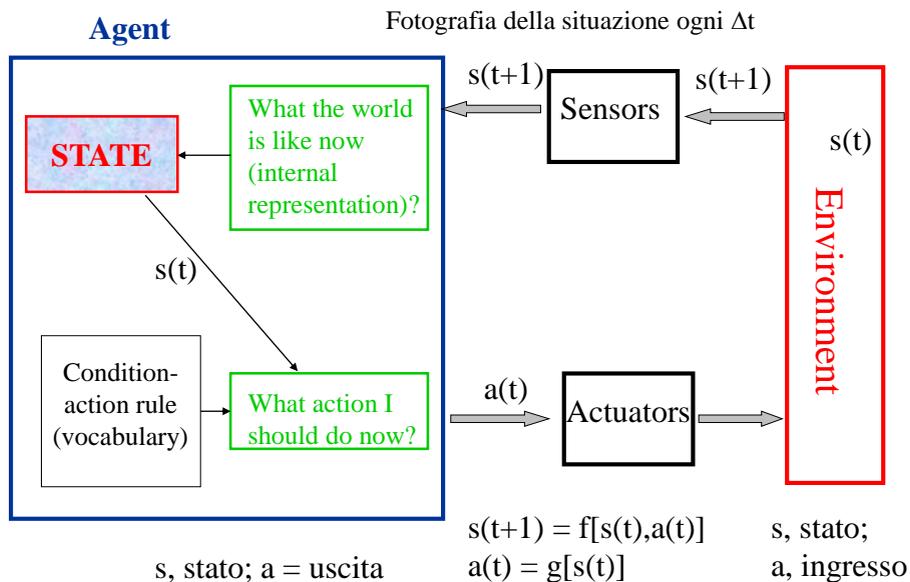
“They are seen as a natural metaphor for conceptualising and building a wide range of complex computer systems (the world contains many passive objects, but it also contains very many *active* components as well);

They cut across a wide range of different technology and application areas, including telecoms, human-computer interfaces, distributed systems, WEB and so on;

They are seen as a natural development in the search for ever-more powerful abstractions with which to build computer systems.“



## Schematic diagram of an agent





## How agents solve a problem



Formulate a problem. Through analysis. State, action, identification.

Solve the problem (by searching).

Implement the solution (execute).

Evaluate the implemented solution.

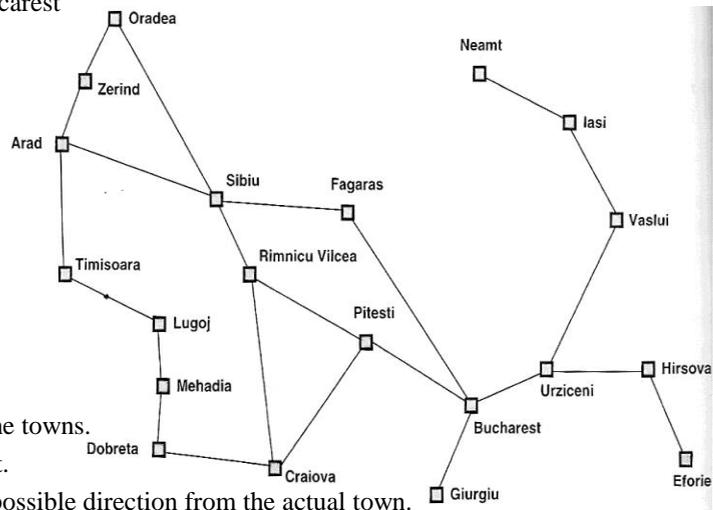
- ◆ Success or fail? Adequate or not adequate?
  - ◆ How much adequate? How to measure the success or failure of the performance?
  - ◆ Optimization of the performance to create better agents.
- 
- Solve a problem = achieve a given goal (= reach a final state)
  - An agent can examine different sequences of actions (deterministic or stochastic response by the environment) and search the best sequence.



## Example: route finding - formalization



- From Arad to Bucarest



State: ensemble of the towns.

Goal: reach Bucarest.

Action: choice of a possible direction from the actual town.

Cost: length of the path.

Initial state: Arad



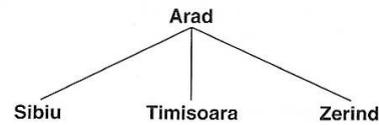
## Example: route finding - Costruzione dell'albero di ricerca



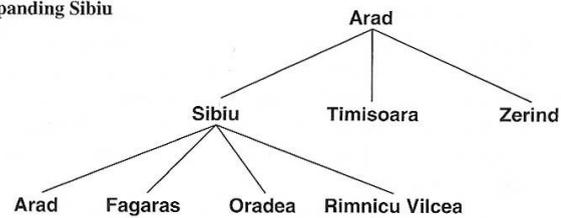
(a) The initial state

Arad

(b) After expanding Arad



(c) After expanding Sibiu



Which action is best for reaching the goal, in each state?



## Tecniche di search



- Valutazione in termini di:
  - ◆ - Completezza: la ricerca trova una soluzione, se esiste?
  - ◆ - Tempo: quanto richiede il calcolo di una soluzione?
  - ◆ - Complessità: quanta memoria viene richiesta per la ricerca?
  - ◆ - Ottimalità: la ricerca trova la soluzione ottima quando esistono più soluzioni possibili?

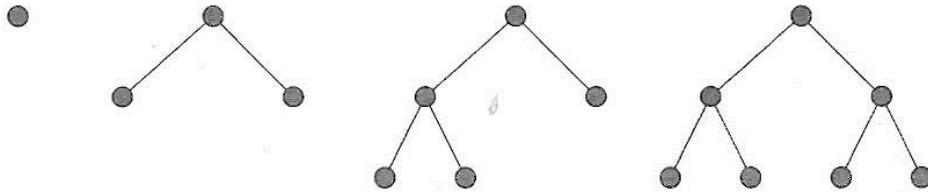
Tecniche prive di informazioni a-priori o con informazioni a-priori.



# Breadth-first search (larghezza prima)



b = branching factor



Ricerca completa.  
Ricerca ottima..

$N = 1 + b^1 + b^2 + b^3 + \dots + b^{d-1} = b^d - 1$   
Crescita non polinomiale in tempo e spazio



## Alcuni risultati

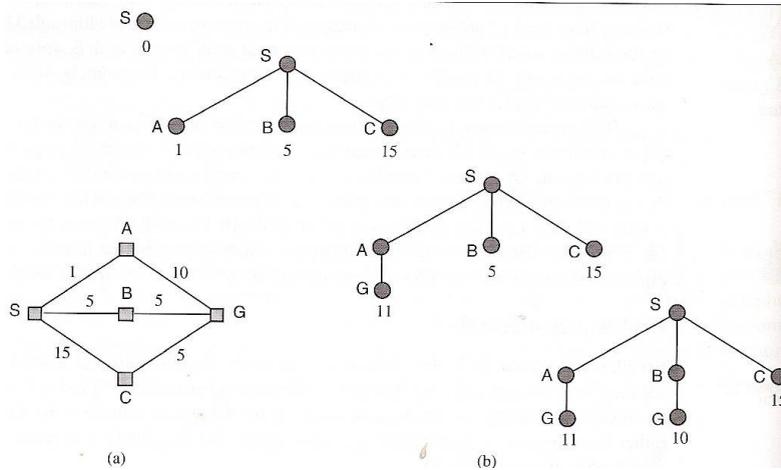


Profondità	Nodi	Tempo	Memoria
0	1	1 ms	100 byte
2	111	0.1 s	11 Kbyte
4	11,111	11 s	1 Mbyte
6	$10^6$	18 min	111 Mbyte
8	$10^8$	31 h	11 Gbyte
10	$10^{10}$	128 gg	1 Tbyte
12	$10^{12}$	35 anni	111 Tbyte
14	$10^{14}$	3500 anni	11,111 Tbyte

Memory requirements and computational time, make optimal search unfeasible as can be expected by a NP problem.



# Uniform cost search



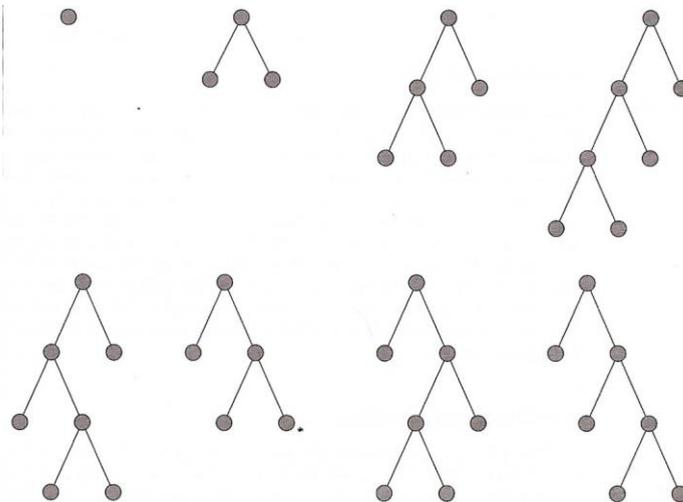
Ricerca completa.  
Ricerca ottima..

$$g(\text{SUCCESSOR}(n)) \geq g(n).$$



# Depth-first search

Come diminuire  
l'occupazione di  
memoria?



La memoria richiesta è pari a:  $M = b \cdot d$  (breadth-first:  $M = b^d$ )  
 $d = 12, b = 10$ ; 12 Kbyte versus 111 Tbyte!!



## Depth-limited search

Problemi della Depth-first search:

- Problemi per alberi profondi.
- Se prendo il ramo sbagliato, devo continuare ad esplorarlo, anche se, dopo pochi livelli mi potrei accorgere che è l'albero sbagliato.

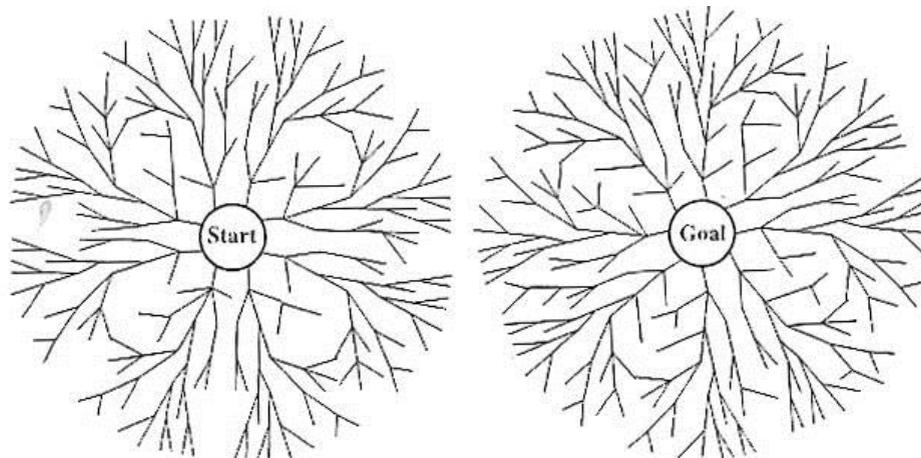
Come evitare ciò?

Imporre una limitazione alla profondità dell'albero visitato. Questo si può fare in modo informato quando si hanno informazioni a-priori sui dati.

Questo limite richiede informazioni e non è detto che sia un buon limite.



## Bidirectional search



$O(2b^{d/2})$  invece di  $O(b^d)$

Esempio:  $b=10$ ;  $d=6$ ; 2,222 nodi versus 1,111,111 of breath-first search



## Problems with bidirectional policies



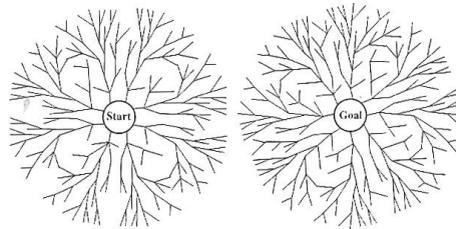
How do we get backwards from the goal (e.g. agent exploring the world)?

Come si inverte la funzione stato-prossimo?

Cosa succede se ci sono diversi goal state?

Match tra nodi dei due sotto-alberi.

Tipo di ricerca nei due sottoalberi.



A.A. 2017-2018

15/30

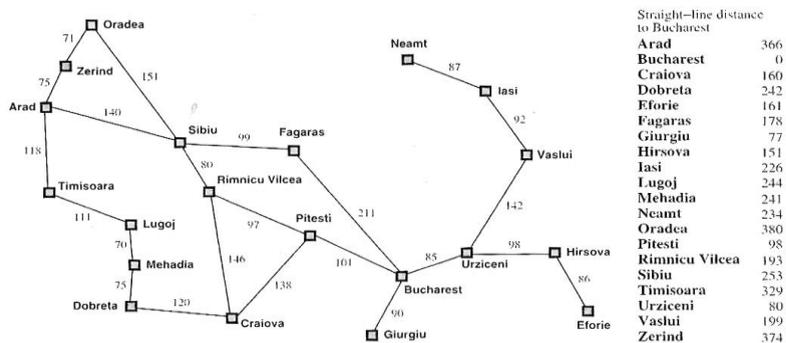
<http://borghese.di.unimi.it/>



## Informed search



- Best-first-search (seemingly-best first-search).
- Evaluation of the node not in terms of the past (uniform cost) but in global terms.
- Minimize the **cost-to-go**. Estimate of the value of each state. For instance, this can be the linear distance to the goal.



Greed (golosità) è considerato uno dei peccati capitali..

“Good is enemy of optimal”

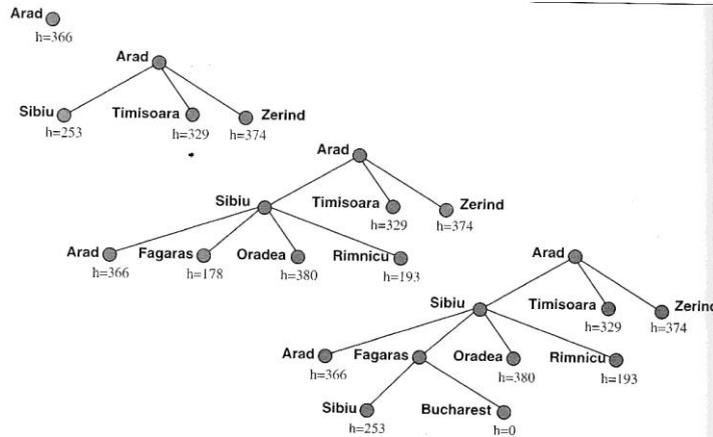
<http://borghese.di.unimi.it/>



# A\* search



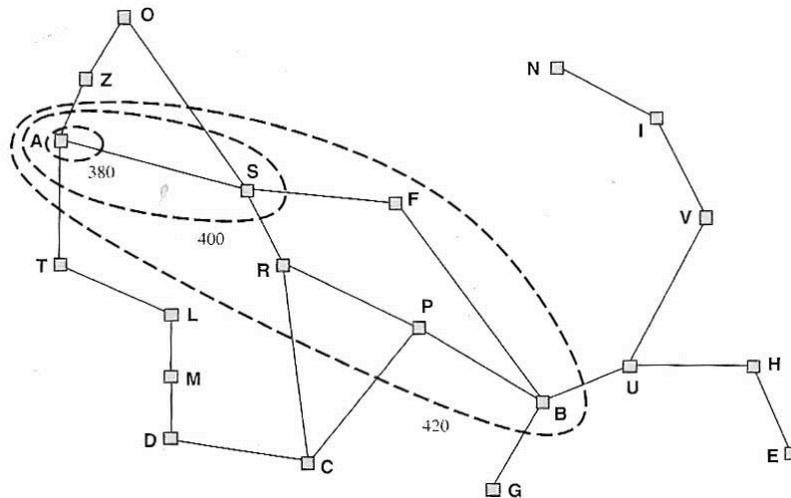
- $f(n) = \text{passato fino al nodo } n \text{ (calcolo)} + \text{futuro fino al goal (stima)}$ .
- $f(n) = g(n) + h(n)$



A\* è completa e ottima se la funzione  $h(n)$  non sovrastima il cost-to-go.  
 In questo caso  $h(n)$  è chiamata **euristica ammissibile**.



# Optimality of A\*



Soluzione ottima e ottimamente efficiente.



## Altre Euristiche



5	4	
6	1	8
7	3	2

1	2	3
8		4
7	6	5



## Sommario



Alberi di decisione

Alberi e teoria dei giochi



# Teoria dei giochi



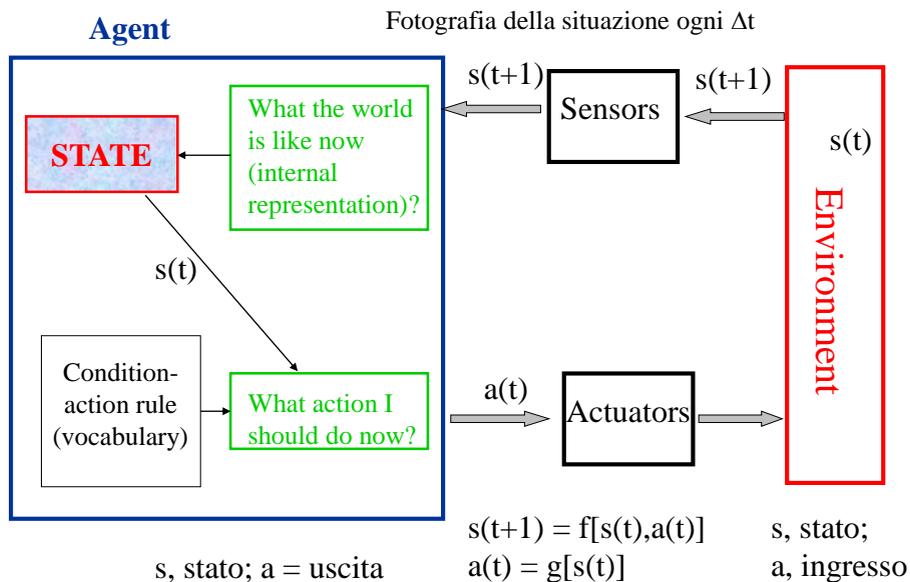
- Giochi su scacchiera: scacchi, tris. Rappresentazione digitale.
  - ◆ Stato. Insieme dei valori associati alle diverse posizioni.
  - ◆ Input. Spostamento dell'antagonista.
  - ◆ Output. Spostamento
  - ◆ Obiettivo: vincere la partita.
  - ◆ Strumento implicito: valutazione della situazione (dello stato)

Esiste una indeterminazione...

Spazio di stato enorme. Scacchi: mosse possibili in media (branching factor,  $b$ ) = 35; numero di mosse (profondità,  $d$ ) = 50. Numero di nodi,  $N = 35^{100}$ .



# Schematic diagram of an agent





# Game theory



- Game is formally defined as:
  - ◆ Initial state. Board position of the different pieces.
  - ◆ Operatori (politica), che definiscono l'insieme di mosse (azioni) che un giocatore può fare.
  - ◆ Final state (win or lose).
  - ◆ (Local) reward, at the end of the game.

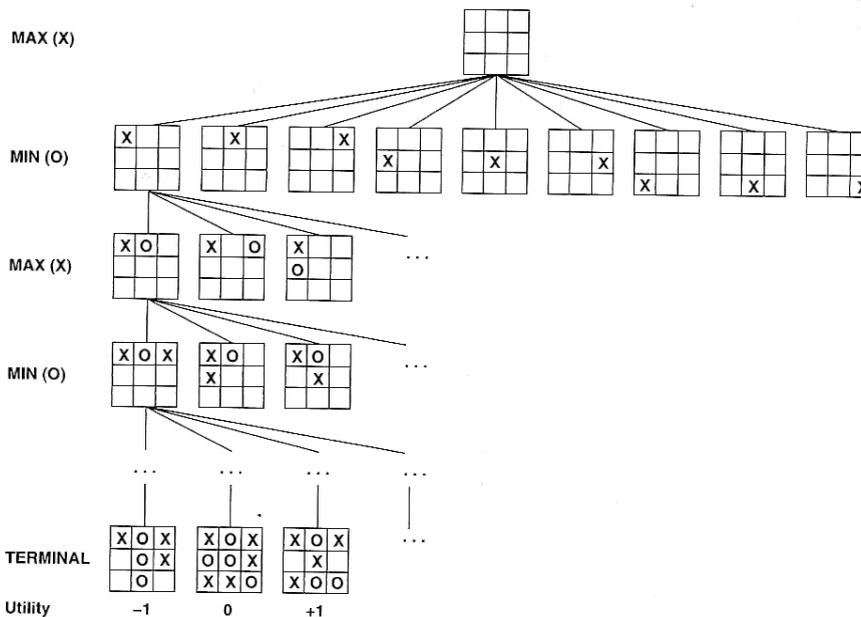
We have two players: MAX (us) and min (computer). We want to win.

We cannot apply a tree search because min has something to say. He will try to not let us win.

MAX should find a strategy (policy) that allows him winning despite min moves.



# Esempio: gioco del tris





## Min-max algorithm



Dopo avere generato l'intero albero:

- Applicare il reward locale ottenuto agli stati terminali.
- Utilizzare questo reward locale per stimare il valore degli stati immediatamente precedenti.
  - Questi stati saranno gli stati da cui è partito MAX nell'ultimo passo.
  - Negli stati precedenti verrà riportato il valore massimo degli stati foglia da cui è partito MAX, sottointendendo che MIN avrà scelto la politica ottimale per lui e quindi il reward finale sarà il minimo. Il valore dello stato precedente è quindi un **lower bound** tra i valori ottenibili possibili.
  - Si arriva alla radice.
  - Si ha la sequenza di azioni che porta al valore ottimo, **qualunque sia la scelta fatta dall'avversario**.

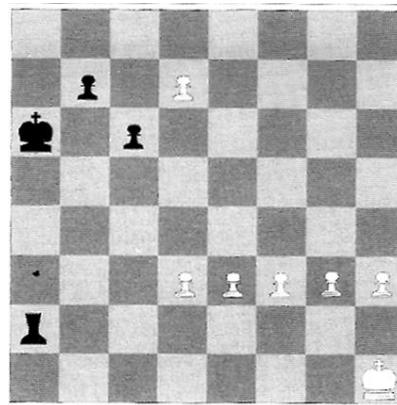
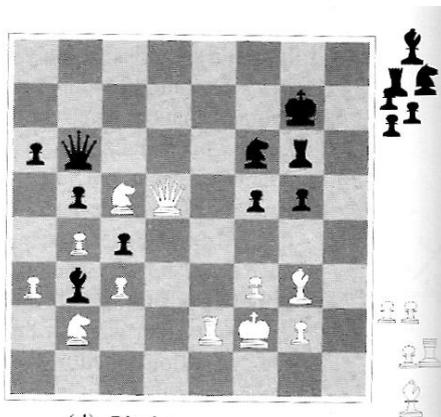
Curse of dimensionality! Too many state to expand the tree.



## Come utilizzare una possibile euristica



- Dobbiamo inserire una funzione valutazione degli stati.
  - ◆ Per gli scacchi, potrebbe essere il valore dei pezzi sulla scacchiera: tanti stati equivalenti, non sempre è un buon indicatore del valore della posizione...



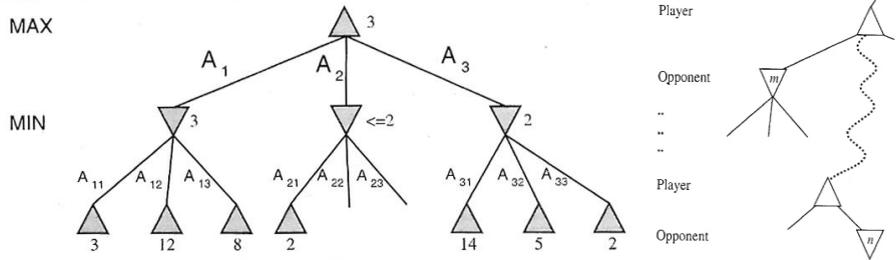
Vince il bianco anche se è in svantaggio di pezzi o apparentemente di posizione



# Alfa-Beta pruning



- Il numero di nodi da esaminare cresce con la profondità,  $d$ , dell'albero in modo esponenziale.
- Ci sono nodi, che si possono eliminare, assieme al ramo associato?

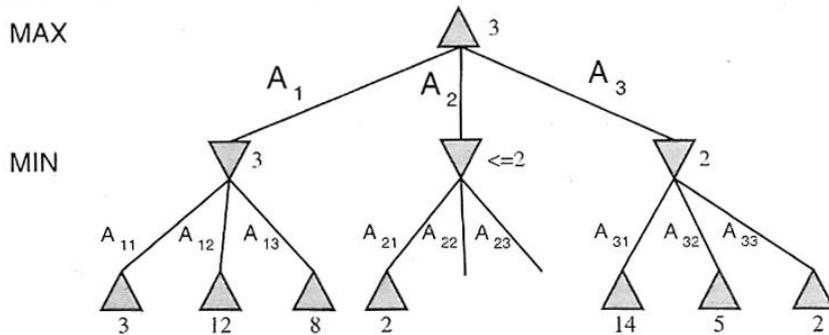


Possiamo analizzare le scelte di MIN per ogni nodo, e definire un upper bound per ogni ramo.

Controllare l'upper bound con l'ottimo, ed eventualmente effettuare il pruning.



# Osservazioni su $\alpha$ e $\beta$ pruning



Il vantaggio dipende dall'ordine dell'analisi dei nodi.

Occorrerebbe comunque analizzare l'albero per l'intera profondità (50 mosse!!).



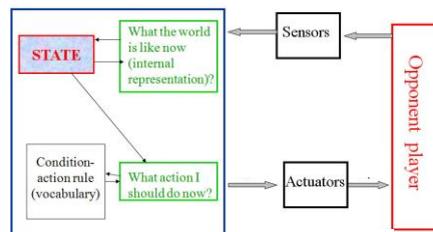
## Problemi di un agente che utilizza gli alberi di decisione



- L'avversario ha lo stesso comportamento del giocatore (vuole vincere) quindi si può modellizzare il suo comportamento e calcolare la mossa migliore che può fare in base alla funzione di valutazione.
- La strategia sviluppata funziona per il miglior avversario possibile. Non sempre l'avversario è il migliore possibile....

Quando si interagisce con un ambiente, il comportamento dell'ambiente è spesso non noto. Come possiamo fare?

La funzione di valutazione è empirica, euristica. Va definita a-priori. Quanto è adatta?



## Sommario



Alberi di decisione

Alberi e teoria dei giochi