



Laboratory for Applied Ontology

Institute of Cognitive Science and Technology
Italian National Research Council

Introduzione all'Intelligenza Artificiale


Lezione 3 - Soluzione di Problemi

Nicola Guarino
Laboratorio di Ontologia Applicata (LOA)
Istituto di Scienze e Tecnologie della Cognizione (ISTC-CNR)
Trento, Italy

www.loa.cnr.it

Sommario (Russel&Norvig cap. 3)

- Risoluzione di problemi con la ricerca
- Tipi di problemi
- Formalizzazione dei problemi
- Strategie di ricerca *cieca* delle soluzioni
- Strategie di ricerca *informata* delle soluzioni




Introduzione AI, Borge - Guarino

2

Architetture di agente

- Basati su **riflessi**
- Basati su **obiettivi** (rappresentazione del **problema**)
- Basati su **modelli** (rappresentazione del **dominio**)



Introduzione AI, Borge - Guarino

3

Agenti basati su obiettivi: Risoluzione di problemi attraverso la ricerca

- L'importanza di avere un obiettivo (non sempre è possibile...)
- Formalizzazione del problema
 - Spazio degli stati
 - Stato iniziale
 - Operatori per le transizioni di stato (da uno stato allo stato *successore*)
 - Test obiettivo
 - Costo di cammino da uno stato all'altro
- Test obiettivo
 - Esplicito: insieme di stati desiderati
 - Implicito: verifica di una certa proprietà
- Strategia di ricerca
 - Opportuno cammino nello spazio degli stati
 - In generale più cammini (con costi diversi) possono portare all'obiettivo
- Esecuzione
 - Applicazione di una sequenza di operatori allo stato iniziale

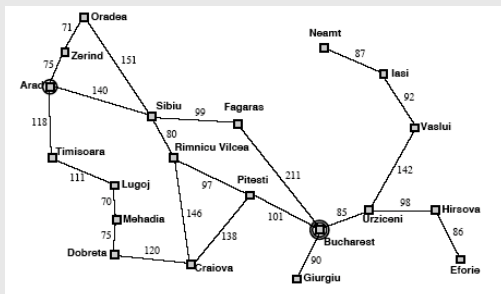


Esempio: viaggio in Romania

- In viaggio in Romania; attualmente in Arad. Il volo parte domani da Bucarest. Raggiungere Bucarest con un'auto a noleggio.
- Stati (*astratti*): presenza nelle varie città (*Ad Arad, A Bucarest...*)
- Operatori (*astratti*): I vari "viaggi minimi" che connettono una città all'altra (*Arad->Zerind, Arad->Sibiu...*)
- Test obiettivo (*esplicito*): $s = A \text{ Bucarest}$
- Costo di cammino: somma delle distanze percorse
- Soluzione: sequenza di città, per es., $\langle \text{Arad, Sibiu, Fagaras, Bucarest} \rangle$



Esempio: viaggio in Romania



Esempio: il gioco dell'otto

5	4	
6	1	8
7	3	2

1	2	3
8		4
7	6	5

Start State
Goal State

stati: coordinate delle tessere (senza posizioni intermedie)
operatori: muovere la tessera vuota a dx, sx, su, giù
test obiettivo: = stato obiettivo prefissato
costo di cammino: 1 per mossa



Esempio: il mondo dell'aspirapolvere

Spazio degli stati:

1		
3		
5		
7		

2		
4		
6		
8		

- Stati: coordinate aspirapolvere e sporco
- Test obiettivo: niente sporco
- Operatori: {dx, sx, aspira}
- Partenza: s5
- Soluzione: <dx, aspira>

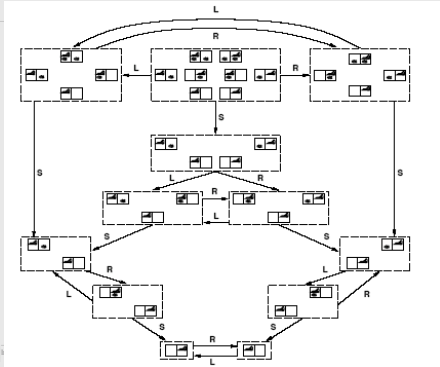


Problemi a stati multipli

- Cosa succede se il robot non ha capacita' sensoriale?
- Lo stato iniziale e' l'insieme di tutti i possibili stati
- Ogni operatore cambia l'insieme dei **possibili stati correnti**
- Test obiettivo: la certezza che l'insieme dei possibili stati correnti non contiene uno stato in cui e' presente sporco (tutti i possibili stati correnti sono stati obiettivo)



Problemi a stati multipli



Tipi di problemi

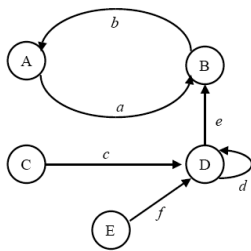
- Azioni deterministiche, stati osservabili: problema a **stati singoli**
- Stati non osservabili: problema a **stati multipli**
- Azioni non deterministiche, stati parzialmente osservabili: problema di **contingenza** (uso di sensori durante l'esecuzione)
- Spazio stati sconosciuto: problema di **esplorazione**.

Problemi "veri"...

- Pianificazione voli aerei
- Commesso viaggiatore
- Assemblaggio automatico
- Movimento robot mobili
- Progettazione di circuiti elettronici
- Progettazione di proteine...

Algoritmi di ricerca su grafi (alberi)

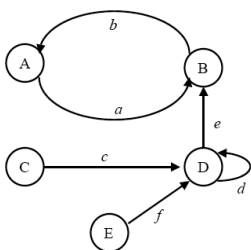
Grafi orientati



Nodi (cerchi):
Archii orientati (freccie)

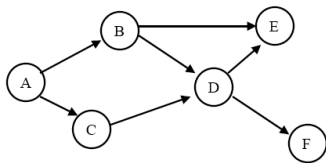
X è un nodo **successore** di Y se c'è un arco entrante in X e proveniente da Y, che si dice **predecessore** di X.

Cammini e cicli



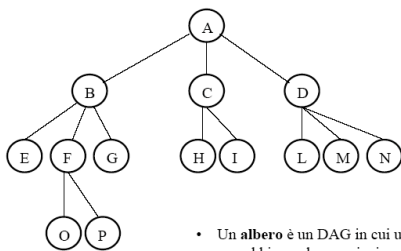
- Un **cammino** tra due nodi X e Y è una sequenza di nodi che inizia con X, termina con Y e in cui i nodi siano uno successore dell'altro.
- Se il primo e il secondo nodo coincidono, il cammino si chiama **ciclo**.
- La **lunghezza** di un cammino è data dal numero di archi che in esso compaiono.

Grafi diretti aciclici (DAG)



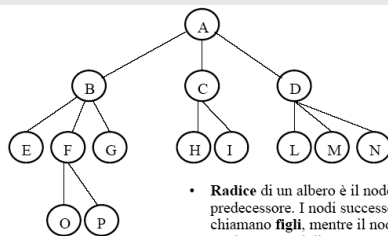
Grafo diretto senza cicli: non essendoci cicli non vi possono essere cammini infiniti.

Alberi (1)



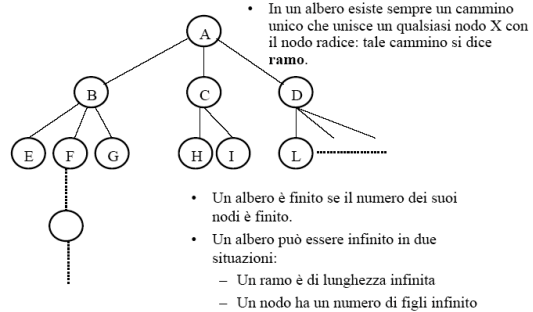
- Un **albero** è un DAG in cui un solo nodo non abbia predecessori e in cui tutti gli altri nodi abbiano un solo predecessore.

Alberi (2)



- **Radice** di un albero è il nodo senza alcun predecessore. I nodi successori si chiamano **figli**, mentre il nodo predecessore si dice **padre**. Due nodi che abbiano lo stesso padre vengono detti fratelli. Un nodo senza successori viene detto **foglia**.

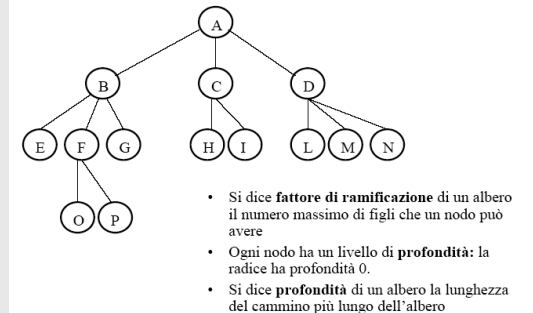
Alberi (3)



Introduzione AI, Borgo - Guarino

19

Alberi (4)



Introduzione AI, Borgo - Guarino

20

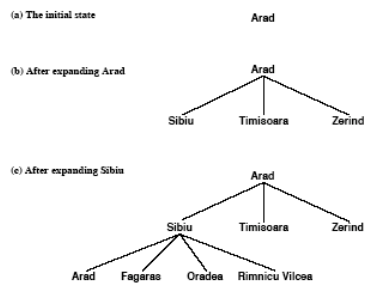
Alberi di ricerca

- Non e' conveniente (o possibile) rappresentare tutti i nodi generabili tramite l'applicazione degli operatori di transizione
- Approccio **generate and test**:
 1. dato uno stato iniziale, si **generano** tutti gli stati raggiungibili applicando i vari operatori (**espansione** dello stato)
 2. Si considerano gli stati generati secondo un'opportuna **strategia**.
 3. Se uno di questi non corrisponde all'obiettivo, viene espanso a sua volta.
- Si ottiene un **albero** in cui i nodi corrispondono agli stati (ma lo stesso stato puo' corrispondere a nodi diversi...)

Introduzione AI, Borgo - Guarino

21

Generazione/espansione di nodi



Strategie di ricerca

- Ricerca **non informata** (cieca)
 - Ricerca in profondita'
 - Ricerca in ampiezza
- Ricerca **informata** (sul costo di un cammino)
 - *Greedy* (golosa, *best-first*)
 - A
 - A*

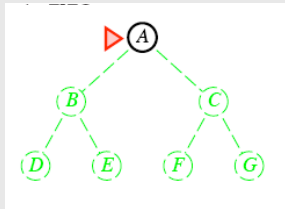


Ricerca non informata ("cieca")

- Esplorazione sistematica dello spazio degli stati
- Nessuna informazione sul costo dei cammini; tutti gli stati sono equivalenti dal punto di vista del costo.
- Solo informazione sulla struttura del problema, non del dominio
- Svantaggio: inefficienza
- Vantaggi: generalita'; baseline per il calcolo dell'efficienza
- Criteri di valutazione:
 - Completezza: si trova la soluzione, se c'e'?
 - Complessita' temporale: tempo necessario per le generazioni/espansioni
 - Complessita' spaziale: numero massimo di nodi in memoria
 - Ottimalita' della soluzione



Ricerca in ampiezza



Strategie di ricerca cieche

Strategie di ricerca *cieche* (o *non informate*) usano solo l'informazione disponibile nella definizione del problema (e non del dominio)

Ricerca in ampiezza

Ricerca a costo uniforme

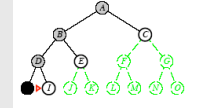
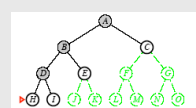
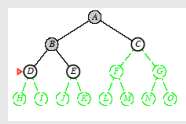
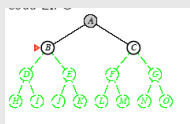
Ricerca in profondità

Ricerca a profondità limitata

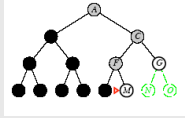
Ricerca ad approfondimento iterativo



Ricerca in profondità



Ricerca in profondita' (2)



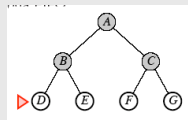
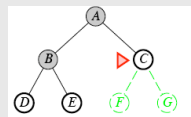
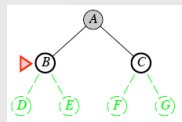
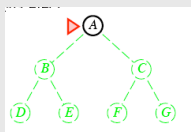
- Non garantita la completezza (possibilita' di cicli dovuti a stati ripetuti)
- Non garantita l'ottimalita'
- Tempo: r^m ; Spazio: $r \cdot m$
(r : ramificazione; m : profondita' max)



Introduzione AI, Borge - Guarino

28

Ricerca in ampiezza



Introduzione AI, Borge - Guarino

29

Ricerca in ampiezza (2)

- E' sempre in grado di trovare una soluzione, se esiste (completezza)
- Garantisce la soluzione ottima
- Tempo: r^p
- Spazio: r^p
- r : ramificazione
- p : profondita' della soluzione
- Entrambe le strategie richiedono tempo di calcolo esponenziale rispetto alla profondita' della soluzione



Introduzione AI, Borge - Guarino

30

Ricerca informata: l'uso delle *euristiche*

- Si sfrutta la conoscenza sul dominio del problema
- ...per selezionare per primi i cammini piu' promettenti
- Sulla base di una *funzione di valutazione*
 - $h(nodo)$ = costo stimato da *nodo* ad uno stato finale

