Intelligenza Artificiale

Tecniche di ricerca basate sullo spazio degli stati

Marco Piastra

Problemi diversi, soluzioni simili

- **Fox, Goat and Cabbage**
  - La definizione prevede
    - Uno stato iniziale (Farmer, Fox, Goat, Cabbage on shore1)
    - Un insieme di mosse lecite (Farmer + X move-to opposite shore)
    - Uno stato goal (Farmer, Fox, Goat, Cabbage on shore2)
  - Si tratta di trovare una **strategia** per raggiungere lo stato goal

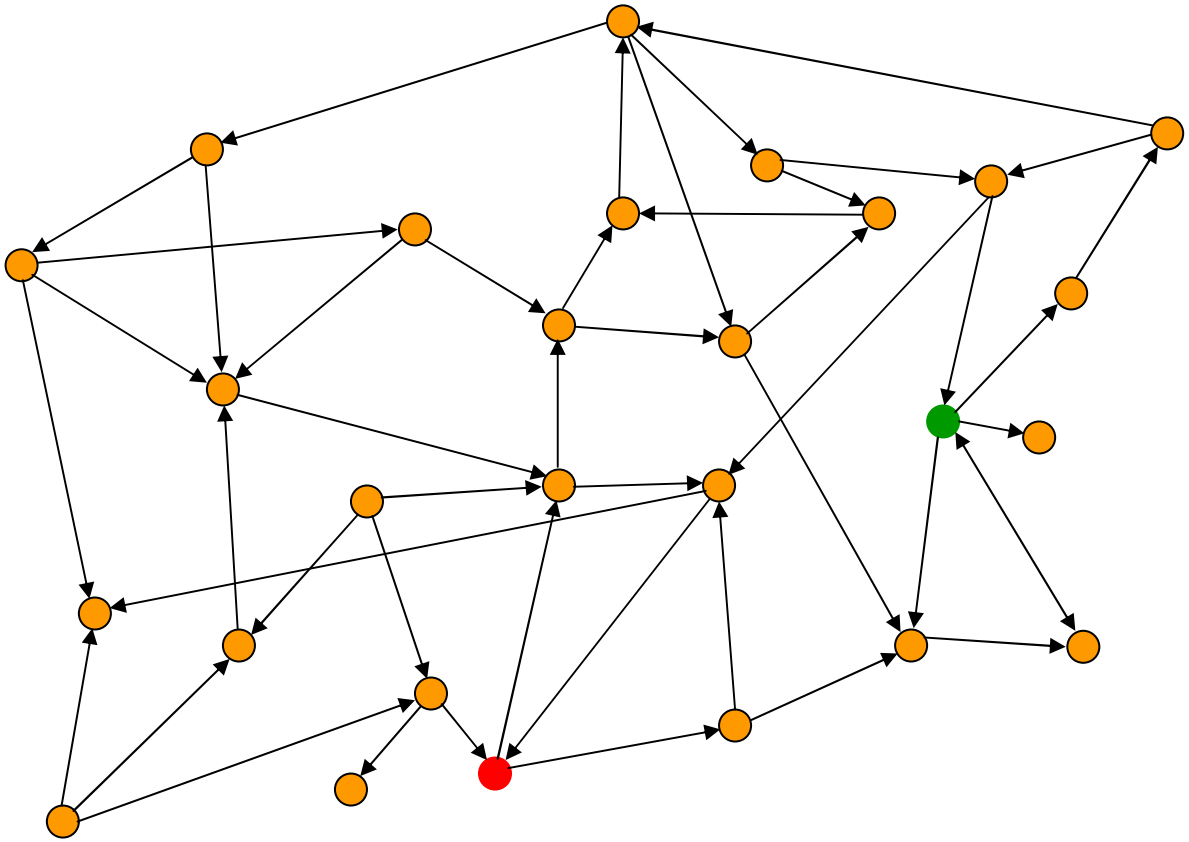
- **Einstein's Ridlle**
  - La definizione prevede
    - Una situazione ignota (4 giocatori + 4 pantaloni)
    - Informazioni parziali (su posizioni, posizioni relative e pantaloni)
  - Si tratta di indovinare la **situazione ignota**

Problemi diversi, soluzioni simili (2)

- **Fox, Goat and Cabbage**
  - Il programma analizzato (dilemma.clp) utilizza
    - Una struttura dati (status)
    - Regole di movimento (aggiungono nuovi stati)
    - Regole di eliminazione di stati
      (stati terminali – qualcuno viene mangiato, stati ciclici, goal)

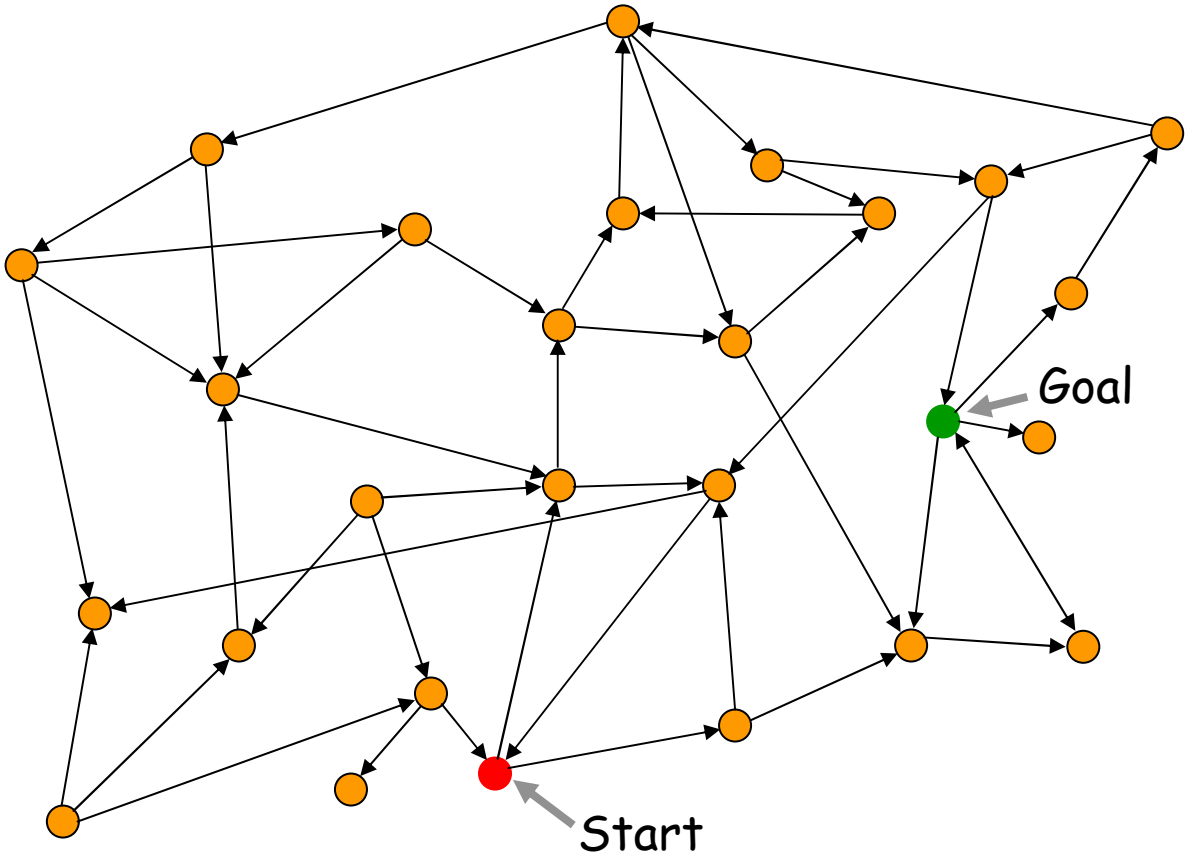
- **Einstein's Ridlle**
  - Il programma realizzato (se avete fatto in questo modo) utilizza
    - Una struttura dati (informazione nulla)
    - Regole che aggiungono informazioni (aggiungono nuovi stati)
    - Regole di eliminazione di stati
      (violazione di uno più vincoli,
      che rappresentano le informazioni parziali disponibili)

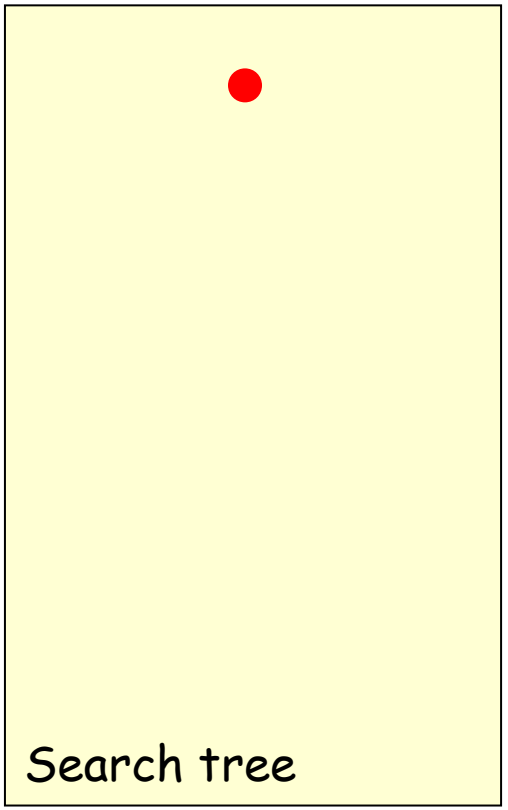
Ricerca nello spazio degli stati (disegno di J.C. Latombe)



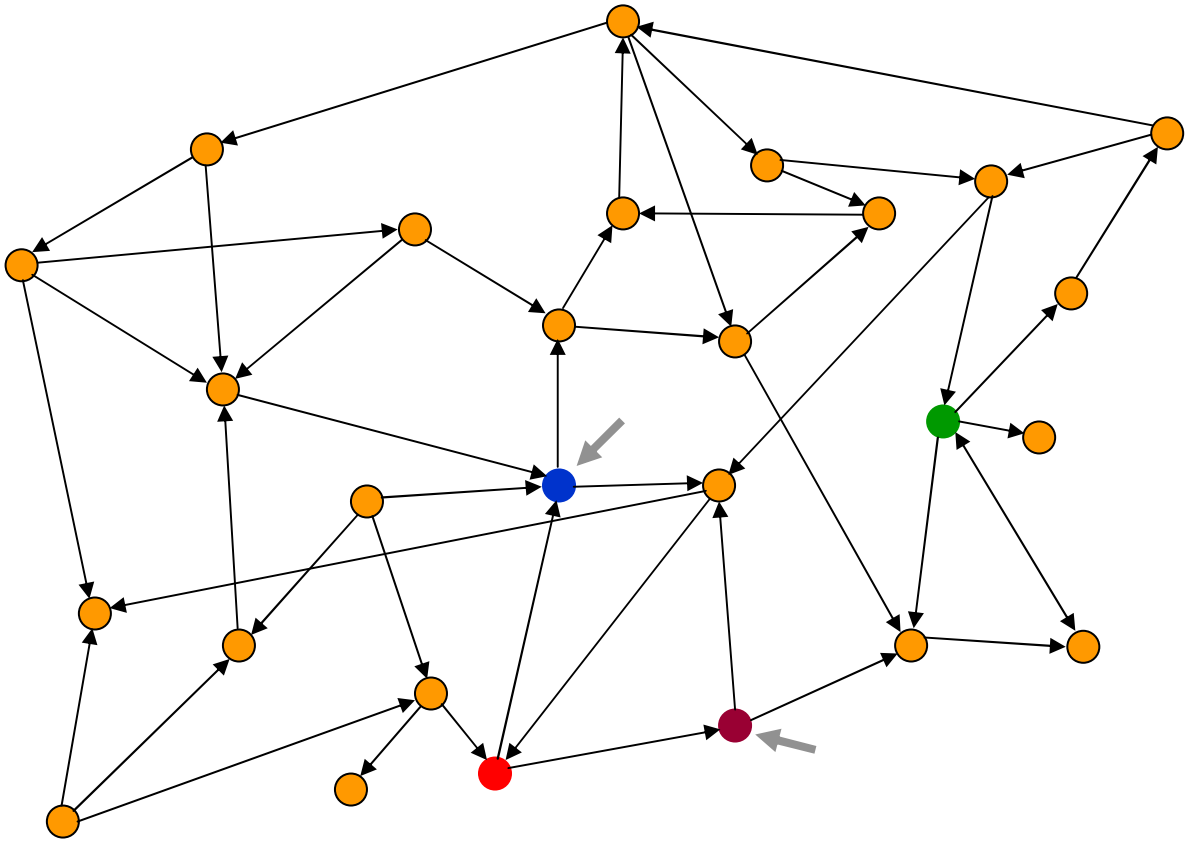
- I nodi rappresentano uno **stato**
- Gli archi (orientati) una **transizione** (potrebbero avere un **costo**)
- La tecnica consiste nell'esplorazione del grafo
- Non sempre è possibile/necessario costruire il grafo completo
- La tecnica deve essere efficiente

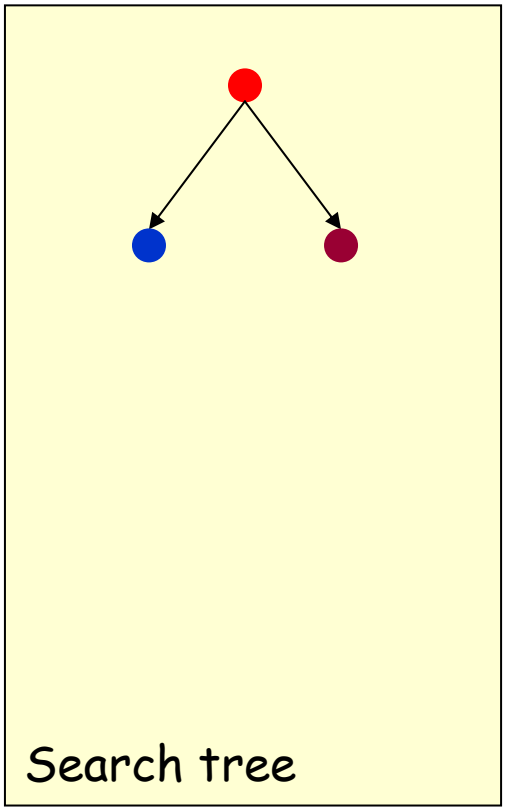
Ricerca nello spazio degli stati (disegno di J.C. Latombe)



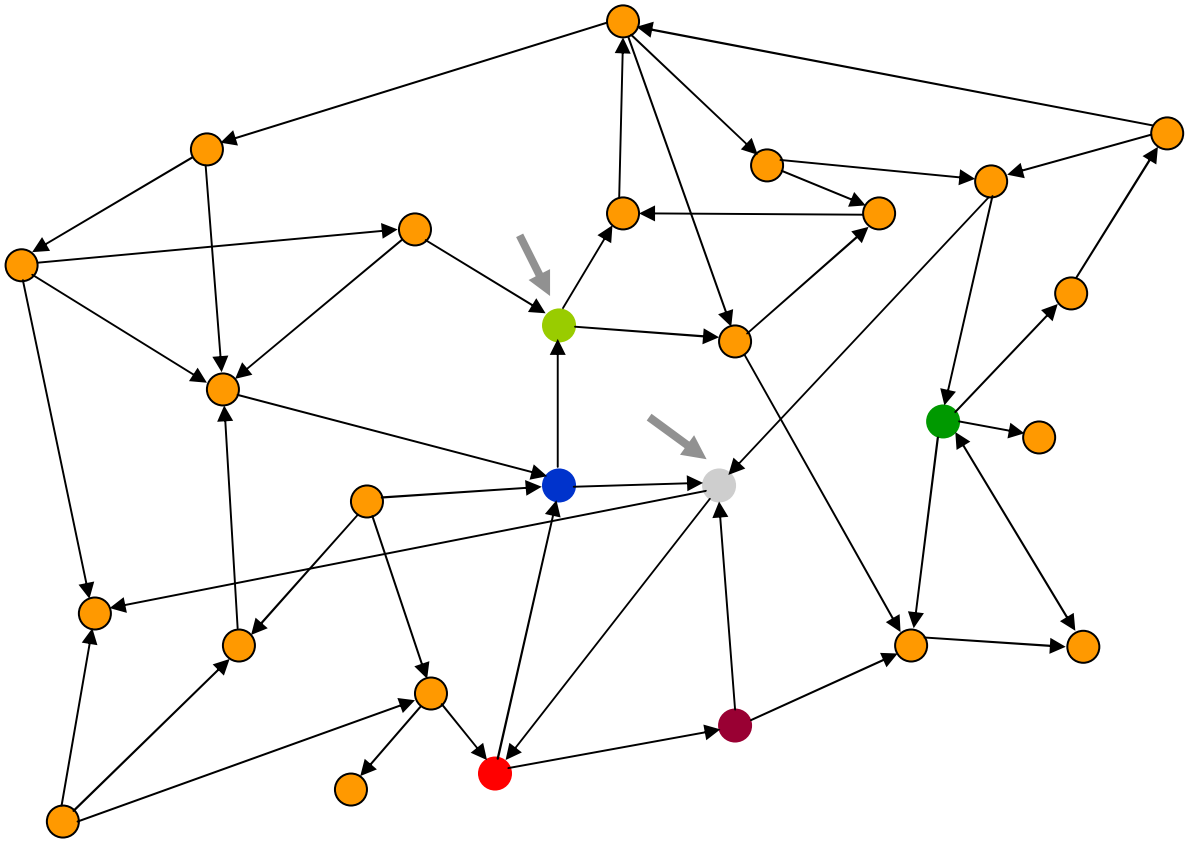


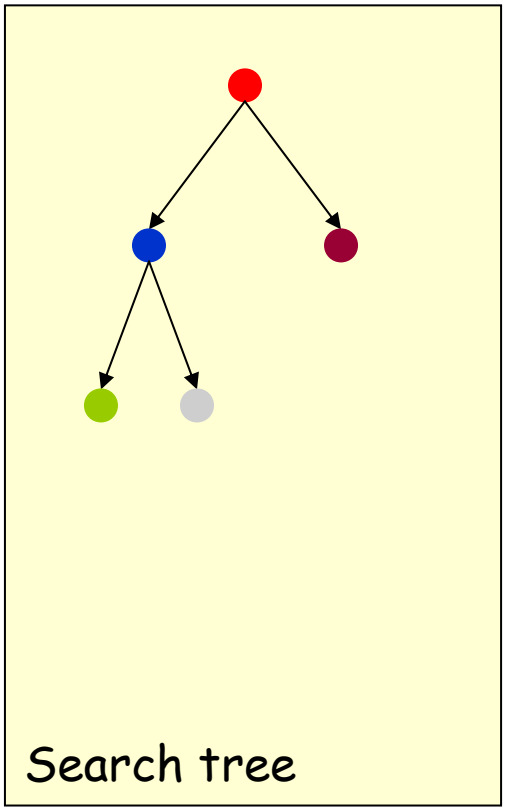
Ricerca nello spazio degli stati (disegno di J.C. Latombe)



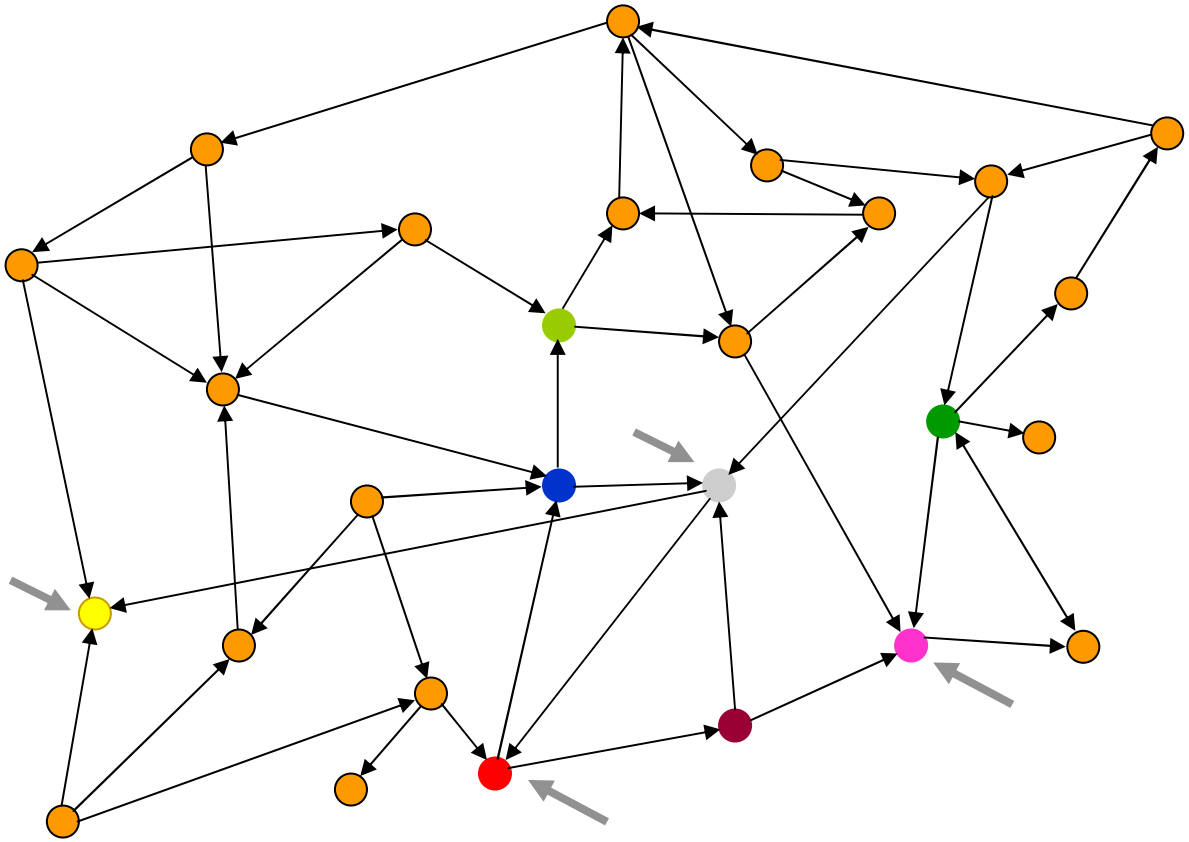


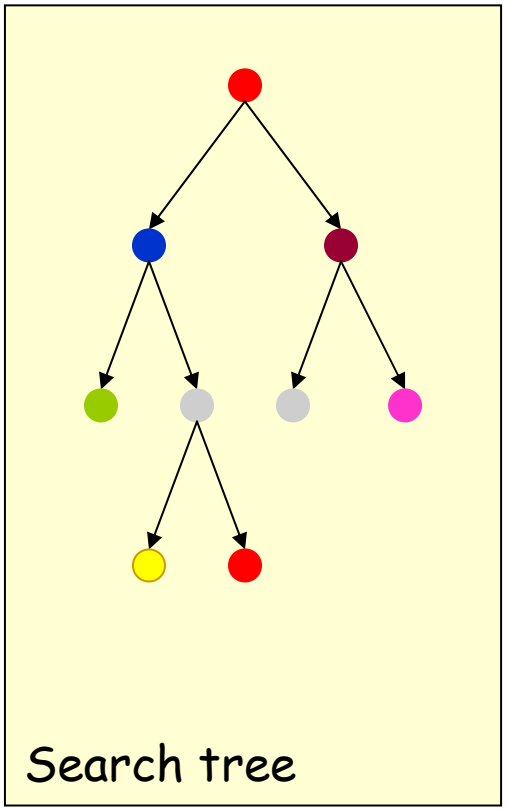
Ricerca nello spazio degli stati (disegno di J.C. Latombe)



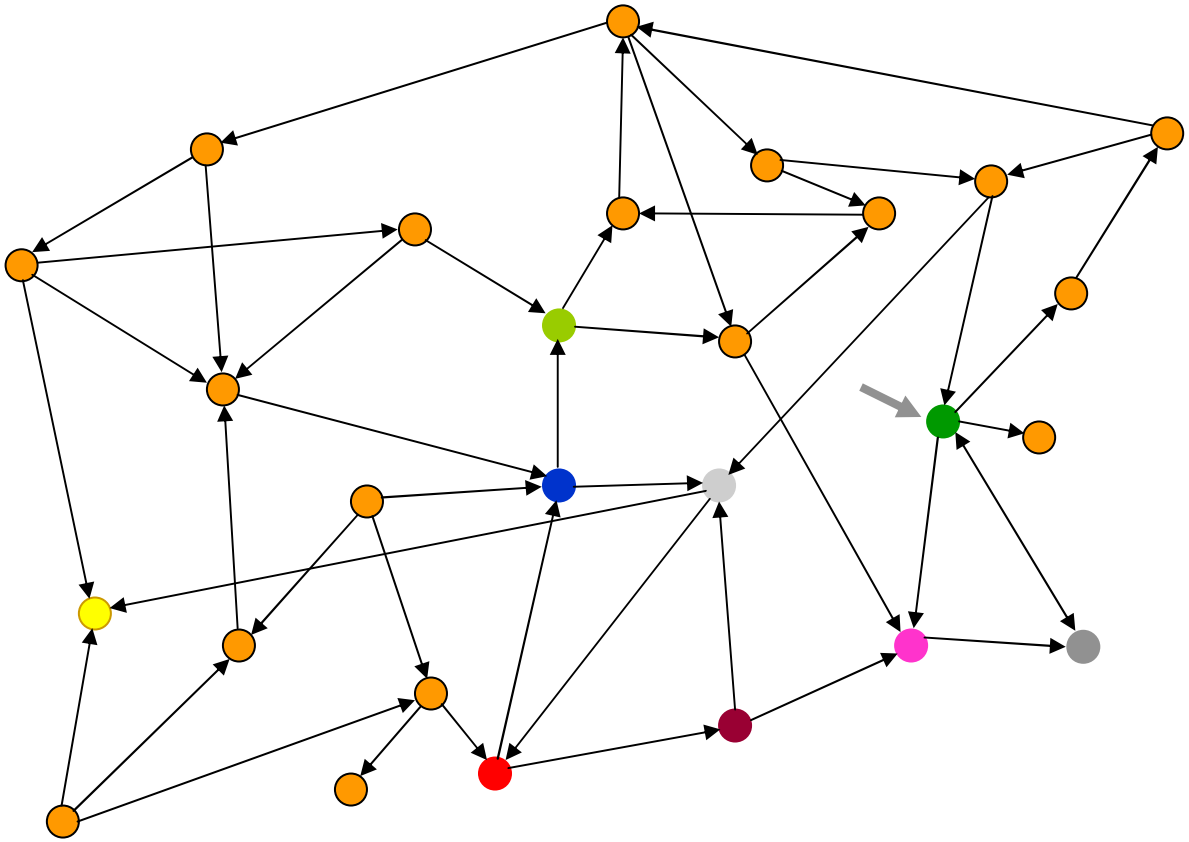


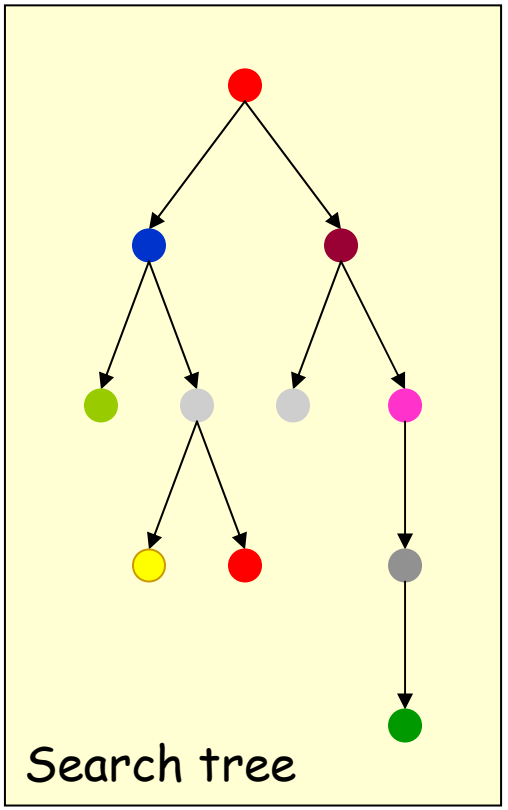
Ricerca nello spazio degli stati (disegno di J.C. Latombe)



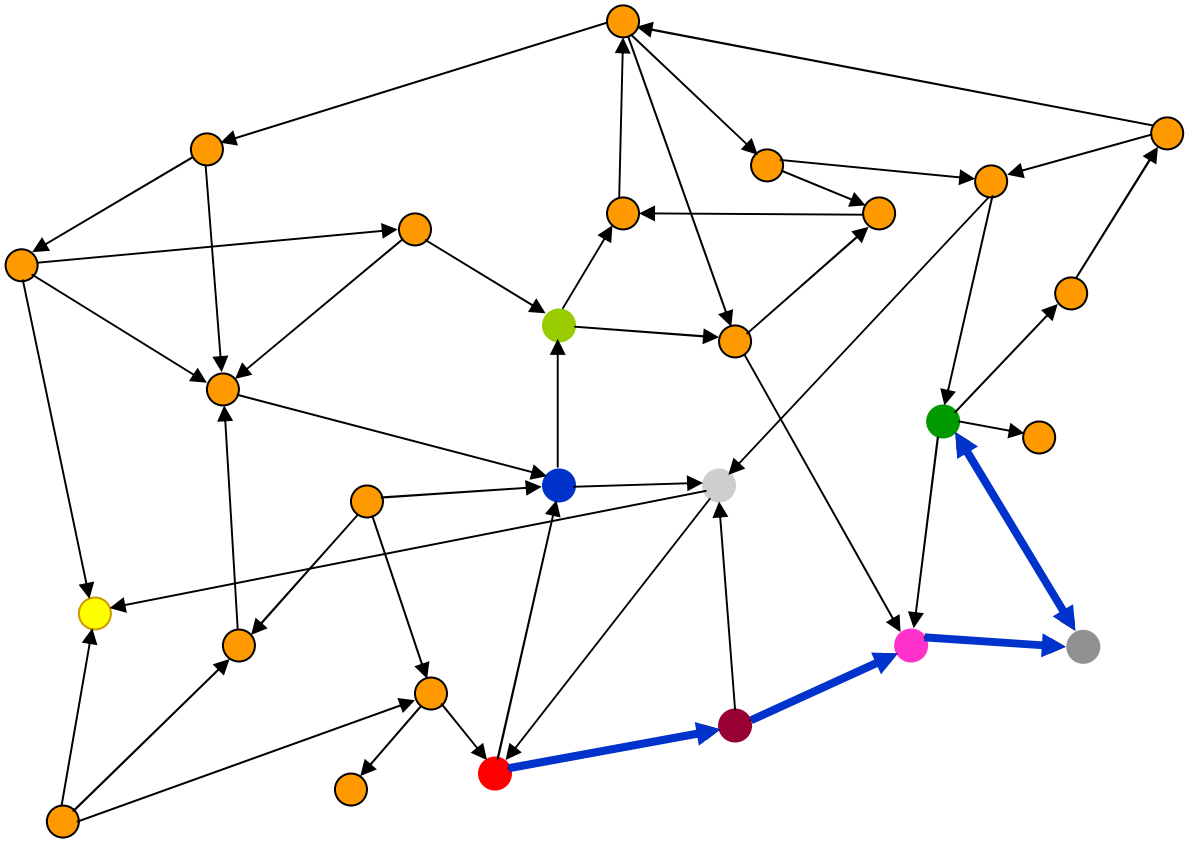


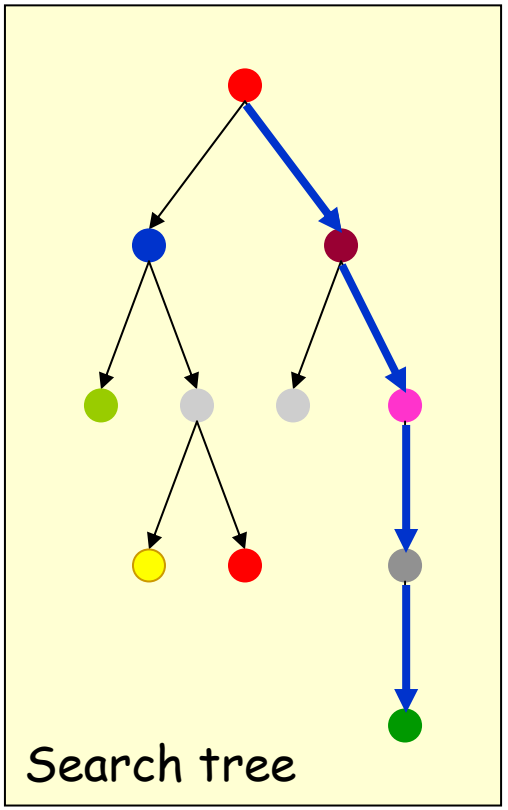
Ricerca nello spazio degli stati (disegno di J.C. Latombe)





Ricerca nello spazio degli stati (disegno di J.C. Latombe)





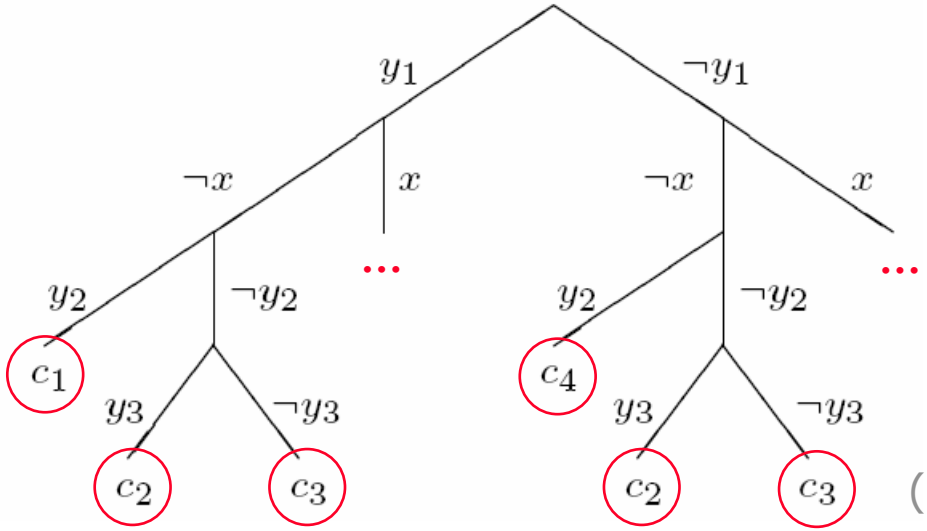
Ricerca nello spazio degli stati

- Stati e transizioni
  - **Stato**: s
    (vale a dire, cosa rappresenta in relazione al problema)
    - Può essere una situazione o uno stato di informazione
  - **Spazio degli stati**: S
    - Insieme di tutti gli stati possibili
  - **Successori** di ciascuno stato: Successors(s)
    - $s \in S \Rightarrow \text{Successors}(s) \in 2^S$
- Algoritmo di ricerca (algoritmo base)
  - Parte da uno stato iniziale s_0
  - Genera un **albero**
    - A partire da uno stato corrente s_c
    - Genera l'insieme Successors(s_c)
  - Terminazione (due possibilità)
    - Viene raggiunto lo stato goal (basta una soluzione)
    - Viene completata la ricerca nell'intero spazio degli stati
      (si cercano tutte le soluzioni)

Un problema tipico: SAT

- Mostrare che una formula logica proposizionale è soddisfacibile
  - Stato s: assegnazioni di valore di verità a variabili proposizionali
    (s_0 è lo stato nullo: nessuna assegnazione)
  - Successors(s): possibili assegnazioni ad una variabile
    (nell'esempio sotto, si usa l'ordine di apparizione)
  - Vincoli: le clausole c_1, c_2, c_3, c_4 (devono essere tutte soddisfatte)

$$\underbrace{(\neg y_1 \vee x \vee \neg y_2)}_{c_1} \wedge \underbrace{(y_2 \vee \neg y_3)}_{c_2} \wedge \underbrace{(y_2 \vee y_3)}_{c_3} \wedge \underbrace{(y_1 \vee \neg x \vee \neg y_2)}_{c_4}$$



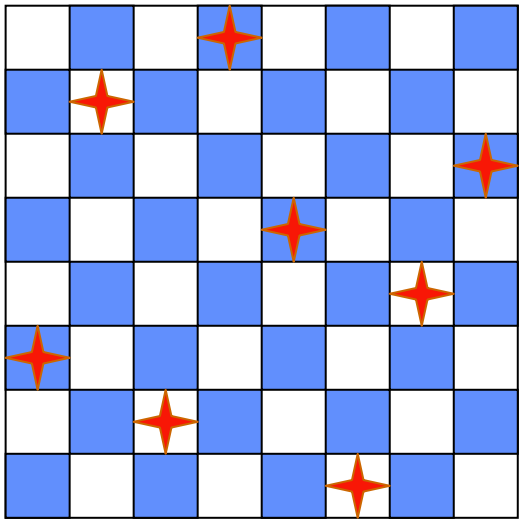
(La formula non è soddisfacibile)

Problemi e spazi degli stati

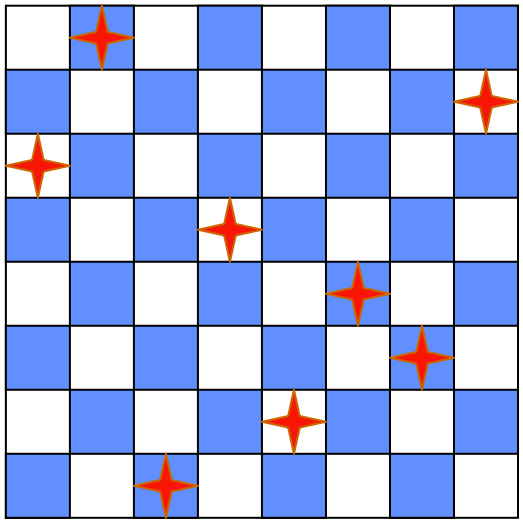
- Due aspetti fondamentali per l'applicazione della tecnica di ricerca nello spazio degli stati ai problemi:
  1) Codifica del problema in uno spazio degli stati
     - Definizione di stati e successori
  2) Uso di un algoritmo di ricerca nello spazio degli stati
     - L'algoritmo base genera un albero secondo una strategia *top-down* (dal nodo radice verso le foglie)
       - Non segue un ordine particolare di generazione
       - A seconda del tipo di problema, esplora o meno lo spazio in modo esaustivo

Codifica dello spazio degli stati

- Problema: le otto regine
  - Piazzare 8 regine su una scacchiera 8 x 8 in modo che nessuna possa attaccare l'altra

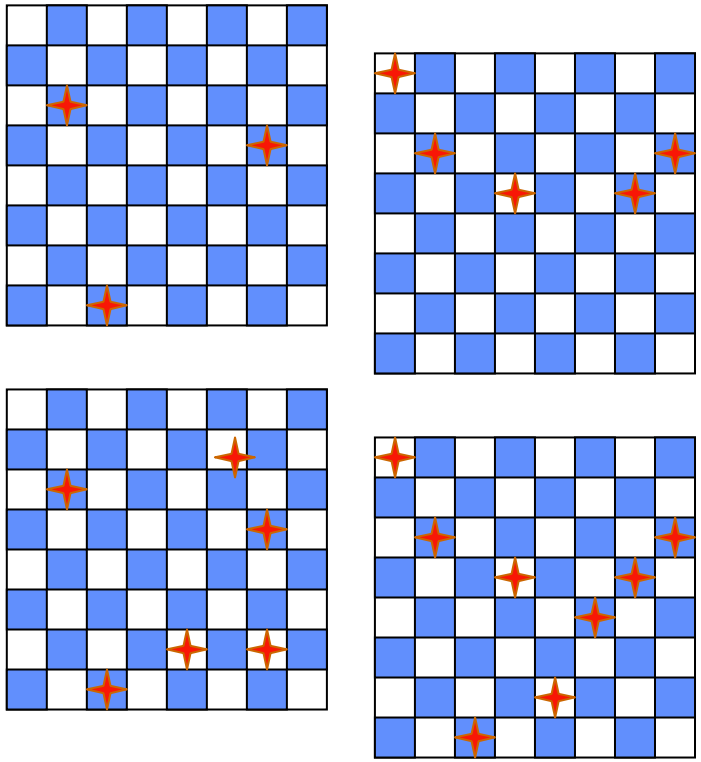


Una soluzione



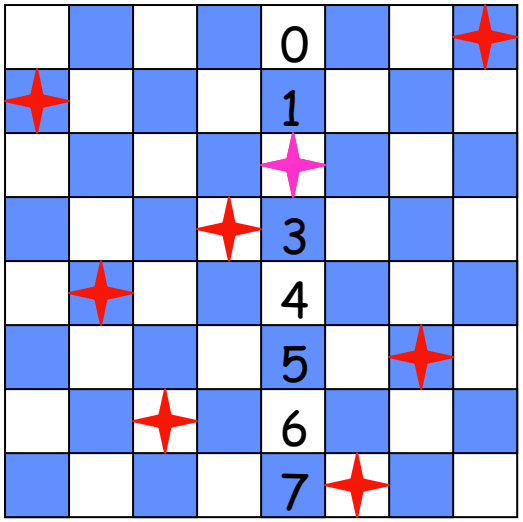
Una non-soluzione

Codifica dello spazio degli stati (2)



- Prima ipotesi di rappresentazione
  - Stati: tutte le disposizioni di 0, 1, 2, 3, 4, 5, 6, 7, 8 regine sulla scacchiera
  - Stato iniziale: 0 regine sulla scacchiera
  - Successors(s): tutti gli stati che si possono ottenere da s aggiungendo una regina
  - Goal (non unico): stato in cui 8 regine non possono attaccarsi
- Complessità:
  - Circa 3×10^{14} stati …

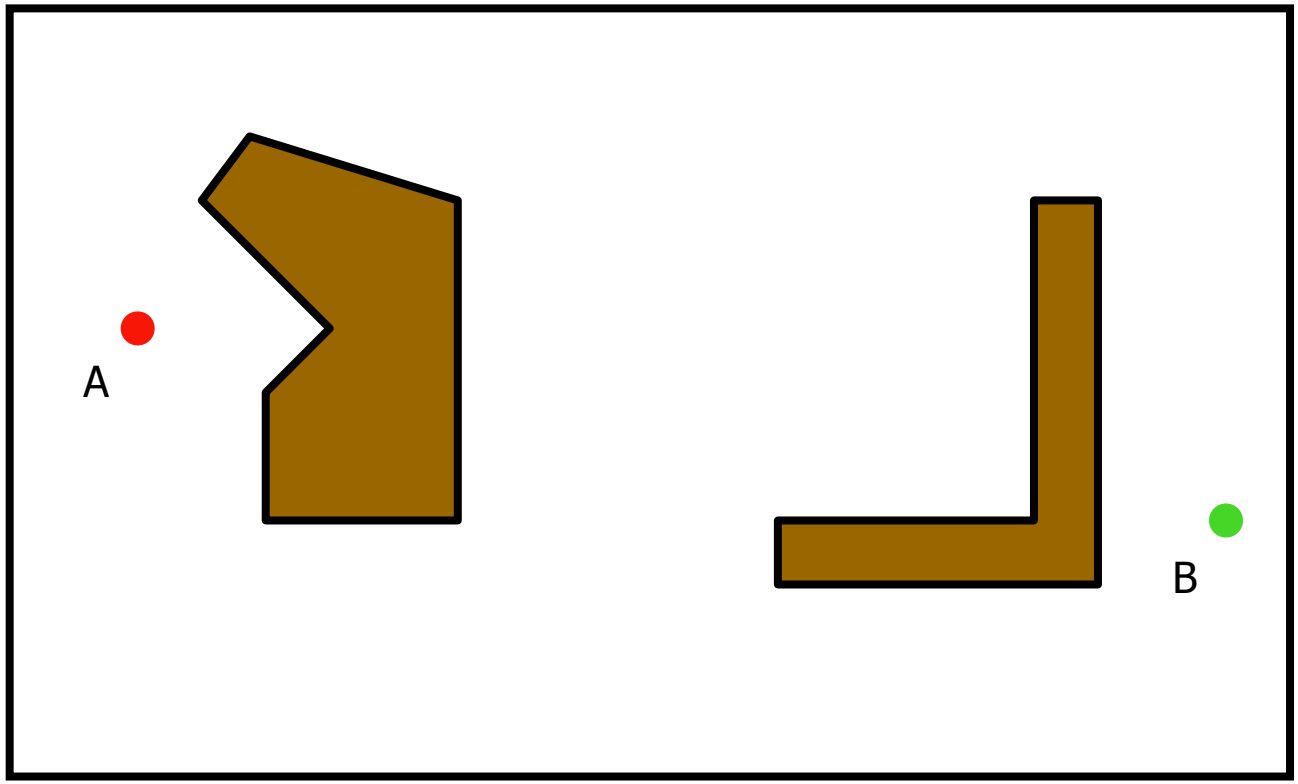
Codifica dello spazio degli stati (2)



- Seconda ipotesi di rappresentazione
  - (Due regine nella stessa colonna possono attaccarsi)
  - Stati: tutte le disposizioni di 8 regine in cui ciascuna regina occupa una colonna
  - Stato iniziale: casuale
  - Successors(s): tutti gli stati che si possono ottenere da s spostando una regina nella propria colonna
- Complessità:
  - Circa 1.6 x 10^7 stati
  - La prima ipotesi era generale (andava bene per il problema delle n regine, con $n \leq 8$)
  - La seconda ipotesi sfrutta meglio i requisiti del problema

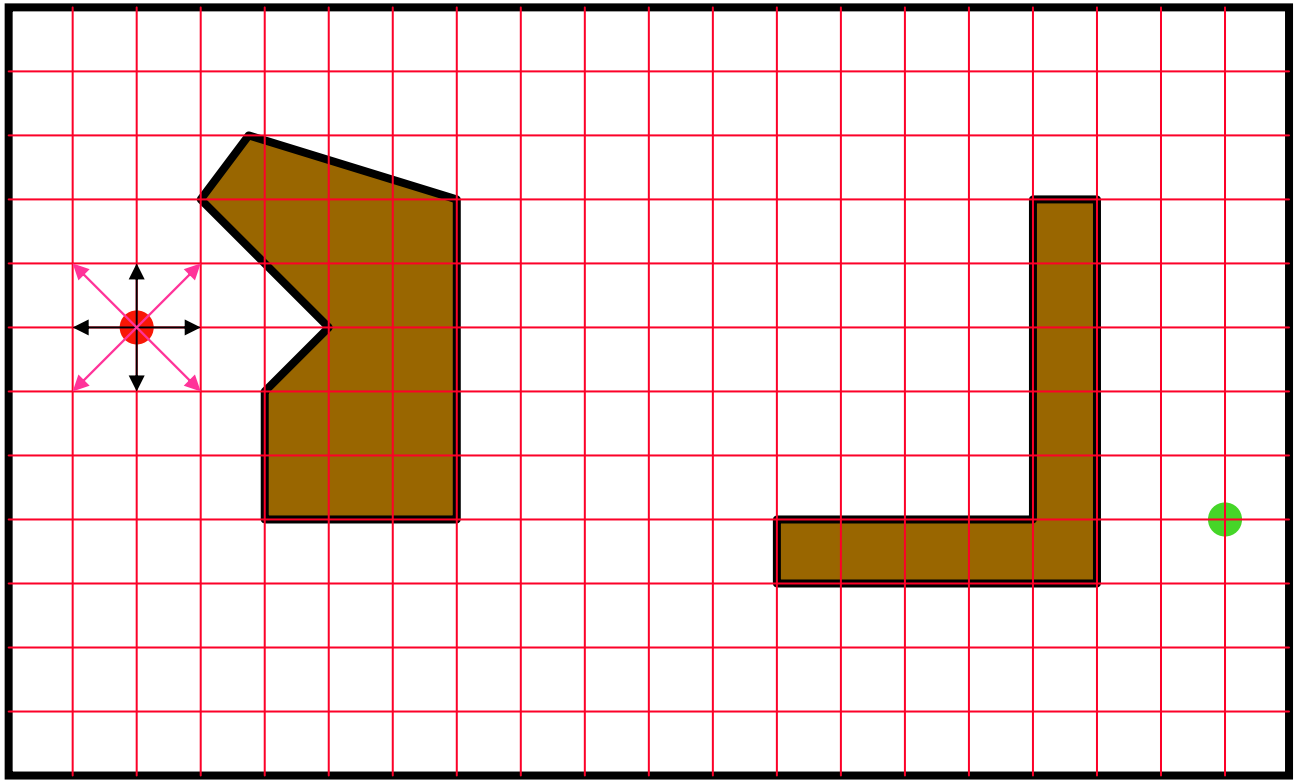
Pianificazione come ricerca (disegno di J.C. Latombe)

– Un robot deve andare da A a B, in una stanza con ostacoli



Pianificazione come ricerca (disegno di J.C. Latombe)

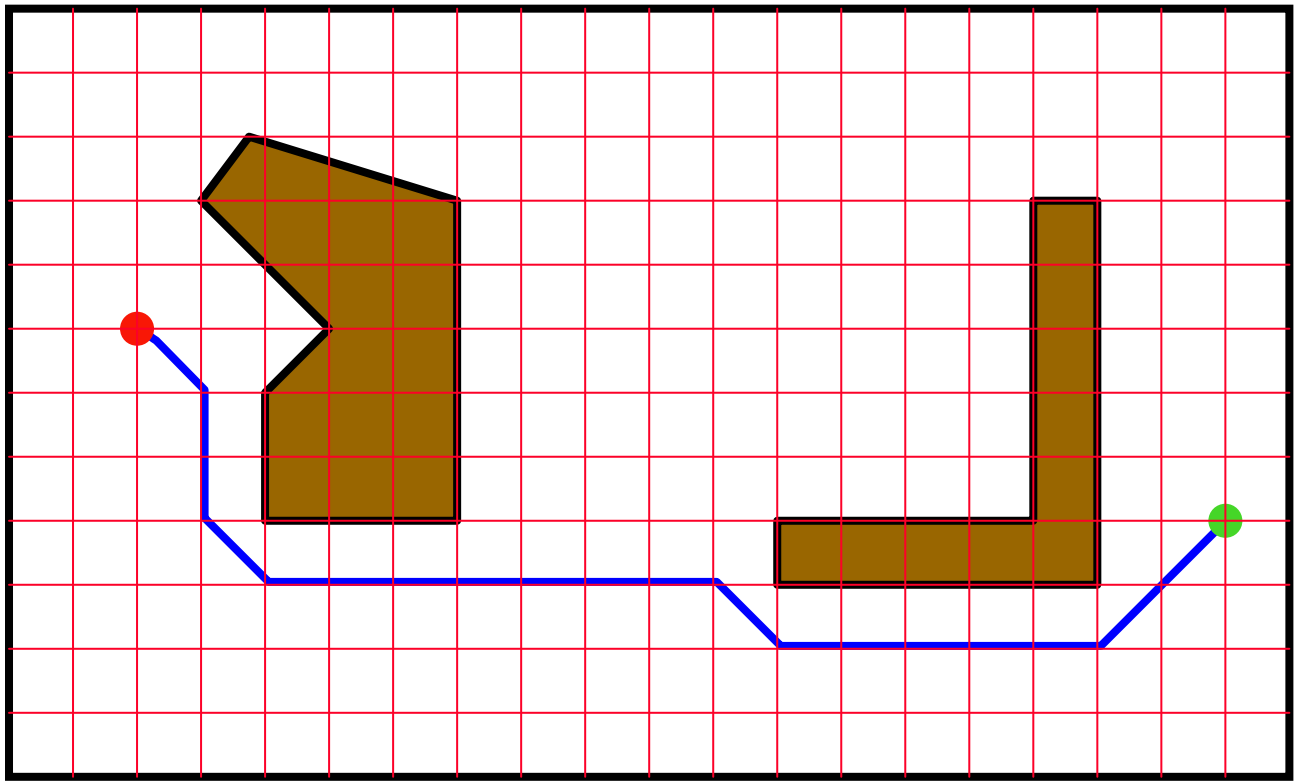
– Stati: punti di una griglia discreta 19 x 11 (~ 2 x 10^2)



– Costi: movimento orizzontale o verticale 1, diagonale $\sqrt{2}$

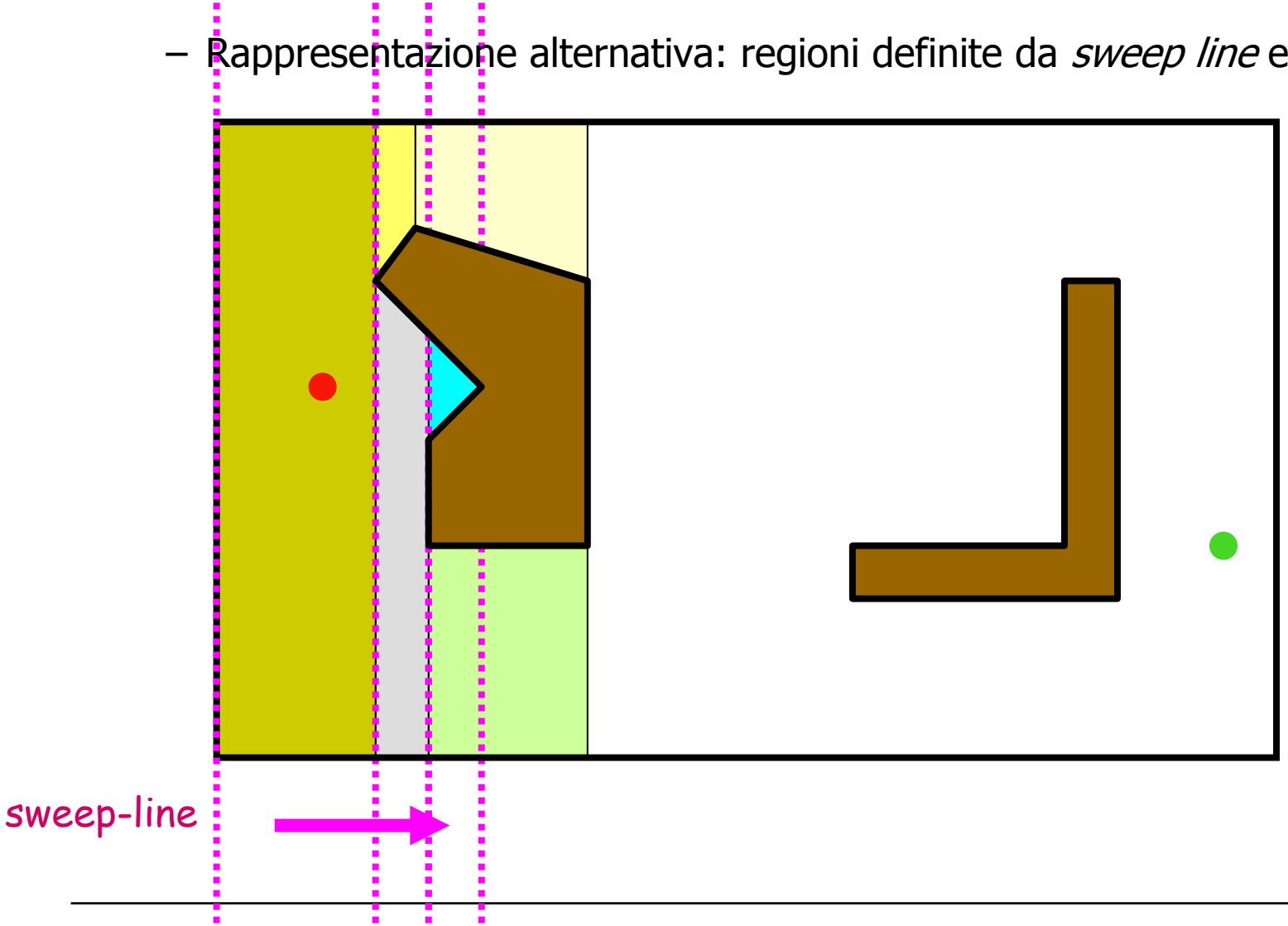
Pianificazione come ricerca (disegno di J.C. Latombe)

- Soluzione ottimale (data la discretizzazione)



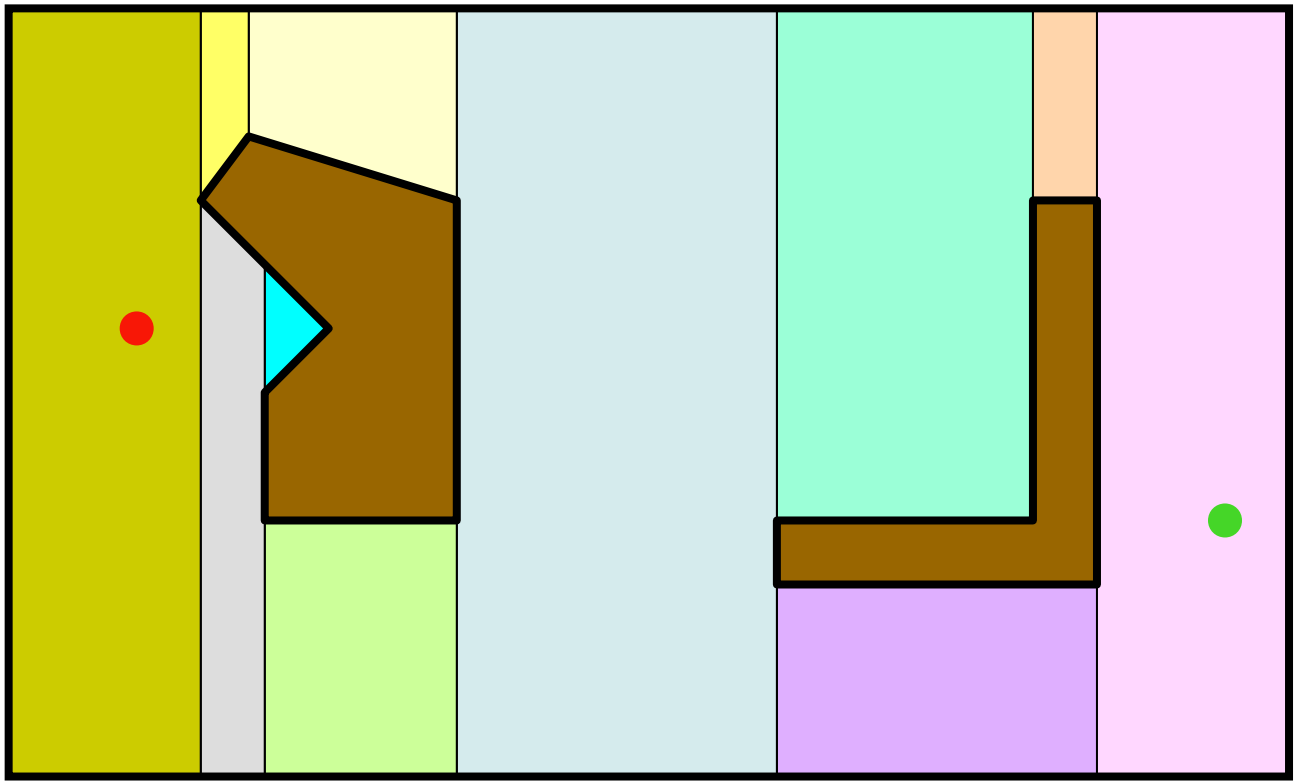
Pianificazione come ricerca (disegno di J.C. Latombe)

– Rappresentazione alternativa: regioni definite da *sweep line* e singolarità



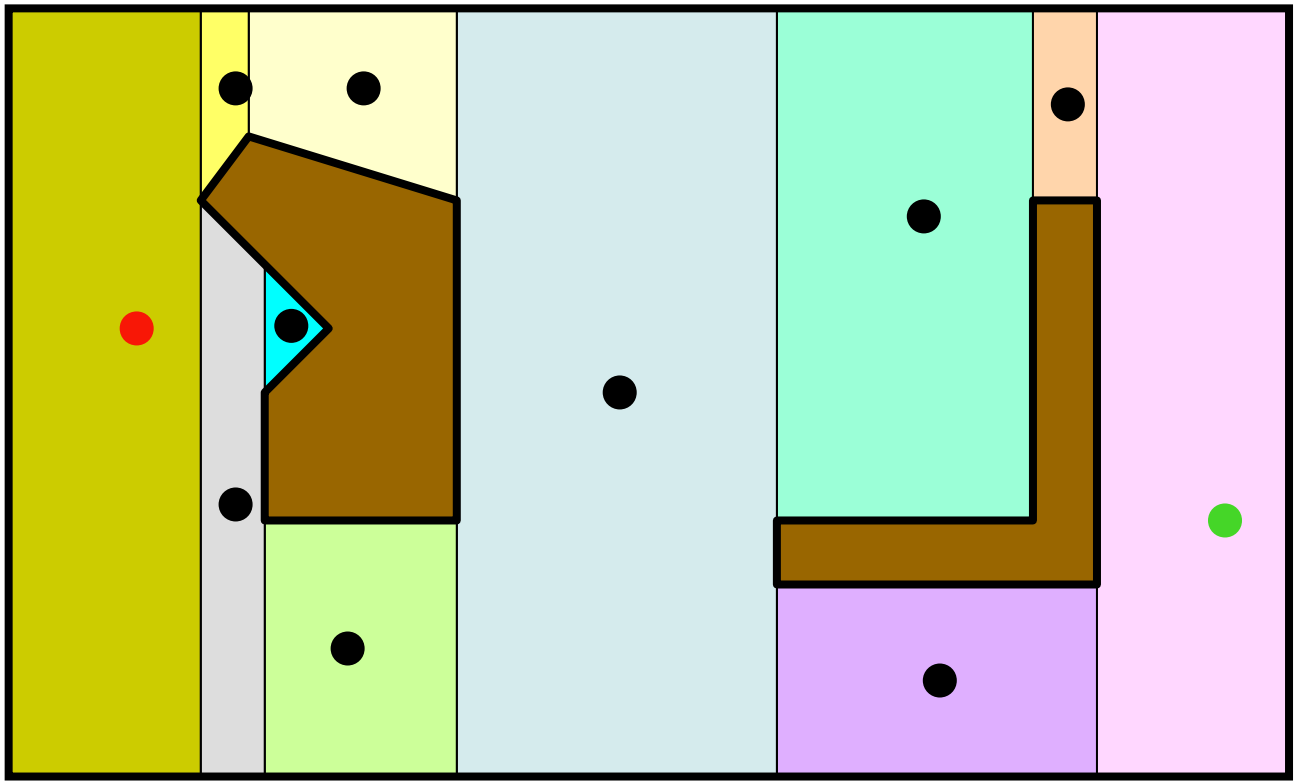
Pianificazione come ricerca (disegno di J.C. Latombe)

– Regioni



Pianificazione come ricerca (disegno di J.C. Latombe)

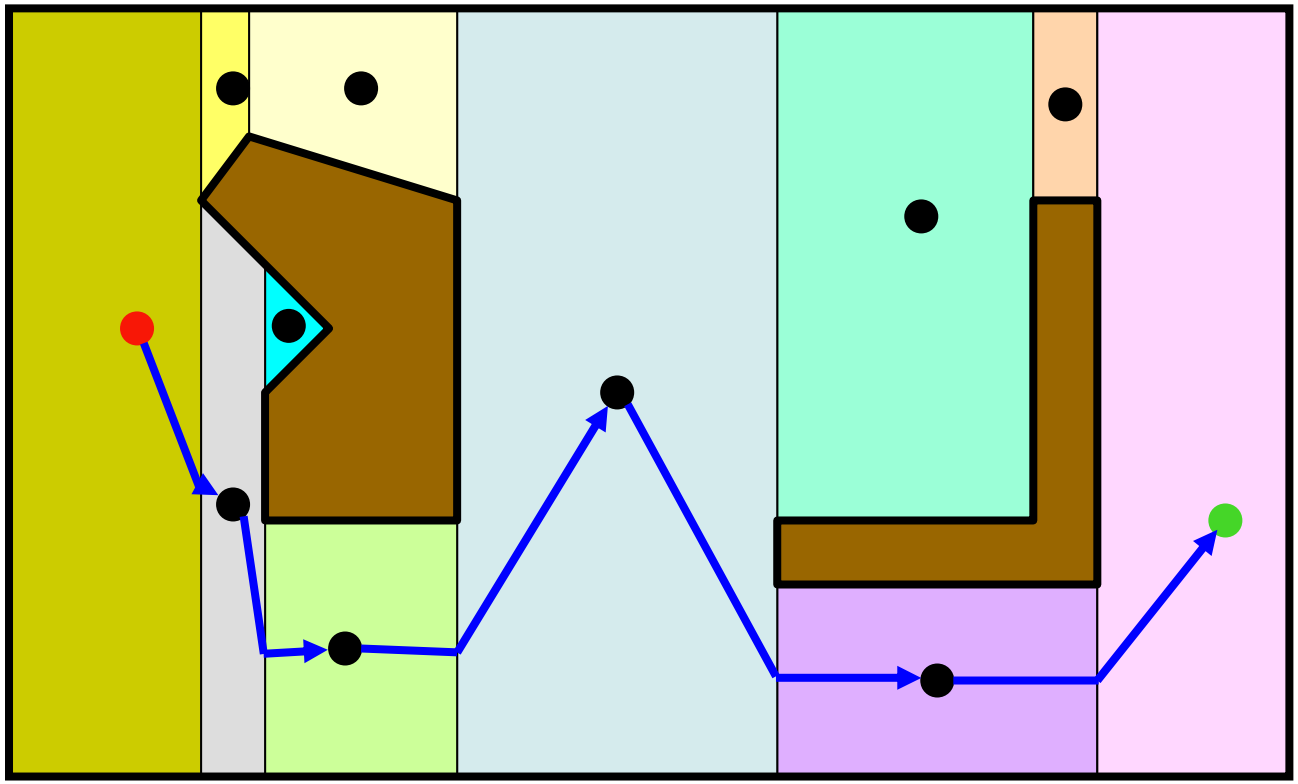
– Stati: il centroide di ogni regione (11 stati)



– Costi: distanza tra i centroidi

Pianificazione come ricerca (disegno di J.C. Latombe)

- Soluzione (approssimata)



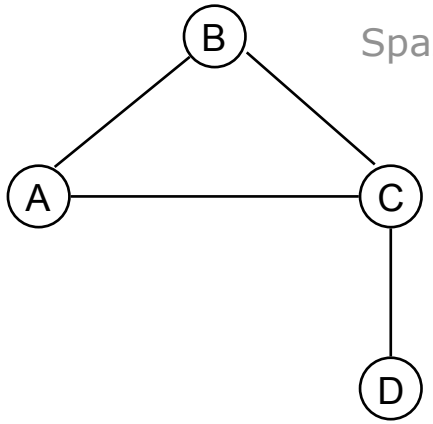
- Occorre un'ulteriore passo di affinamento, per accorciare il percorso

Problemi come ricerca nello spazio degli stati

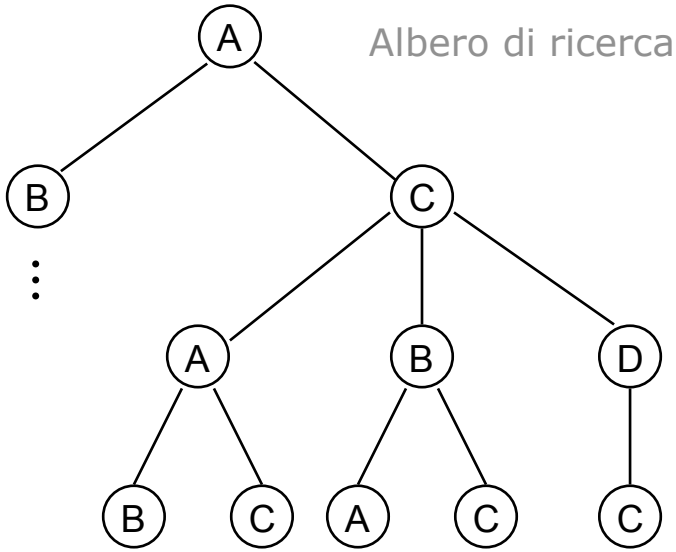
- Molteplicità
  - A stato singolo: un solo stato corrente
  - A stati multipli: più stati correnti
    (p.es. più agenti o incertezza rispetto allo stato iniziale)
- Ricerca soluzioni
  - Contingente: una sola soluzione (ottimale o meno)
  - Esaustiva: tutte le soluzioni
- Variabili
  - Stati: tipicamente rappresentati in modo parziale, con struttura aggiuntiva (albero)
  - Stato corrente: uno o più
  - Risultato delle azioni: p.es. Successors(s), dove s è lo stato corrente

Strategie di ricerca

- Definizioni



Spazio degli stati



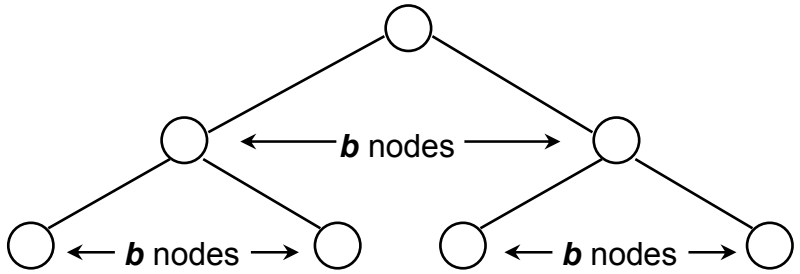
Albero di ricerca

  - Albero di ricerca
  - Nodo (dell'albero)
  - Nodo radice
  - Nodo genitore
  - Profondità

Strategie di ricerca (2)

- Parametri di valutazione
  - Completezza: è certo che si trovi una soluzione?
    (o tutte le soluzioni, se necessario)
  - Ottimalità: in presenza di diverse soluzioni, verrà trovata la soluzione migliore?
  - Time complexity: quanto tempo sarà necessario?
  - Space complexity: quanta memoria sarà necessaria?

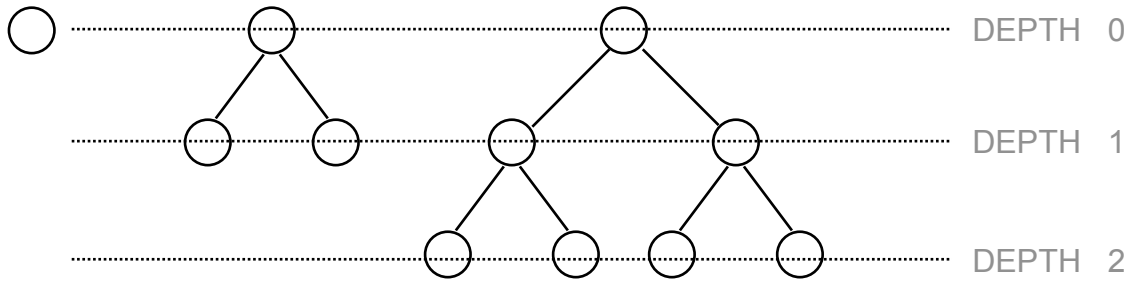
  - Branching factor (b):
    fattore di biforcazione degli archi



$$1 + b + b^2 + b^3 + \ldots + b^d$$

Strategie di ricerca (3)

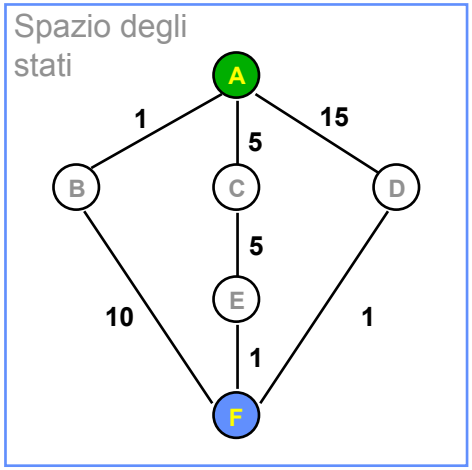
- **Strategia breadth-first** (ricerca in ampiezza)
  - Si espandono tutti i nodi di un livello prima di passare al successivo

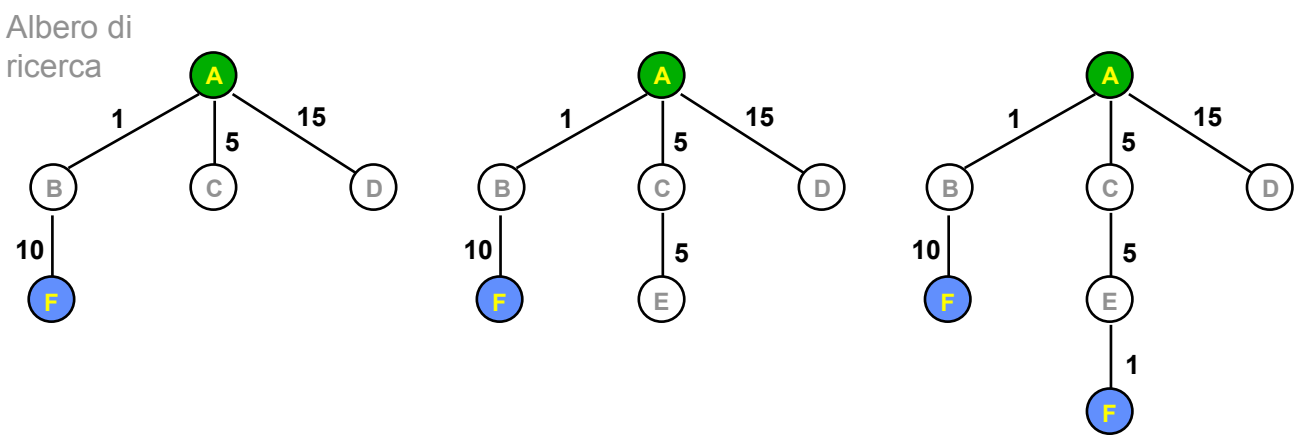


  - Completa
  - Ottimale
  - Time complexity: $O(b^d)$
  - Space complexity: $O(b^d)$

Strategie di ricerca (4)

- **Strategia a costo uniforme** (simile a breadth-first)
  - Si privilegiano i cammini a costo complessivo più basso

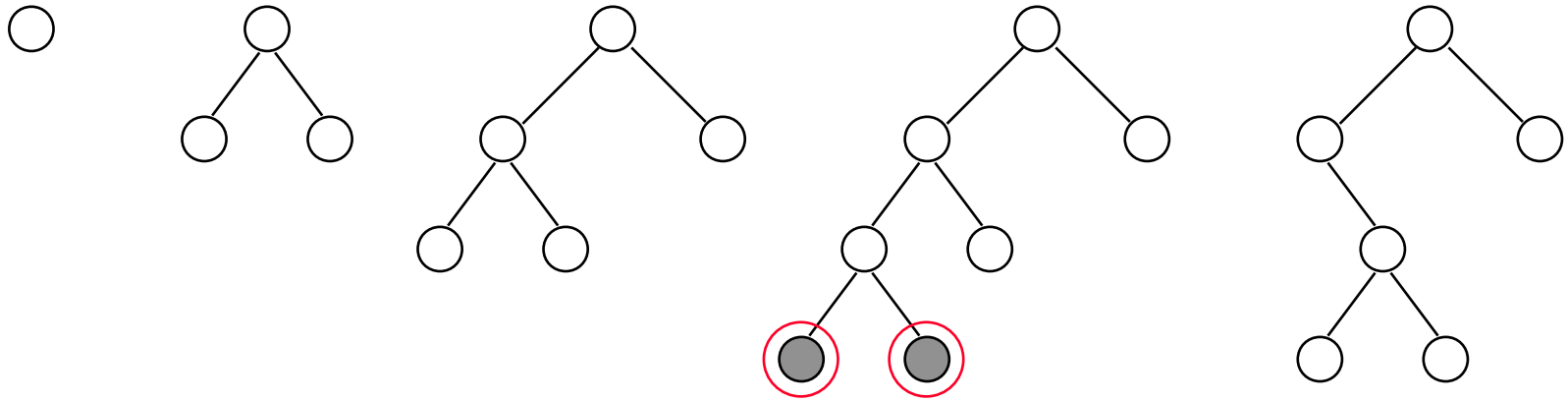




  - Completa
  - Ottimale
  - Time complexity: $O(b^d)$
  - Space complexity: $O(b^d)$
  - Euristica: si trovano prima i cammini meno costosi

Strategie di ricerca (5)

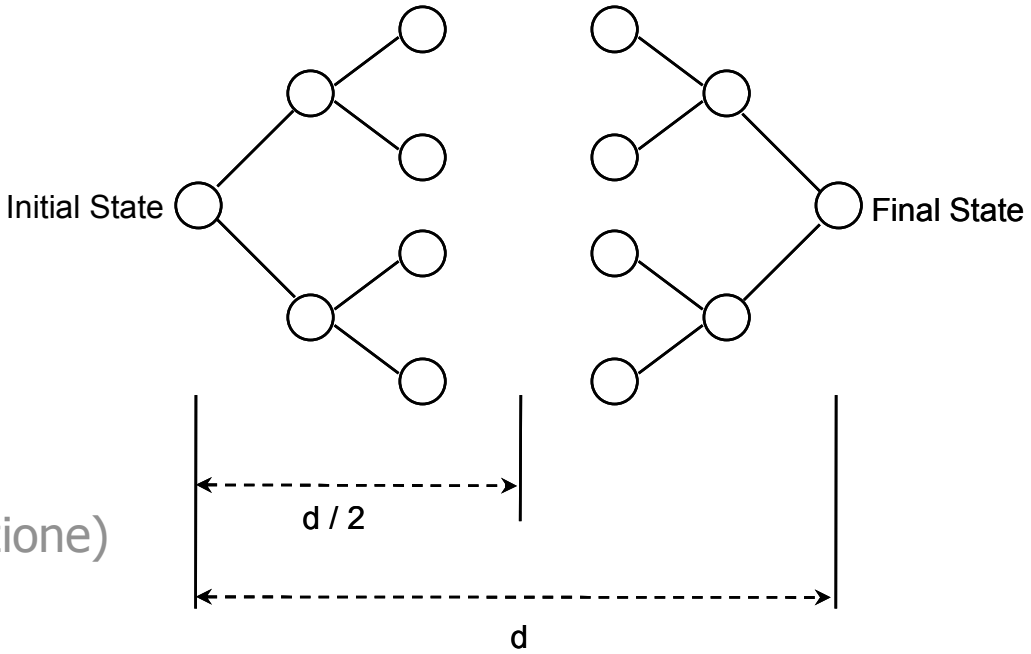
- **Strategia depth-first** (ricerca in profondità)
  - Si espandono i nodi privilegiando la profondità
  - Si ritrattano i rami che non hanno successo



  - Completa (se si cerca una sola soluzione)
  - Non ottimale
  - Time complexity: $O(b^d)$
  - Space complexity: O(bd) (si espande un solo ramo alla volta)
  - Euristica: si trova prima una soluzione (qualsiasi)

Strategie di ricerca (6)

- **Strategia bidirezionale**
  - Si espandono i nodi nelle due direzioni, *top-down* e *bottom-up*



  - Completa
    (se si cerca una sola soluzione)
  - Non ottimale
  - Time complexity: $O(b^{d/2})$
  - Space complexity: $O(b^{d/2})$
  - Euristica: 'soffre meno' il *branching factor* …

Strategie di ricerca (7)

- Aspetti critici
  - Strategia depth-first e cicli nello spazio degli stati
    - La cancellazione dei rami già esplorati potrebbe portare all'esplorazione ripetuta di intere porzioni dello spazio
    - Spunto di riflessione:
      com'è risolto questo problema in Fox, Goat and Cabbage?
      (anche se la strategia non è depth first)