



Università di Parma

Dipartimento di Ingegneria e Architettura

Introduzione all'Intelligenza Artificiale

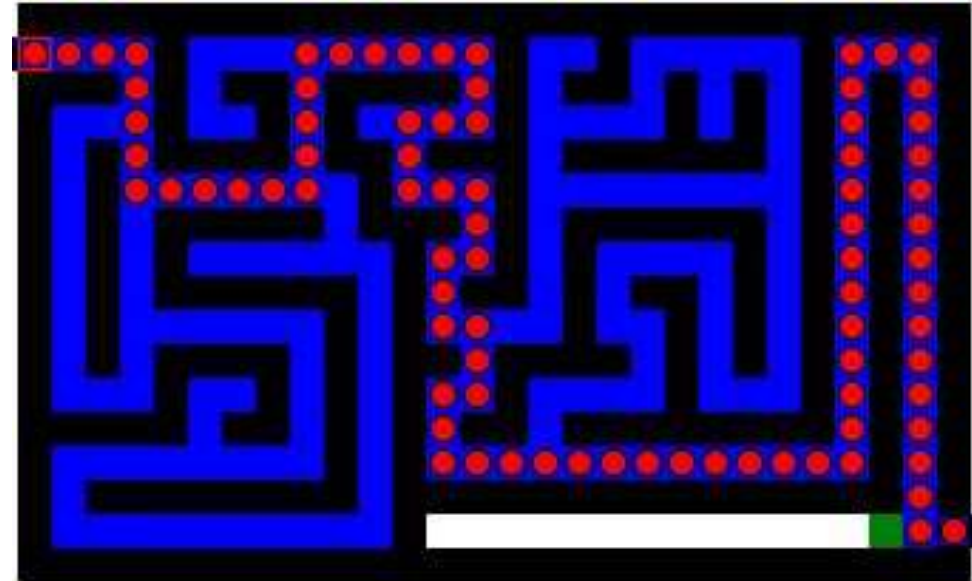
A.A. 2022/2023

Big Data & Business Intelligence

Corso di «Introduzione all'Intelligenza Artificiale» Corso di «Big Data & Business Intelligence»

Agenti Intelligenti

Monica Mordonini (monica.mordonini@unipr.it)



PROBLEM SOLVING AGENTS

RISOLVERE PROBLEMI TRAMITE LA RICERCA

Agenti risolutori di problemi

Gli agenti reattivi sono i più semplici e basano le loro azioni su una mappatura diretta dagli stati alle azioni.

Tali agenti non possono funzionare bene in ambienti per i quali questa mappatura sarebbe troppo grande per essere memorizzata e richiederebbe troppo tempo per essere appresa.

Gli agenti basati sugli obiettivi, d'altra parte, considerano le azioni future e l'opportunità dei loro risultati.

Agenti basati su obiettivi



Conoscere lo stato attuale dell'ambiente non sempre è sufficiente per decidere cosa fare

Bisogna avere un obiettivo

Ricerca di una soluzione al problema

Pianificazione

Agenti risolutori di problemi: ragionare per cercare soluzioni

- ❑ L'agente risolutore di problemi è un tipo particolare di agente basato su obiettivi;
- ❑ Decide cosa fare *ricercando* sequenze di azioni che conducono a stati desiderabili:
 - gli obiettivi aiutano a selezionare le sequenze.
- ❑ Quindi cerca di massimizzare la misura delle prestazioni

Agenti *risolutori di problemi*



- ❑ Adottano il paradigma della **risoluzione di problemi come ricerca** in uno spazio di stati (*problem solving*)
- ❑ Sono agenti con modello che adottano una rappresentazione atomica dello stato
- ❑ Sono particolari agenti con obiettivo, che pianificano l'intera sequenza di mosse prima di agire

Risoluzione di problemi tramite ricerca

- ❑ Gran parte dei problemi di Intelligenza Artificiale hanno la ricerca (o controllo) come componente fondamentale.
- ❑ I problemi si possono modellare come Problemi di **Ricerca** in uno spazio degli stati (**Strategie di Ricerca**).

Risoluzione di problemi tramite ricerca

- ❑ Un problema viene risolto ricercandone la soluzione in un ampio spazio di possibili soluzioni.
- ❑ La ricerca di una soluzione è la traduzione della conoscenza in una rappresentazione opportuna del mondo composta da stati (situazioni del mondo) e un insieme di operatori per passare da uno stato ad un altro.

Risoluzione di problemi tramite ricerca

Ruolo della
Conoscenza



Rappresenta informazioni
su cui poi ragionare per
agire razionalmente

□ *Agire per risolvere problemi*

– Formulazione dell'obiettivo

- Organizza il comportamento, limitando il numero degli scopi da raggiungere
- Si basa sulla situazione corrente e sulle capacità del soggetto

– Formulazione del problema

- Processo che porta a decidere quali azioni da compiere dato un obiettivo
- La selezione della sequenza di azioni che porta all'obiettivo è detta **ricerca**

Agente risolutore di problemi

Ove esistano più alternative (sequenze di azioni che raggiungono l'obiettivo), l'agente

Se non conosce lo stato risultante dopo aver compiuto ciascuna azione, né altre informazioni aggiuntive,
potrà solo scegliere a caso

Se possiede informazioni sugli stati nei quali potrebbe portarsi e sulle azioni che potrebbe compiere,
userà queste informazioni per scegliere la sequenza di azioni da intraprendere.

Tale conoscenza extra al di là dello spazio di ricerca è la cosiddetta **conoscenza euristica** (equivalente a soluzione ad hoc per gli umani)

Agente risolutore di problemi

- Un agente con diverse opzioni immediate di esito sconosciuto può decidere cosa fare esaminando prima le differenti sequenze possibili di azioni che conducono a stati di esito conosciuto scegliendo, poi, quella migliore.
- Il processo di cercare tale sequenza è chiamato **ricerca**.
- L'idea è quella di mantenere ed estendere un insieme di sequenze di soluzioni parziali.
- *È utile pensare al processo di ricerca come la costruzione di un albero di ricerca i cui nodi sono stati e i cui rami sono operatori.*

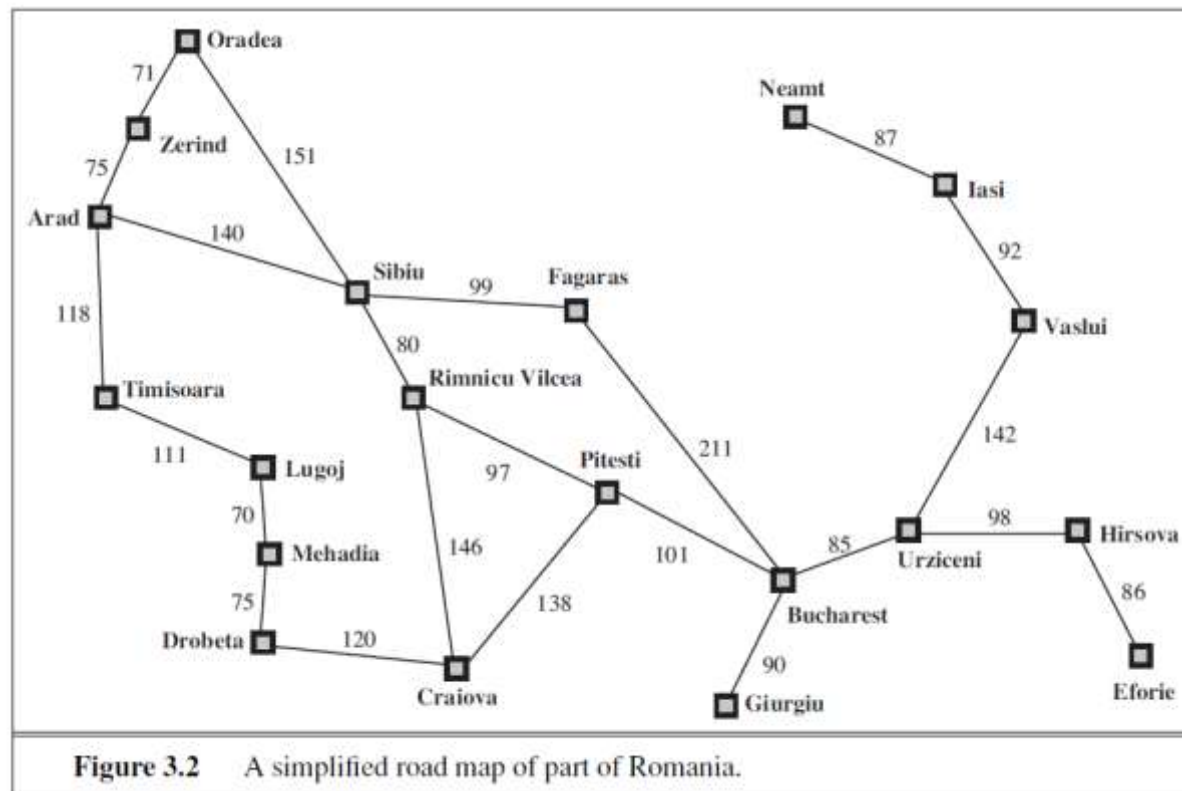
Agente risolutore di problemi

- ❑ **Spazio degli stati**
- ❑ Lo spazio degli stati è l'insieme di tutti gli stati raggiungibili dallo stato iniziale con una qualunque sequenza di operatori.
- ❑ **E' caratterizzato da:**
 - Uno stato iniziale in cui l'agente sa di trovarsi (non noto a priori);
 - Un insieme di azioni possibili che sono disponibili da parte dell'agente (Operatori che trasformano uno stato in un altro o più formalmente una funzione successore $S(X)$ che riceve in ingresso uno stato e restituisce l'insieme degli stati raggiungibili).
 - Un cammino è una sequenza di azioni che conduce da uno stato a un altro.

Agente risolutore di problemi

- Un problema può essere definito formalmente da cinque componenti:
 1. Lo stato iniziale in cui inizia l'agente.
 2. Una descrizione delle possibili azioni disponibili per l'agente. Dato un particolare stato s , è possibile restituire l'insieme di azioni eseguibili in s .
 3. Una descrizione di ciò che fa ogni azione (si dice funzione transizione o successore
 4. Un percorso nello spazio degli stati è una sequenza di stati collegati da una sequenza di azioni
 5. test obiettivo, che determina se un determinato stato è uno stato obiettivo.

Agente risolutore di problemi: un esempio di problema



*Si è a Oradea e si vuole
arrivare a Bucarest per
prendere un aereo
Si ha a disposizioen una
automobile*

Agente risolutore di problemi



- Un algoritmo di ricerca prende un problema come input e restituisce una soluzione sotto forma di una sequenza di azioni
- Un agente razionale semplice opera quindi in tre fasi
 - Formulazione del problema
 - Ricerca della soluzione
 - Esecuzione della sequenza di azioni

Un semplice agente risolutore di problemi

function SIMPLE-PROBLEM-SOLVING-AGENT(*percept*) **returns** an action

persistent: *seq*, an action sequence, initially empty
state, some description of the current world state
goal, a goal, initially null
problem, a problem formulation

state \leftarrow UPDATE-STATE(*state*, *percept*)

if *seq* is empty **then**

goal \leftarrow FORMULATE-GOAL(*state*)

problem \leftarrow FORMULATE-PROBLEM(*state*, *goal*)

seq \leftarrow SEARCH(*problem*)

if *seq* = *failure* **then return** a null action

action \leftarrow FIRST(*seq*)

seq \leftarrow REST(*seq*)

return *action*

NB durante l'esecuzione delle azioni le percezioni vengono ignorate: si ipotizza che la soluzione funzioni sempre

Figure 3.1 A simple problem-solving agent. It first formulates a goal and a problem, searches for a sequence of actions that would solve the problem, and then executes the actions one at a time. When this is complete, it formulates another goal and starts over.

Agente risolutore di problemi

- Poiché durante l'esecuzione delle azioni le percezioni vengono ignorate si ipotizza che l'ambiente sia statico ed osservabile
- L'idea poi di esaminare “sequenze di azioni” porta a pensare che l'ambiente possa essere considerato discreto
- Infine si presuppone che l'ambiente sia deterministico

➤ *E' il tipo di problema più semplice MA non l'unico*

(come vedremo in seguito...)

Problemi ben definiti e soluzioni

- Un problema è definito da un goal e dall'insieme di strumenti a disposizione per raggiungerlo.
- La prima fase della risoluzione di un problema è la *formulazione del problema*.
- In questa fase, oltre a dare una buona definizione del goal, bisogna decidere quali sono le azioni, quale è lo spazio degli stati (insieme degli stati raggiungibili) e quale è lo stato da cui si parte (stato iniziale).

Risoluzione di Problemi



- ❑ A partire dallo stato iniziale bisogna trovare una sequenza di azioni, **soluzione**, che soddisfi l'obiettivo
- ❑ E' essenziale scegliere delle azioni, e quindi degli stati, che permettano di formulare soluzioni al giusto livello di dettaglio (il processo di ***astrazione***) eliminando quelli inutili per ottenere soluzioni «semplici» in modo efficiente
- ❑ Una buona astrazione comporta l'eliminazione di più dettagli possibili mantenendo la validità ed assicurando che le azioni astratte siano facili da realizzare

Agente risolutore di problemi: Formulazione del problema



- ✓ Nella formulazione del problema il processo di **astrazione** esegue il passaggio dal problema concreto al modello del problema con la rimozione dei dettagli «inutili» per la ricerca di una soluzione.
- ✓ Trovare il giusto livello di astrazione è una procedura a volte complessa che richiede, in questo tipo di agenti, una attività umana

Agente risolutore di problemi: Formulazione del problema



- ❑ Esempio:
- ❑ Una formulazione del goal essere in Bucarest è diverso che dire essere in aeroporto a Bucarest
- ❑ Sembra ragionevole, ma è comunque un modello, cioè una descrizione matematica astratta
- ❑ Così come guidare per 99 km non tiene conto delle condizioni stradali, del tipo di paesaggio, del meteo, dei limiti orari, ...

- ❑ La scelta di una buona astrazione implica quindi la rimozione di quanti più dettagli possibile, pur mantenendo la validità e garantendo che le azioni astratte siano facili da eseguire.
- ❑ Se non fosse per la capacità di costruire utili astrazioni, gli agenti intelligenti sarebbero completamente sommersi dal mondo reale.



- ❑ L'esecuzione di ogni azione ha un **costo**, che può variare da azione ad azione, che costituisce il criterio utilizzato nella ricerca della soluzione migliore.
- ❑ Quindi, ad ogni sequenza di azioni è possibile assegnare un **costo complessivo** (somma dei costi associati alle varie azioni), per valutare quale sia la **soluzione migliore (soluzione ottima)**.

Agente risolutore di problemi

- ❑ La scelta della sequenza corrisponde alla seconda fase della risoluzione di un problema e viene individuata con il nome di **ricerca**.
- ❑ Infine, trovata la soluzione, nella terza fase, detta di **esecuzione**, le azioni sono eseguite per raggiungere il goal.

Agente risolutore di problemi



❑ *Misurare l'efficacia della ricerca:*

- Si riesce a trovare una soluzione?
- È una buona soluzione? (con basso costo di cammino - costo in linea)
- Qual è il costo della ricerca? (tempo per trovare una soluzione - costo fuori linea)

Costo totale della soluzione = costo di cammino + costo di ricerca.

Ancora una volta...

Scegliere stati e azioni -> L'importanza dell'astrazione

Agente risolutore di problemi: costo della ricerca

- ❑ L'agente deve decidere quali risorse dedicare alla ricerca e quali all'esecuzione.
- ❑ Per spazi degli stati piccoli, si considera il costo di cammino più basso
- ❑ Per problemi complessi trovare il punto di equilibrio (l'agente può cercare per un tempo molto lungo di ottenere una soluzione ottimale, oppure può cercare per un tempo più breve ed ottenere una soluzione con costo di cammino lievemente maggiore)

Agente risolutore di problemi e classi di problemi



- ❑ L'agente risolutore di problemi può essere applicato a un insieme vasto di problemi da giocattolo (toy problem) che problemi reali.
- ❑ Un problema giocattolo ha lo scopo di illustrare o esercitare vari metodi di risoluzione dei problemi. Può essere fornita una descrizione concisa ed esatta e quindi è utilizzabile da diversi ricercatori per confrontare le prestazioni degli algoritmi.
- ❑ Un problema del mondo reale è quello le cui soluzioni interessano davvero alle persone. Tali problemi tendono a non avere un'unica descrizione concordata, ma possiamo solo dare una indicazione generale delle loro formulazioni.

Ma ci sono tipi di problemi che non possono essere risolti tramite questo tipo di agente

Esempi di problemi

Un esempio: **vacanza in Romania**.

Attualmente in **Arad**. L'aereo parte domani da **Bucarest**.

Formulare un goal:

esssere in Bucarest

Formulare un problema:

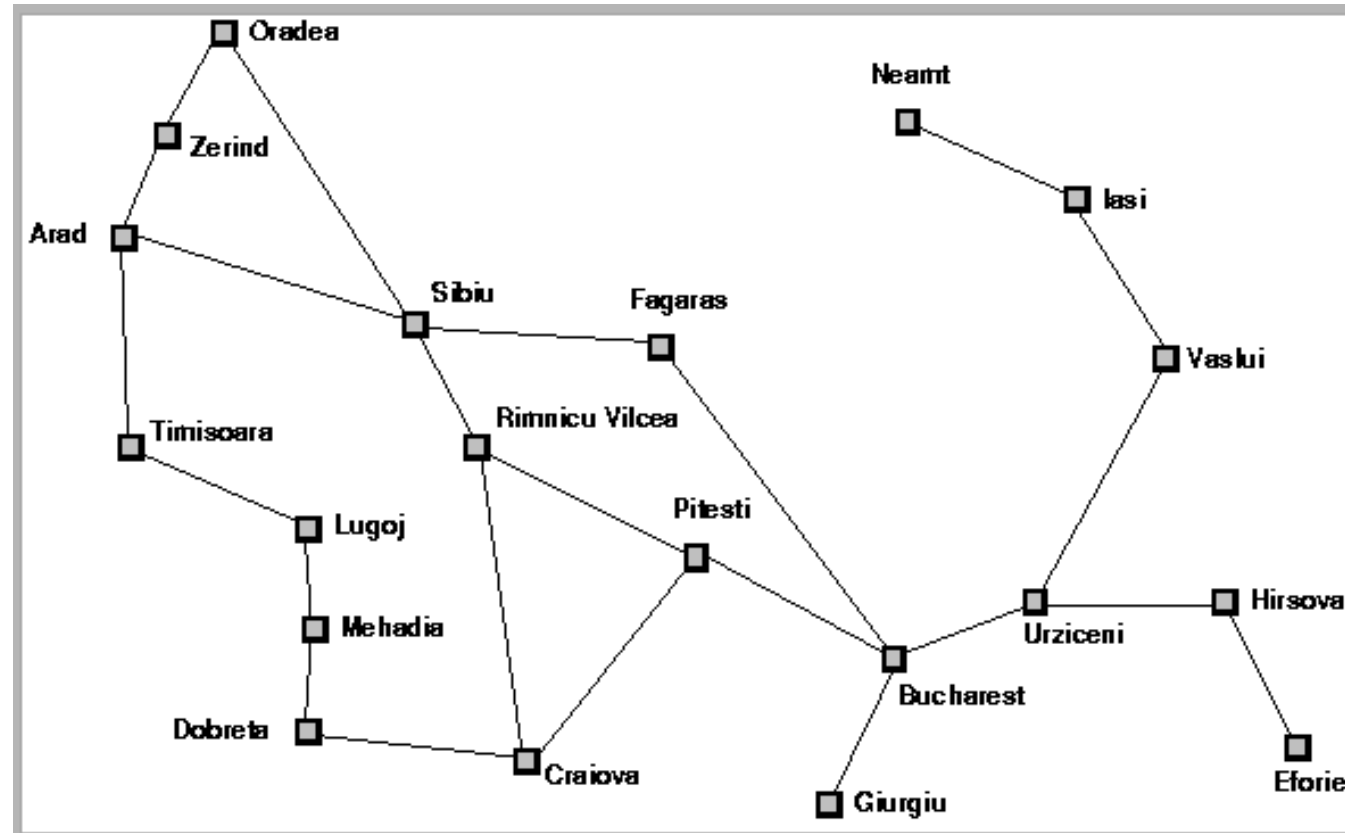
stati: varie città

operatori: guidare da una città all'altra

Trovare una soluzione:

sequenza di città, es: Arad, Sibiu, Faragas, Bucarest

Esempi di problemi



Esempi di problemi



Formulazione di un **problema** a singolo stato:

stato iniziale: essere “in Arad”

operatori: Arad -> Zerind, Arad -> Sibiu etc.

La funzione successore S fa passare dallo stato x agli stati $S(x)$.

L'insieme degli stati raggiungibili definisce lo **spazio degli stati**.

test obiettivo:

esplicito: “in Bucarest”

implicito: “NonSporco(x)”

costo del cammino: es. Somma delle distanze, numero di operatori applicati, etc.

soluzione: una sequenza di operatori che porta da uno stato iniziale a uno stato obiettivo.

□ Problemi giocattolo

- (modelli semplificati di classi di problemi reali):
 - ✓ Il problema dell'8-puzzle (versione ridotta del gioco del 15)
 - ✓ Il problema delle 8 regine (disporre 8 regine su una scacchiera in modo che nessuna dia scacco alle altre)
 - ✓ Le parole crociate (spazio di ricerca amplissimo)
 - ✓ Il problema dei tre missionari e dei tre cannibali (trasportare 3 m. e 3 c. da un lato a un altro di un fiume con una barca a due posti in modo tale che sulle 2 rive non ci siano mai c. in sovrannumero rispetto ai m.)

Esempi di Problemi



5	4	
6	1	8
7	3	2
Start State		

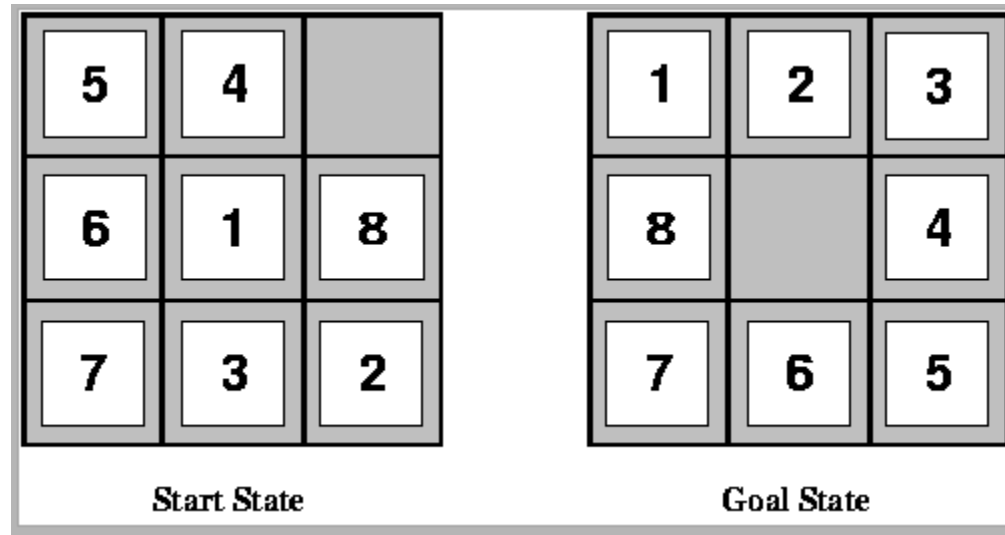
1	2	3
8		4
7	6	5
Goal State		

Una tipica istanza del rompicapo dell'8

Si hanno otto tessere scorrevoli numerate in una griglia 3×3 . Le tessere possono scorrere nelle quattro direzioni (su, giù, destra e sinistra) ma solo se vanno ad occupare la posizione lasciata vuota. Lo scopo è quello di ordinare le tessere, lasciando lo spazio vuoto in fondo.

Pur esistendo $9! = 362880$ possibili disposizioni dei numeri nelle caselle, gli stati raggiungibili usando gli operatori consentiti sono 181433, ovvero circa la metà, e solo uno di questi è quello vincente.

Esempi di Problemi



Una tipica istanza del rompicapo dell'8

Stati: uno stato specifica la posizione di ciascuna delle 8 tessere.

Operatori: lo spazio vuoto si muove a destra, a sinistra, sopra, sotto.

Test obiettivo: lo stato rispecchia la configurazione obiettivo (Goal).

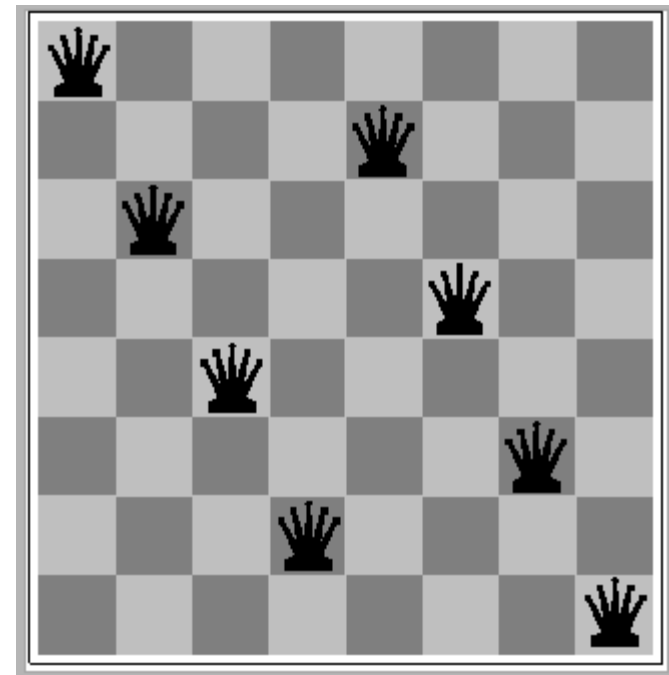
Costo del cammino: ciascun passo costa 1. Il costo del cammino coincide con la sua lunghezza.

Esempi di problemi



Il problema delle 8 regine

Si dispongono 8 regine su una scacchiera 8×8 facendo in modo che queste non si mangino l'una con l'altra e tenendo presente che le regine dominano in orizzontale, verticale e lungo le diagonali.



Esempi di problemi



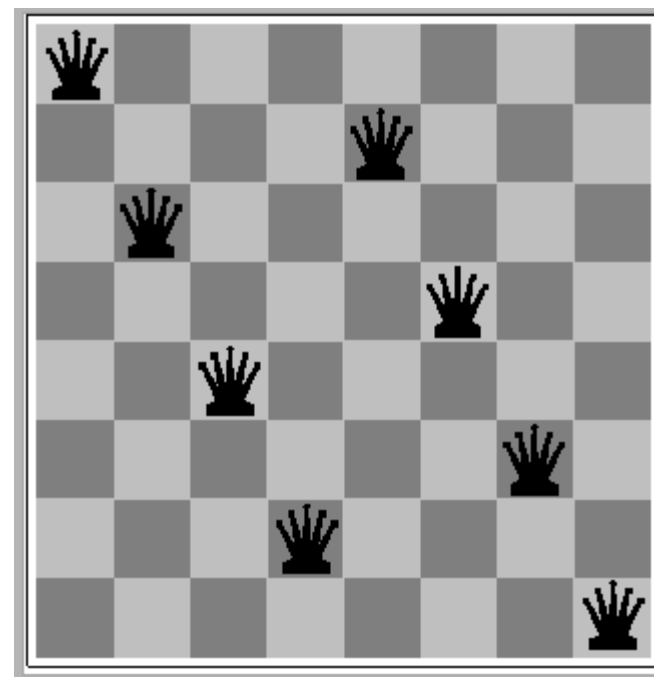
Il problema delle 8 regine

Sebbene esistano algoritmi efficienti specializzati per questo tipo di problema rimane un utile problema di test per algoritmi di ricerca.

Ci sono due principali tipi di formulazione.

Una formulazione incrementale coinvolge operatori che aumentano la descrizione dello stato, che inizia con uno stato vuoto e che per il problema delle 8 regine, questo significa che ogni azione aggiunge una regina allo stato. La seconda formulazione è a stato completo che inizia con tutte e 8 le regine sul tabellone e le sposta attorno.

In entrambi i casi, il costo del percorso non interessa perché conta solo lo stato finale.



Esempi di problemi



Il problema delle 8 regine

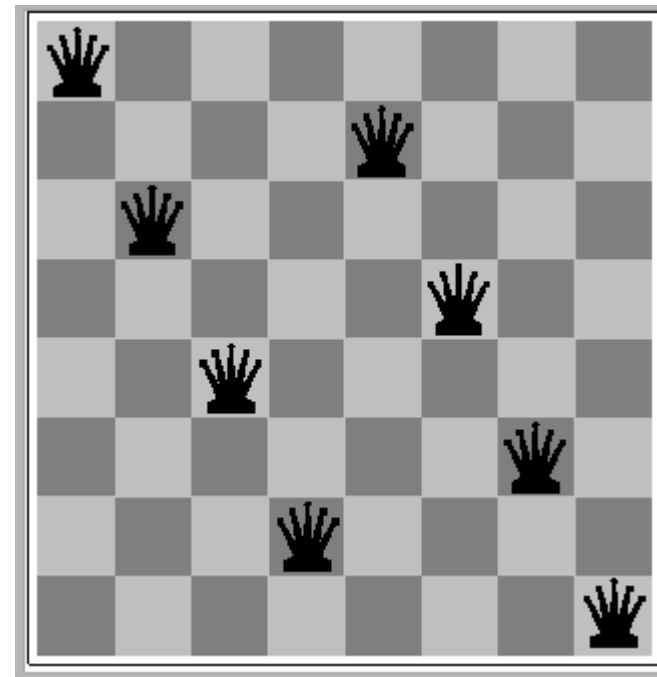
Nella formulazione incrementale

Stati: qualsiasi configurazione da 0 a 8 regine sulla scacchiera.

Operatori: aggiungi una regina in qualsiasi quadrato

Test obiettivo: 8 regine sulla scacchiera, nessuna minacciata.

Costo cammino: 0 (perché conta solo lo stato finale).



$64 \times 63 \times \dots \times 57$ $1,8 \times 10^{14}$
sequenze da considerare

Esempi di problemi



Il problema dei Missionari e dei Cannibali:

3 missionari e 3 cannibali devono attraversare un fiume. C'è una sola barca che può contenere al massimo due persone. Per evitare di essere mangiati i missionari non devono mai essere meno dei cannibali sulla stessa sponda (stati di fallimento).

- ❑ Stato: sequenza ordinata di tre numeri che rappresentano il numero di missionari, cannibali e barche sulla sponda del fiume da cui sono partiti.
- ❑ Stato iniziale è: $(3,3,1)$ (*importanza dell'astrazione*).
- ❑ Operatori: gli operatori devono portare in barca 1 missionario, 1 cannibale, 2 missionari, 2 cannibali, 1 missionario e 1 cannibale.
Al più 5 operatori (grazie all'astrazione sullo stato scelta).
- ❑ Test Obiettivo: $(0,0,0)$
- ❑ Costo di cammino: numero di traversate.

Esempi di Problemi

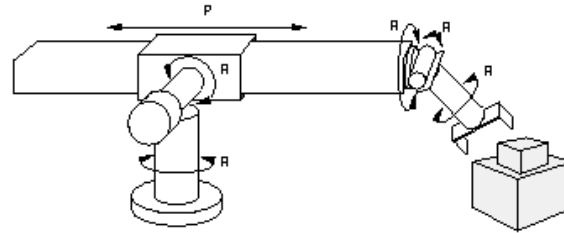
□ Problemi reali

- ✓ Il calcolo di un percorso su un grafo
- ✓ Il problema del commesso viaggiatore
(date n città, trovare il percorso più breve che passa una e una sola volta per ciascuna di esse)
- ✓ Il disegno di un layout VLSI
- ✓ La navigazione di robot
- ✓ Il calcolo di una sequenza di assemblaggio

Esempi di problemi



Example: robotic assembly



states??: real-valued coordinates of
robot joint angles
parts of the object to be assembled

operators??: continuous motions of robot joints

goal test??: complete assembly *with no robot included!*

path cost??: time to execute

Classi di Problemi

- ✓ L'agente risolutore di problemi riesce a risolvere problemi in mondo osservabile, statico e deterministico.
 - ✓ Non è l'unico tipo di problema
- ✓ Vediamo allora se è possibile raggruppare di problemi secondo alcune caratteristiche generali
- I problemi possono essere divisi in quattro classi in dipendenza della conoscenza che l'agente ha sullo stato del mondo e sulle azioni che si possono eseguire su di esso
 - La conoscenza che l'agente ha sulle sue azioni e sugli stati del mondo dipende da come è connesso al suo ambiente, tramite “percezioni e azioni”

Classi di Problemi



1. ***Problemi a stato singolo*** (mondo accessibile)
2. ***Problemi a stati multipli*** (mondo non accessibile - no sensori)
3. ***Problemi di contingenza*** (ignoranza dell'agente sulla sequenza di azioni)
4. ***Problemi di esplorazione*** (nessuna conoscenza sugli effetti delle proprie azioni)

□ *Problemi a stato singolo (single-state problems)*

si ha conoscenza

- dello stato del mondo (tramite i sensori l'agente riceve informazioni sufficienti sullo stato in cui si trova)
- e degli effetti delle proprie azioni

problema deterministico e osservabile:

l'agente può calcolare esattamente in quale stato sarà dopo qualsiasi sequenza di azioni

Problemi a stato singolo *Formulazione di un problema*

Un tale problema è definito da :

1. *Stato iniziale x*
2. *Operatore / funzione successore $S(x)$*
3. *Test obiettivo*
4. *Funzione costo cammino*

Una soluzione è una sequenza di operatori che conducono dallo stato iniziale ad uno stato obiettivo

□ *Problemi a stato multiplo (multiple-state or conformant problems)*

si ha conoscenza completa degli effetti delle azioni come nel problema a stato singolo

ma non si ha conoscenza sufficiente sullo stato del mondo (problema non osservabile, p.e. agente senza sensori)

A partire dallo stato iniziale l'agente deve ragionare su insiemi di stati in cui potrebbe giungere invece che su stati singoli.

Problemi a stato multiplo :problemi conformanti

- ***Infatti***
- Se l'agente è privo di sensori per quanto ne sa potrebbe trovarsi in uno tra più stati iniziali e ogni azione potrebbe portarlo a uno tra molti possibili stati successivi
- L'agente deve quindi parlare di insiemi di stati raggiungibili e non di singoli stati
- Ogni insieme viene chiamato stato-credenza (belief state)

Problemi a stato multiplo: *Formulazione di un problema*

Un tale problema è definito da :

1. *Insieme di stati iniziali*
2. *Insieme di operatori / funzione successore $S(x)$* (per ciascuna azione viene specificato l'insieme di stati raggiunti da qualsiasi stato considerato. Un cammino collega insiemi di stati)
3. *Test obiettivo*
4. *Funzione costo cammino*

Una soluzione è un cammino che conduce ad un insieme di stati (sottoinsieme dello spazio degli stati) che sono tutti stati obiettivo.

Classi di Problemi



- *Problemi con imprevisti (contingency problems)*
 - ✓ alcune azioni possono avere effetti non del tutto determinati (incertezza, malfunzionamenti ecc.)
 - ✓ Sono problemi non deterministici e, parzialmente osservabili
 - ✓ E' presente una capacità di rilevamento durante la fase di esecuzione

Classi di Problemi



□ *Problemi con imprevisti (contingency problems):*

In questi casi l'agente deve

- calcolare un intero albero di azioni piuttosto che una singola sequenza di azioni
 - *un ramo dell'albero tratta una situazione contingente possibile che si potrebbe verificare*

Spesso per la soluzione è necessario agire (*e verificare l'effetto della propria azione*) prima di avere trovato una soluzione

- Tipico dei casi reali in cui una predizione esatta è impossibile

Problemi con imprevisti : *Problemi di contingenza*

□ **Ora:**

- Se l'ambiente è parzialmente osservabile, o le azioni sono incerte, le percezioni dell'agente forniscono effettivamente nuove informazioni dopo ogni azione.
- Ogni possibile azione definisce una contingenza che deve essere pianificata
- L'agente neccessiterà quindi di piani di contigenza per gestire le circostanze sconosciute che si potranno verificare
- Un problema è detto con avversari se l'incertezza è causata da un altro agente

Problemi con imprevisti: *Formulazione di un problema*

- ✓ *Necessari algoritmi complessi*
- Spesso per la soluzione è necessario agire (e verificare l'effetto della propria azione) prima di avere trovato una soluzione (*interleave di ricerca e esecuzione*):
 - L'agente incomincia cioè ad eseguire un piano possibile
 - Sulla base di quali soluzioni contingenti si verificano ricalcola ed attua un nuovo piano.
- Si sfruttano le informazioni supplementari trovate runtime per continuare a risolvere il problema

- ❑ ***Problemi di esplorazione (exploration problems):***
 - ✓ non si ha conoscenza né sullo stato del mondo né sugli effetti delle proprie azioni.
 - ✓ Lo spazio degli stati è sconosciuto.
 - ✓ Quando gli stati e le azioni dell'ambiente sono sconosciuti, l'agente deve fare in modo di scoprirle
- ✓ I problemi di esplorazione possono essere considerati casi estremi di problemi di contingenza

Problemi di esplorazione

- ✓ E' necessario *apprendere* i possibili stati del mondo e gli effetti delle proprie azioni attraverso l'*esperienza*.
- ✓ La ricerca si svolge nel mondo reale e non in un modello: agire può comportare danni significativi per un agente privo di conoscenza
- ✓ Se sopravvive, acquisisce conoscenza che può riusare per problemi successivi

Classi di Problemi : *riassunto*

❑ Classi di problemi:

- ❑ Problemi a stato singolo
 - ❑ Problemi a stato multiplo
- SEARCH*
-
- ❑ Problemi di contingenza
- PLANNING*
-
- ❑ Problemi di esplorazione
- LEARNING*

Classi di problemi con un *toy problem*

Il mondo dell'aspirapolvere: problema a stato singolo

Due soli stati : A, B

Percezioni:

In quale stato si trova,
se c'è lo sporco

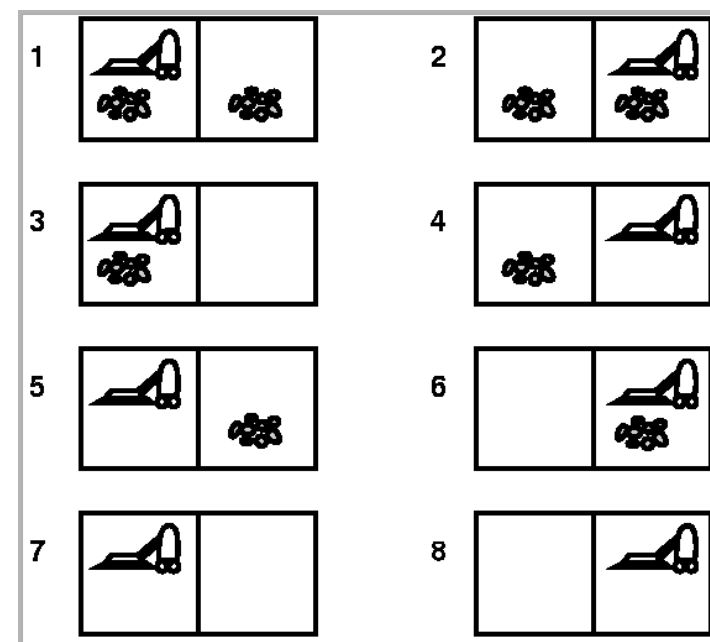
Azioni:

Aspirare lo sporco

Muoversi a destra o a sinistra

Test obiettivo

Non c'è sporco nè a destra nè a sinistra



Azioni: destra, sinistra, aspira

Classi di problemi con un *toy problem*

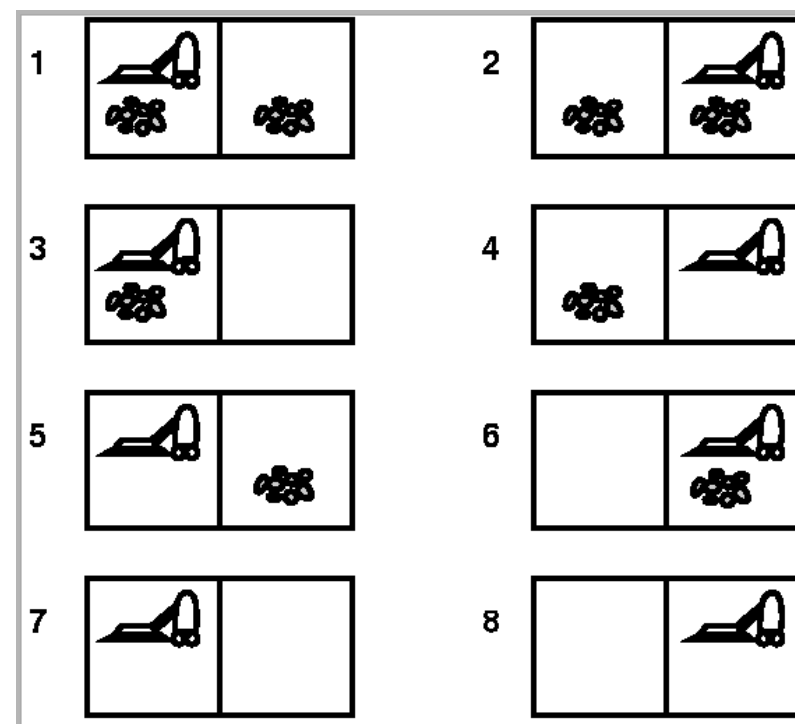
Il mondo dell'aspirapolvere: problema a stato singolo

Funzione successore

Genera gli stati legali che risulta dal tentativo di intraprendere le azioni {1,2,3,4,5,6,7,8,}

Spazio degli stati

L'agente si trova in uno dei due riquadri possibili, ognuno dei quali può contenere sporco oppure no ($2 \times 2^2 = 8$)



Azioni: destra, sinistra, aspira

Classi di problemi con un *toy problem*



Mondo dell'aspirapolvere (singolo stato).

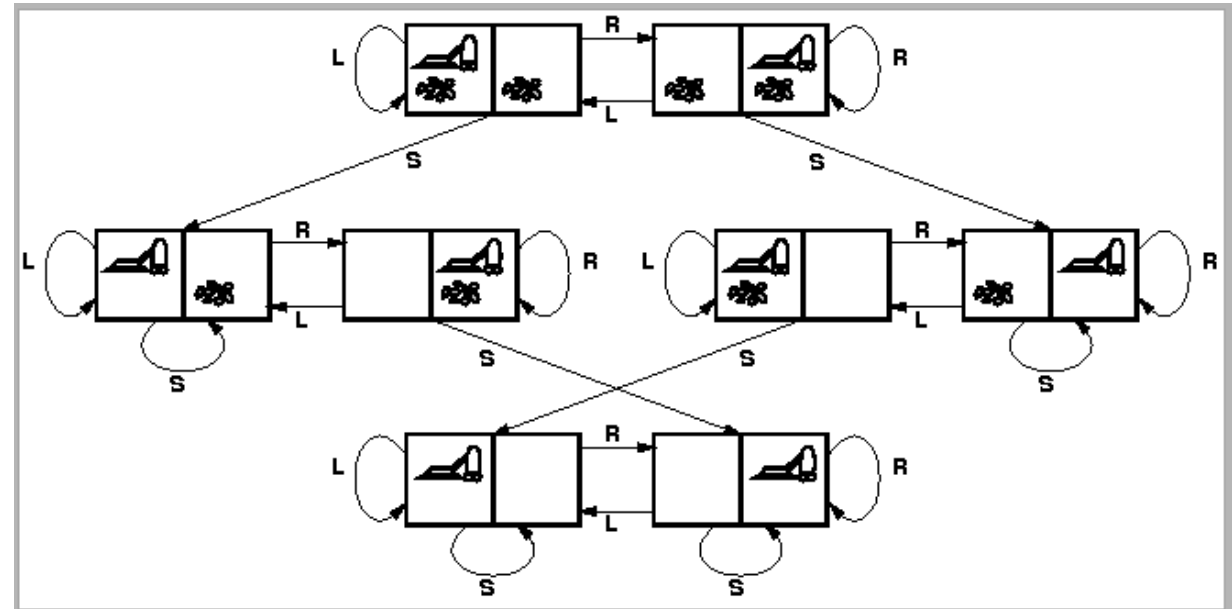
Stati: uno degli 8 stati della figura.

Operatori: spostati a destra, spostati a sinistra, aspira.

Test obiettivo: non lasciare alcuna sporcizia nei quadrati.

Costo del cammino: ciascuna azione costa 1.

Risolvere il problema da uno stato di partenza comporta seguire le frecce nel diagramma degli stati fino a uno stato obiettivo.



Classi di problemi con un *toy problem*

Problemi stati singoli vs. stati multipli

□ Problema a stati singoli:

- Lo stato è sempre accessibile.
- L'agente conosce esattamente che cosa produce ciascuna delle sue azioni e può calcolare esattamente in quale stato sarà dopo qualunque sequenza di azioni.

□ Problema a stati multipli:

- Lo stato non è completamente accessibile. L'agente deve ragionare su possibili stati che potrebbe raggiungere.
- In più: anche l'effetto delle azioni può essere sconosciuto o imprevisto.
- Agente deve ragionare su insiemi di stati piuttosto che su stati singoli

Classi di problemi con un *toy problem*



Mondo dell'aspirapolvere senza sensori: problema a stato multiplo

In ogni istante l'aspirapolvere si trova in uno stato di un insieme ma non sa quale stato dell'insieme è.

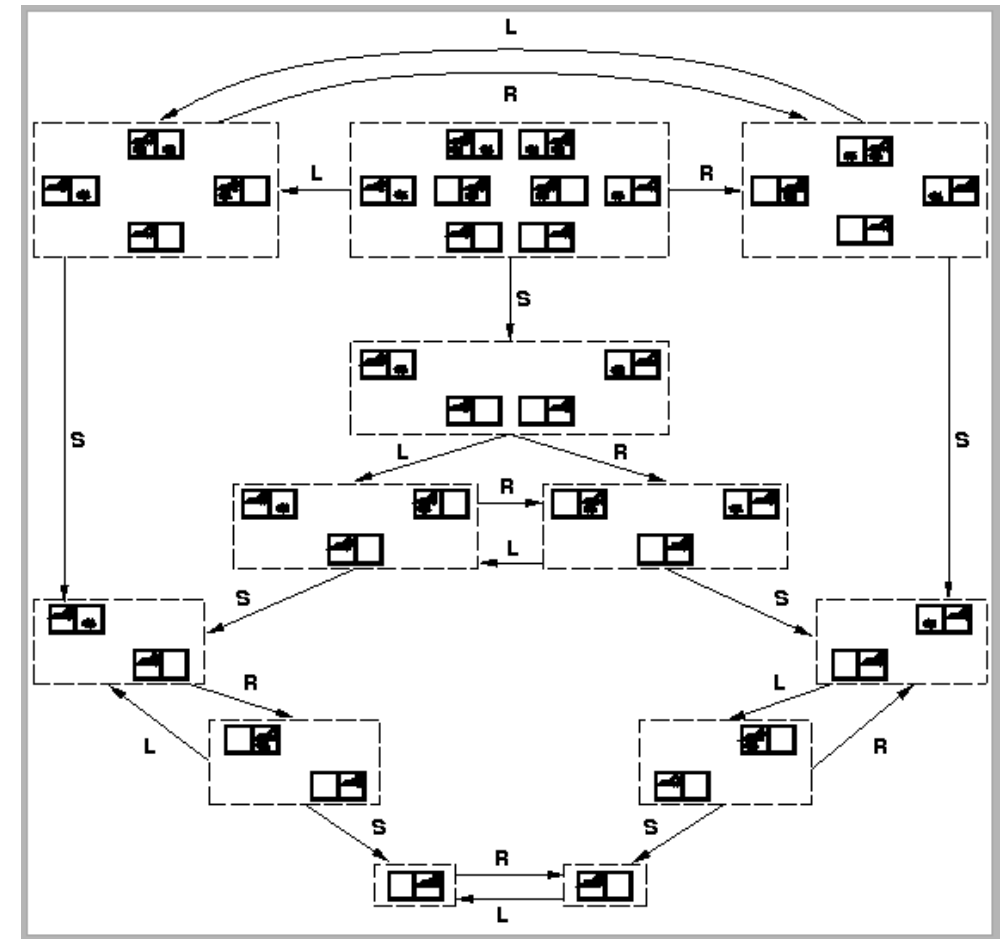
Stati: sottoinsiemi degli stati 1-8 della figura.

Operatori: spostati a destra, spostati a sinistra, aspira.

Test obiettivo: tutti gli stati dell'insieme degli stati non contengono sporcizia.

Costo del cammino: ciascuna azione costa 1.

Una soluzione del problema è una qualsiasi sequenza che porti dall'insieme iniziale degli stati ad un insieme di stati senza sporcizia.





□ Problemi di contingenza

- conoscenza incompleta degli **effetti delle azioni**:
 - le conseguenze dell'azione non sono predicibili e quindi vanno considerati tutti possibili effetti: alcune azioni possono avere malfunzionamenti
 - **SOLUZIONE : alberi di stati**



□ Problemi di contingenza

- conoscenza incompleta degli **effetti delle azioni**:
 - le conseguenze dell'azione non sono predicibili e quindi vanno considerati tutti possibili effetti: alcune azioni possono avere malfunzionamenti
 - **SOLUZIONE : alberi di stati**

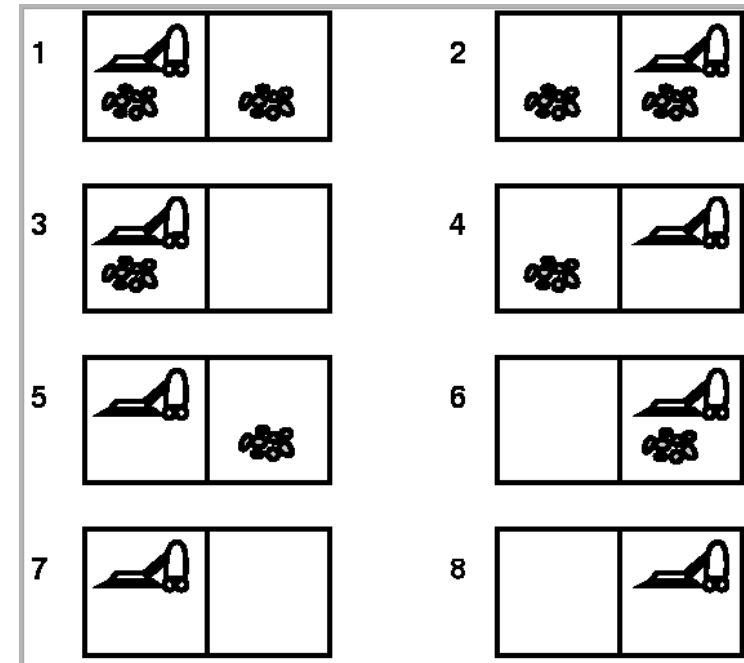
Classi di problemi con un *toy problem*

Mondo dell'aspirapolvere con legge di Murphy: non deterministico (problema di contigenza)

“Aspirare” talvolta deposita sporco, ma solo se in quel punto non c’è già sporco (legge di Murphy).

Se l’aspirapolvere si trova in 4 aspirare potrebbe rimanere nello stato 4 oppure andare a finire nello stato 2

Se ho capacità di rilevamento durante la fase di esecuzione ecco che posso scegliere l’azione di aspirare solo nel caso ci sia sporco



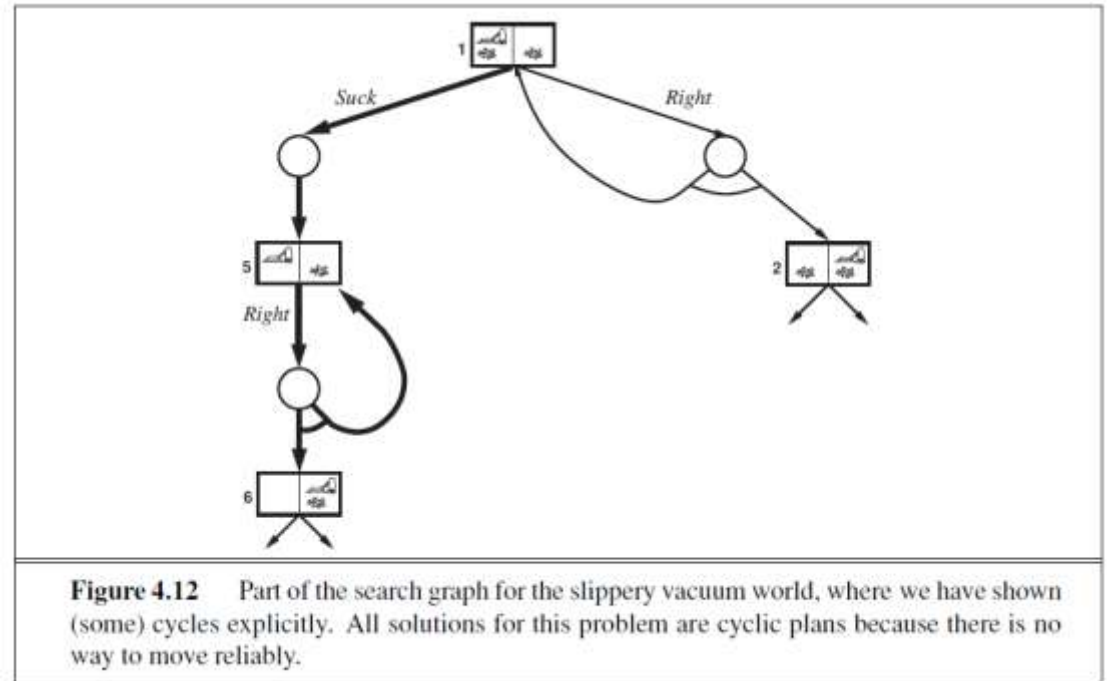
Classi di problemi con un *toy problem*

Mondo dell'aspirapolvere con legge di Murphy: non deterministico (problema di contigenza)

“Aspirare” talvolta deposita sporco, ma solo se in quel punto non c’è già sporco (legge di Murphy).

Se l’aspirapolvere si trova in 4 aspirare potrebbe rimanere nello stato 4 oppure andare a finire nello stato 2

Se ho capacita’ di rilevamento durante la fase di esecuzione ecco che posso scegliere l’azione di aspirare solo nel caso ci sia sporco





□ Problemi di esplorazione

- nessuna informazione degli **effetti delle azioni**
- nessuna informazione **dello stato del mondo**
- l'agente deve compiere azioni esplorative, provare ad agire

Nel mondo dell'aspirapolvere devo dare all'agente la capacità di poter apprendere l'abilità di riconoscere un ambiente sporco (o pulito)