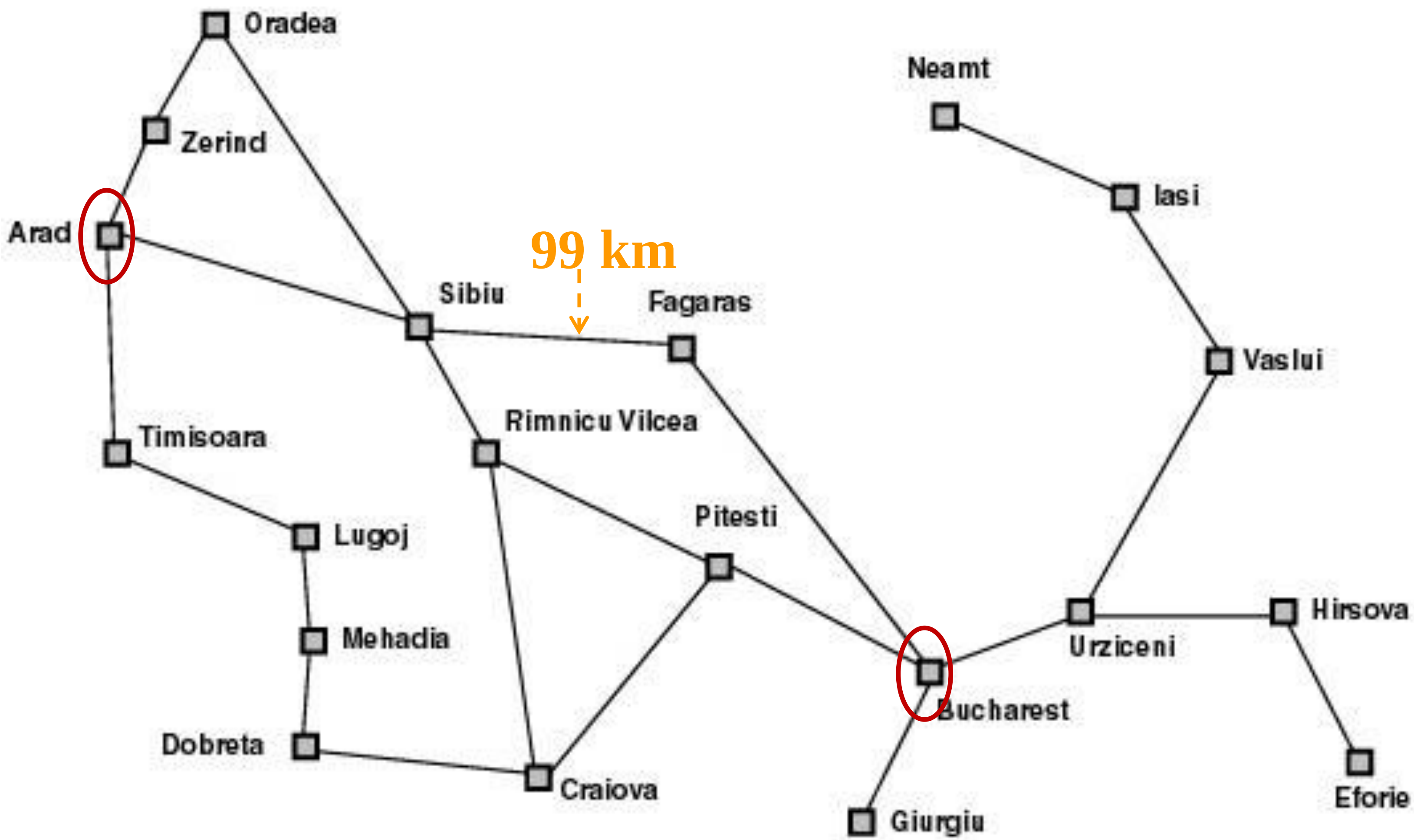
Itinerario: il problema

Caso che vedremo:
trovare il percorso più breve (in km) da una città di partenza a una città di arrivo



Itinerario: la formulazione (la scelta del livello di astrazione)

- *Stati*: <u>le città</u>. Es. *In*(Pitesti)

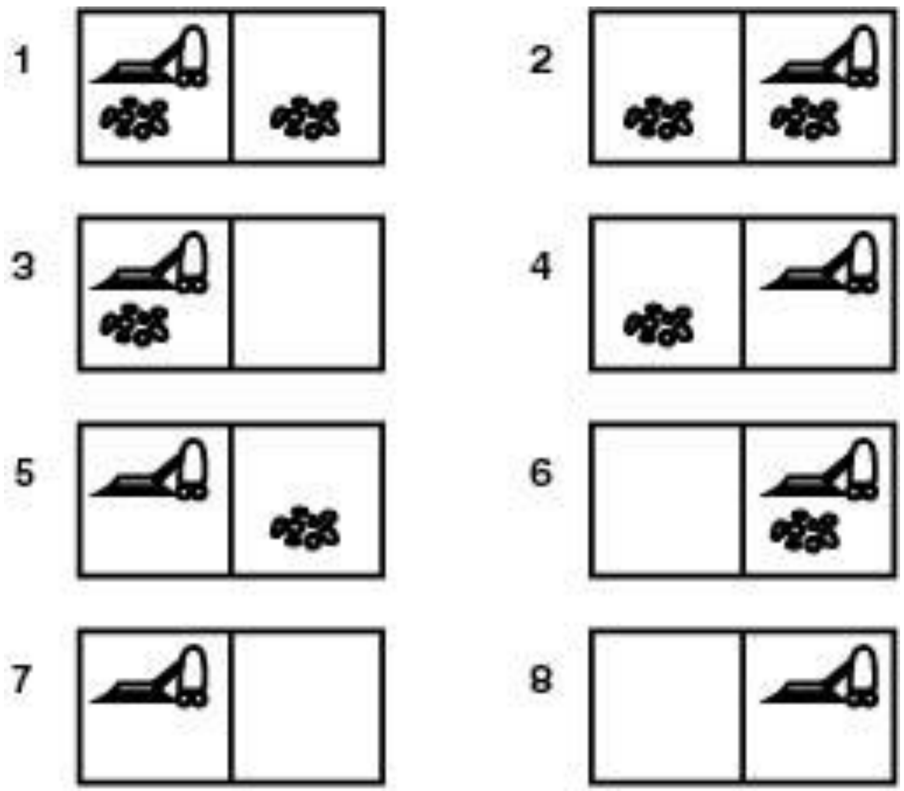
1. *Stato iniziale*: la città da cui si parte. *In*(Arad)
2. *Azioni*: spostarsi su una città vicina collegata
   - Azioni(*In*(Arad)) ={*Go*(Sibiu), *Go*(Zerind) …}
3. Modello di transizione
   - Risultato(*In*(Arad), *Go*(Sibiu)) = *In*(Sibiu)
4. *Test Obiettivo: {In(Bucarest)}*
5. *Costo del cammino*: somma delle lunghezze delle strade

- Lo **<u>spazio degli stati</u>** coincide con la rete (*grafo*) di collegamenti tra città i.e. grafo di stati collegati da azioni, rappresentabile in modo esplicito *in questo caso semplice*, tramite la mappa
- Astrazione dai dettagli: essenziale per “modellare”

Aspirapolvere: il problema (toy problem)

Versione semplice: solo due locazioni, sporche o pulite, l'agente può essere in una delle due

STATI



Percezioni:

Sporco

NonSporco

Azioni:

Sinistra (L)

Destra (R)

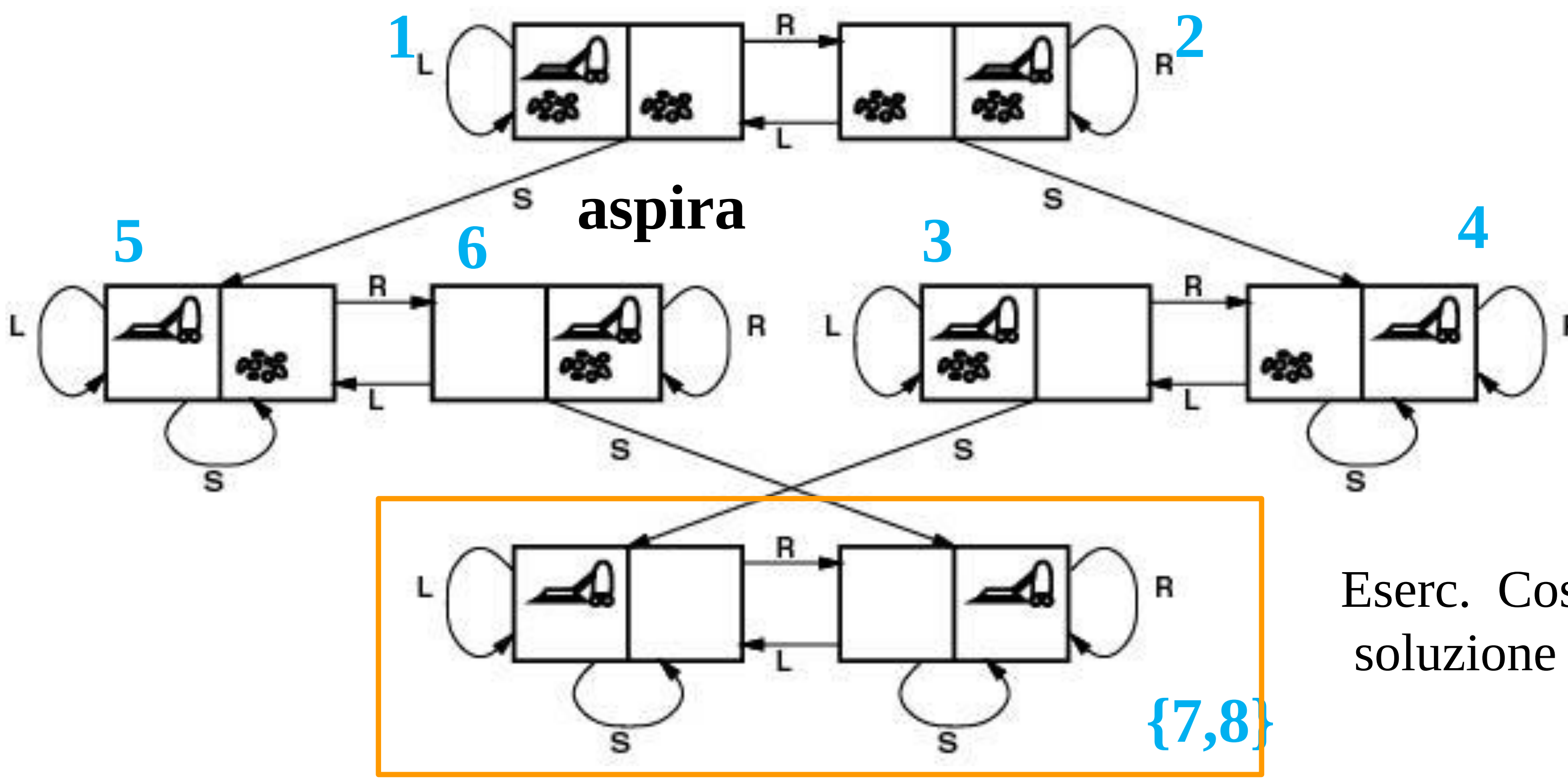
Aspira (S)

Aspirapolvere: formulazione

- *Obiettivo*: rimuovere lo sporco { 7, 8 }
- Ogni azione ha costo 1

SPAZIO DEGLI STATI :

Grafo



Eserc. Costo di
soluzione min.?

Il puzzle dell'otto (o "rompicapo" a 8 tasselli)

5	4	
6	1	8
7	3	2

Start State

1	2	3
8		4
7	6	5

Goal State

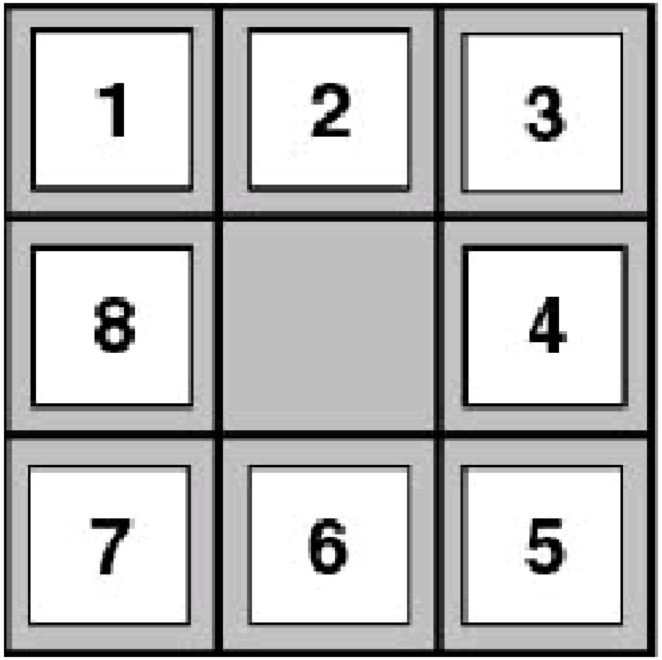
Puzzle dell'otto: formulazione

- *Stati*: possibili configurazioni della scacchiera
- *Stato iniziale*: una configurazione
- *Obiettivo*: una configurazione --->

  Goal-Test: Stato obiettivo? --->
- *Azioni*: mosse della *casella bianca*

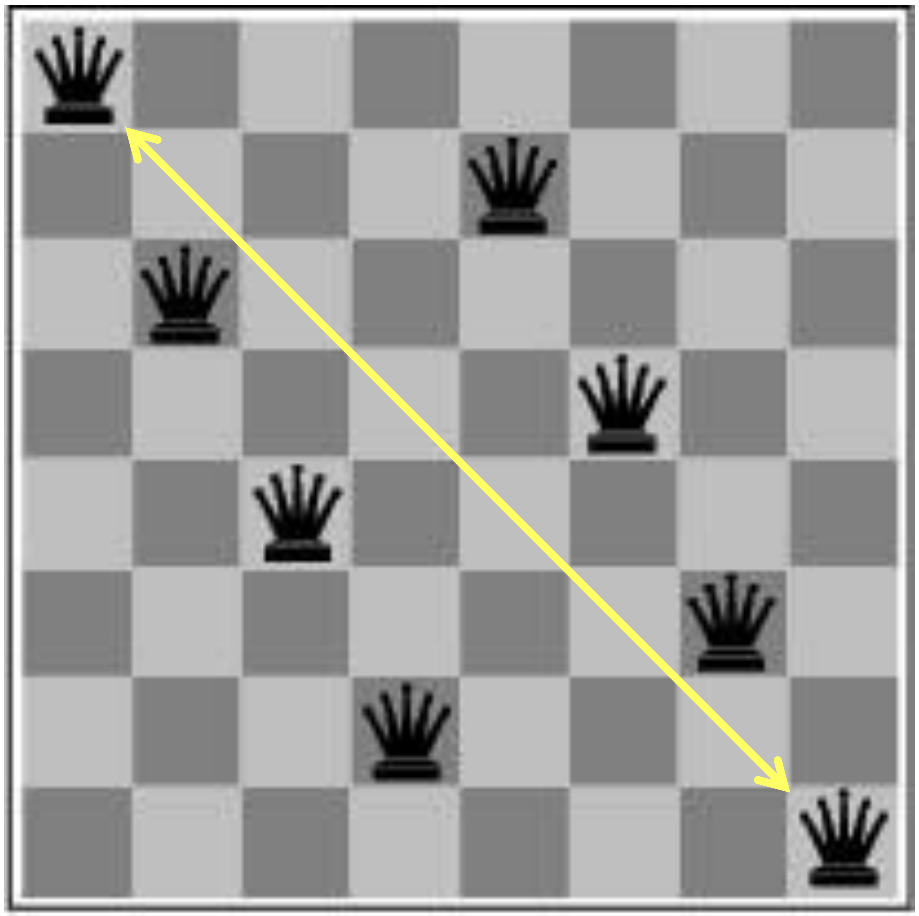
  in sù: ↑ in giù: ↓

  a destra: → a sinistra: ←
- *Costo cammino*: ogni passo costa 1



- Lo *spazio degli stati* è un grafo con possibili cicli.
- NP-completo. Per 8 tasselli: 9!/2 =181K stati (*)! Ma risolvibile in poco tempo (ms). Se cresce no! (*)= http://www.cut-the-knot.com/pythagoras/fifteen.shtml

Le otto regine: il problema



Collocare 8 regine sulla scacchiera in modo tale che nessuna regina sia attaccata da altre: **Questa è una soluzione?**

Le otto regine:

Formulazione incrementale 1

Si aggiungono le regine una alla volta

- *Stati*: scacchiere con 0-8 regine
- *Goal-Test*: 8 regine sulla scacchiera, nessuna attaccata
- *Costo cammino*: zero (resta 8, per le 8 mosse effettive, e non è rilevante, interessa solo lo stato finale)
- *Azioni*: aggiungi una regina
- *Spazio stati*: $64 \times 63 \times \ldots \times 57 \sim 1.8 \times 10^{14}$ sequenze possibili da considerare! (quanti miliardi?)

I.e. la ricerca può essere molto onerosa!

Le otto regine:
Formulazione incrementale 2

- *Stati*: scacchiere con 0-8 regine, nessuna minacciata
- *Goal-Test*: 8 regine sulla scacchiera, nessuna minacciata
- *Costo cammino*: zero
- *Azioni*: aggiungi una regina nella colonna vuota più a destra ancora libera in modo che non sia minacciata

2057 sequenze da considerare (*)

Le 8 regine:

Formulazione a stato completo

- *Goal-Test*: 8 regine già sulla scacchiera, nessuna minacciata
- *Costo cammino*: zero
- *Stati*: scacchiere con 8 regine, una per colonna
- *Azioni*: sposta una regina nella colonna, se minacciata
- **Messaggio**: formulazioni diverse → portano a spazi stati diversi