

# Inteligență Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023

Cursul 8

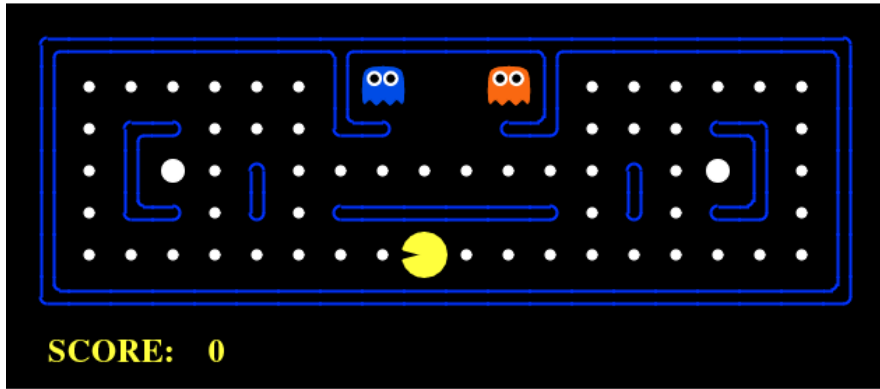
# Recapitulare – cursul trecut

1. Rețele feedforward multistrat de perceptroni
2. Regresia liniară simplă și multiplă

# Cuprinsul cursului de azi

1. Agenți inteligenți și mediile în care aceștia funcționează
2. Rezolvarea problemelor prin căutare
  - Graful stărilor
  - Arborele de căutare

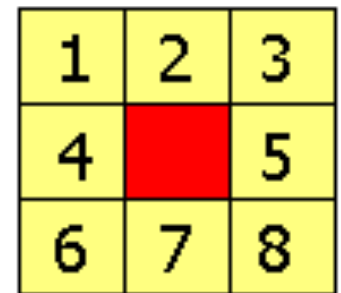
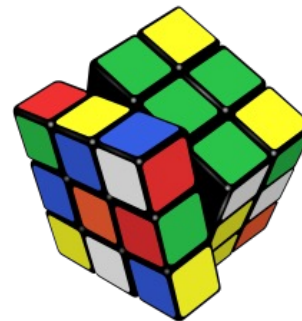
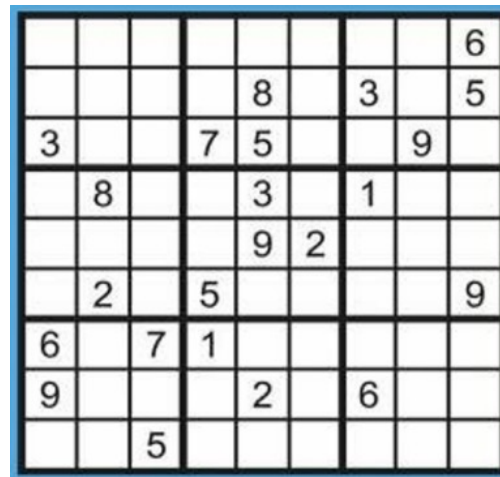
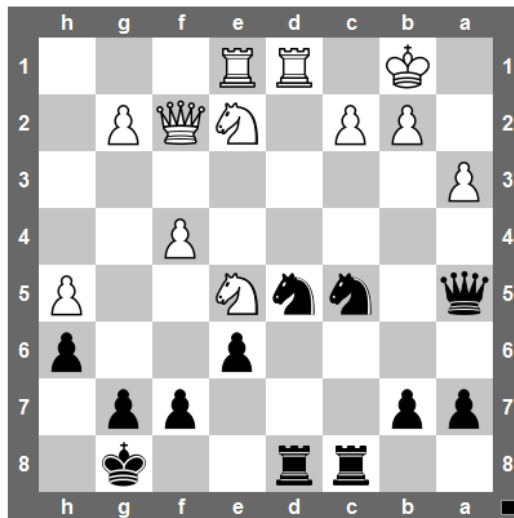
# Agenți inteligenți și mediile în care aceștia funcționează



1	2	3
8		4
7	6	5

# Căutarea

- căutare = tehnică de rezolvare a problemelor ce explorează în mod sistematic un spațiu de stări ale problemei
- exemplu de stare a unei probleme: configurația tablei de sah, a unui cub Rubik, a unui careu Sudoku, a unui puzzle



# Căutarea

- spațiu de căutare = graf (arbore) în care un nod desemnează o soluție parțială iar o muchie reprezintă un pas în construcția unei soluții
- scopul căutării: a găsi un drum în graf de la un *nod-inițial* (configurație inițială) la un *nod-scop* (configurație finală)
- un program reprezintă un agent inteligent; agenții cu care vom lucra vor adopta un *scop* și vor urmări *satisfacerea* lui.

1	2	3
4		5
6	7	8

Nod inițial



	4	3
2	1	5
6	7	8

Nod scop



Nod inițial



Nod scop

# A doua parte a semestrului: curs

## 1. Rezolvarea problemelor prin căutare

Stare Scop Funcție succesor Acțiuni Spațiul stărilor

Arbore de căutare

Căutare neinformată

Căutare informată

## 2. Strategii de căutare neinformată

Căutare în lățime

Căutare în adâncime

Căutare cu cost uniform

Căutare cu adâncime limitată

Căutare cu adâncime incrementală

## 3. Strategii de căutare informată

Euristică

Greedy

A\*

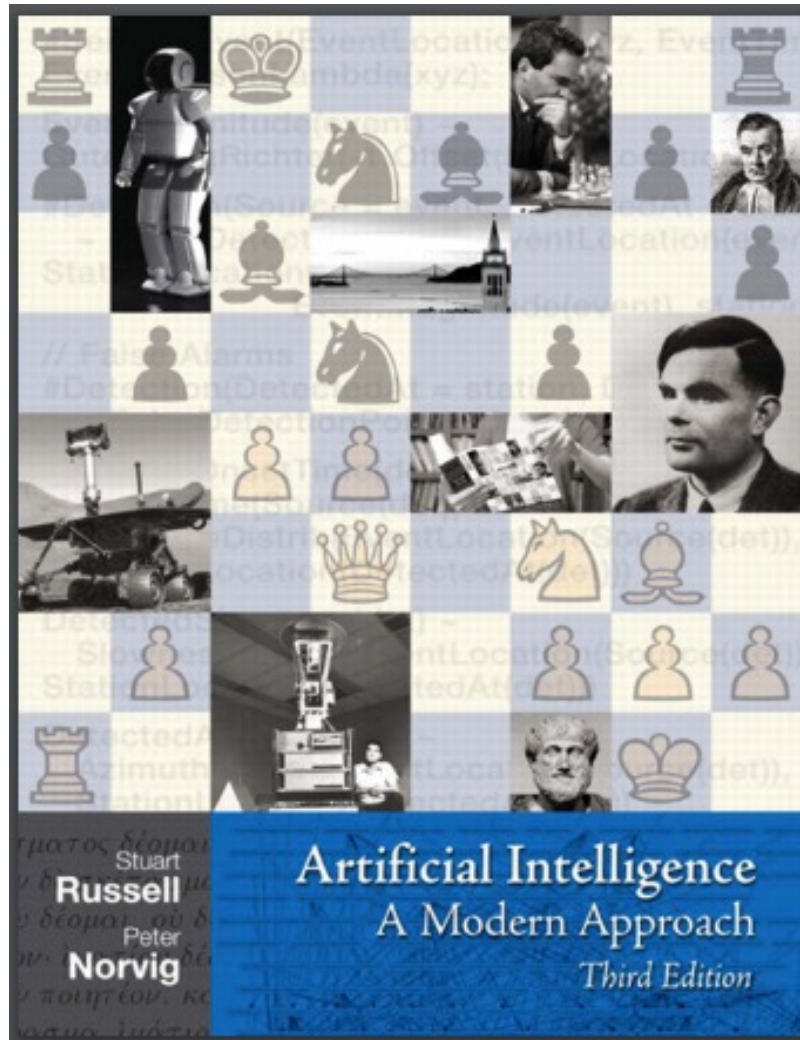
Euristică admisibilă

## 4. Strategii de căutare adversariale

Algoritmul minimax

Alpha-Beta retezare

# Bibliografie



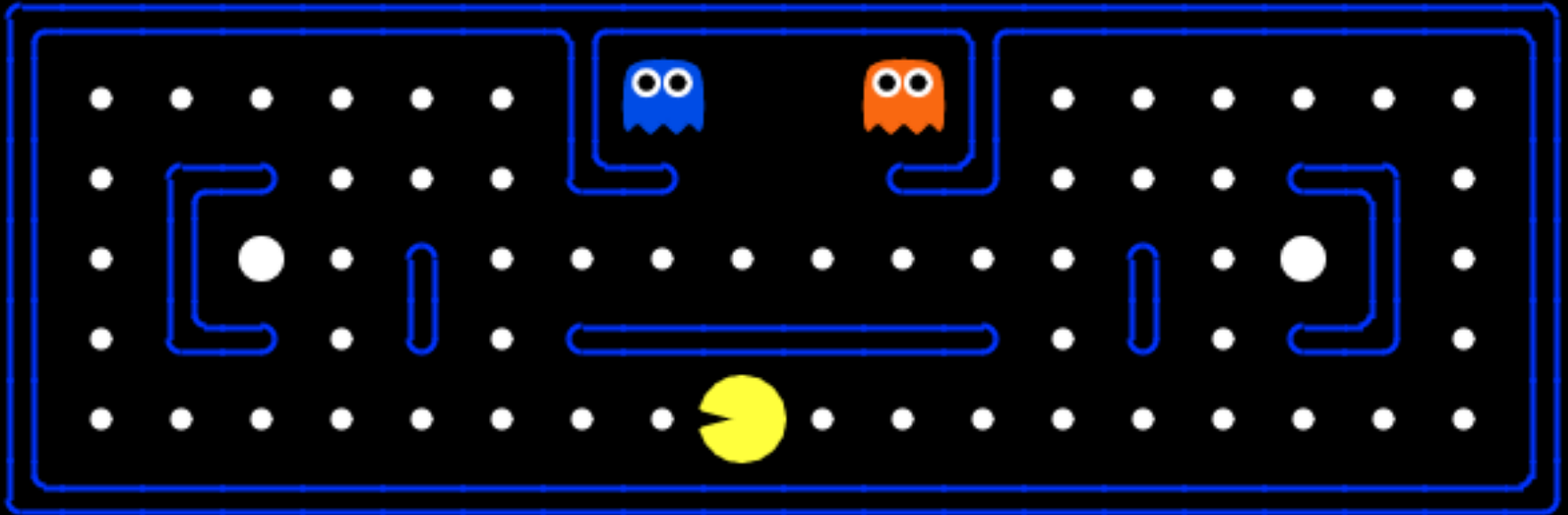
## Prima și a doua parte



# Agenți inteligenți și mediile în care aceștia funcționează

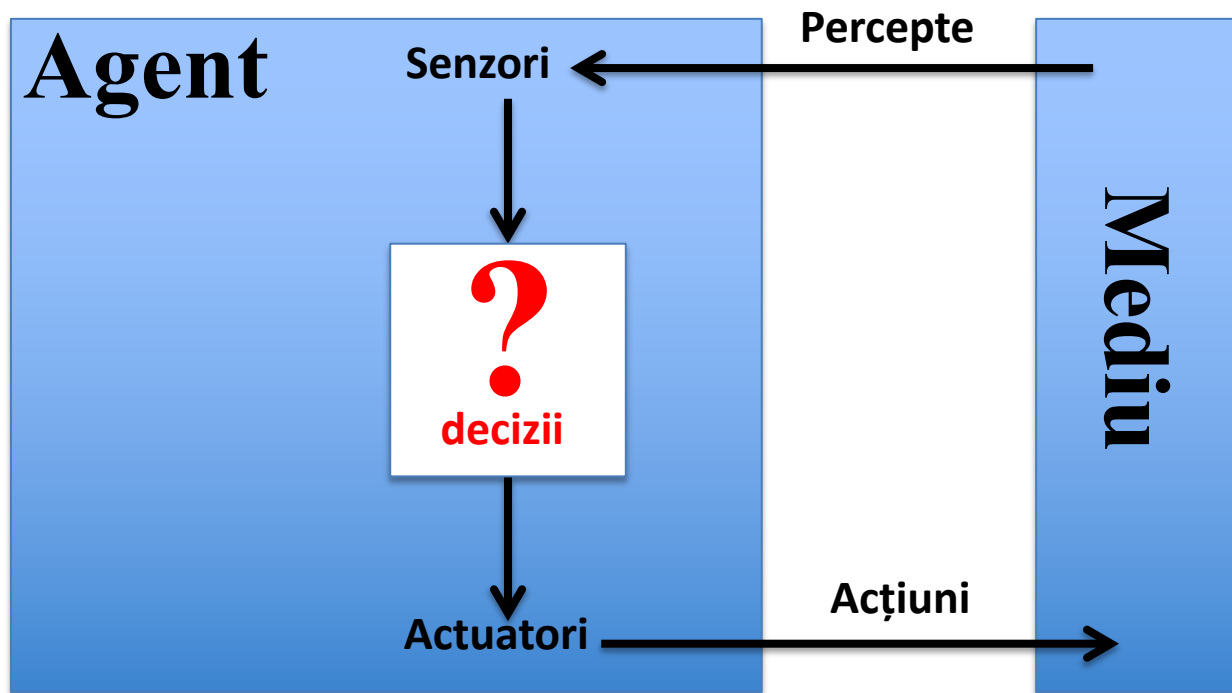


# Agenți inteligenți și mediile în care aceștia funcționează



**SCORE: 0**

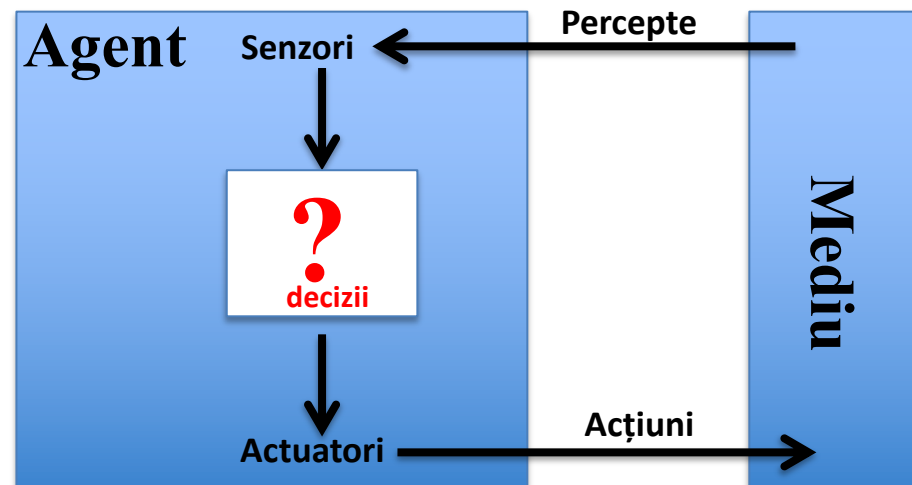
# Agenți inteligenți și mediile în care aceștia funcționează



- Un agent este o entitate care **percepe** și **acționează**.
- **Percepția** – input-urile perceptuale la fiecare moment de timp.
- **Secvența de percepții** – istoria completă a tot ceea ce a fost perceput de către agent

# Agenți inteligenți și mediile în care aceștia funcționează

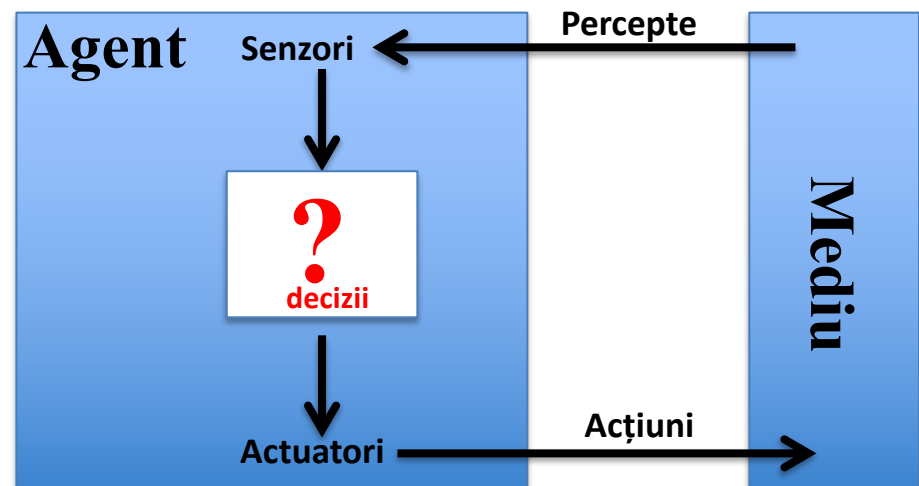
- Scopul celei de-a doua părți a a cursului: studiul unui set de principii pentru construirea unor agenți **raționali**.
- Exemplu de agenți: om, robot, mașină autonomă, program software
- Acțiunile unui agent depind în totalitate numai de ce percepte (input-ul) primește agentul din mediul în care funcționează prin intermediul senzorilor. Comportamentul unui agent este modelat de o funcție  $f$  care transformă cunoașterea agentului (orice secvență de percepte) în acțiuni



$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

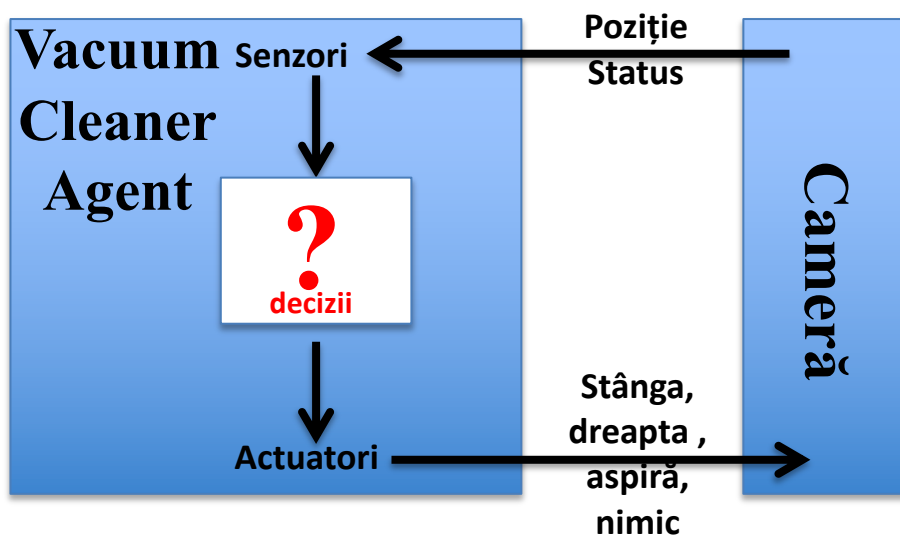
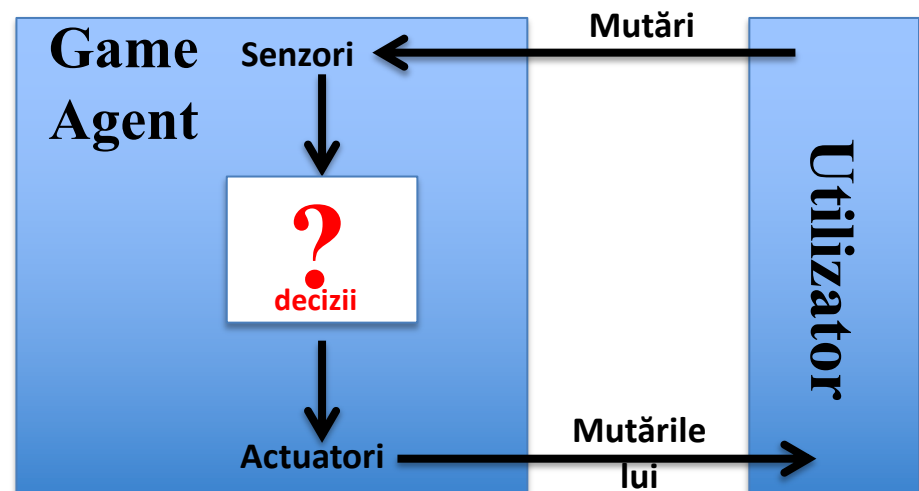
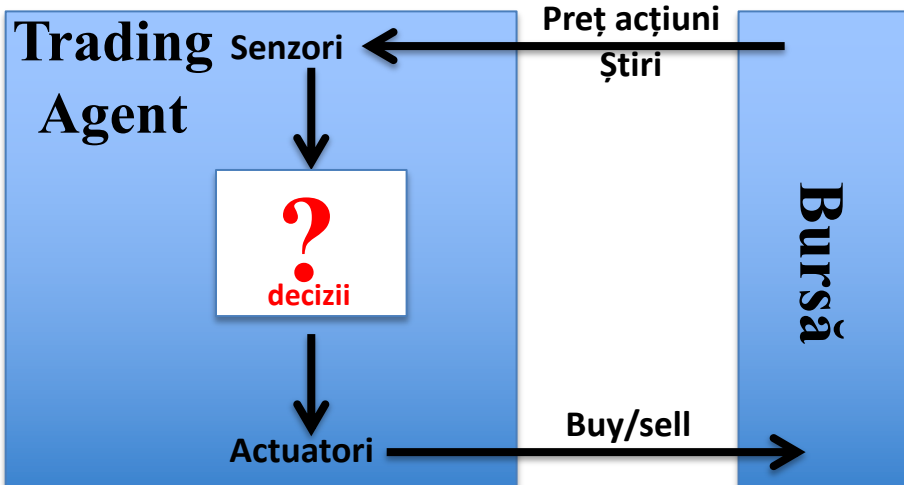
# Agenți inteligenți și mediile în care aceștia funcționează

- Scopul celei de-a doua părți a a cursului: studiul unui set de principii pentru construirea unor agenți **raționali**.
- Exemplu de agenți: om, robot, mașină autonomă, program software
- Vrem să găsim funcția **f** prin care agentul ia decizii
- Mediu → Senzori → Decizii → Actuatori

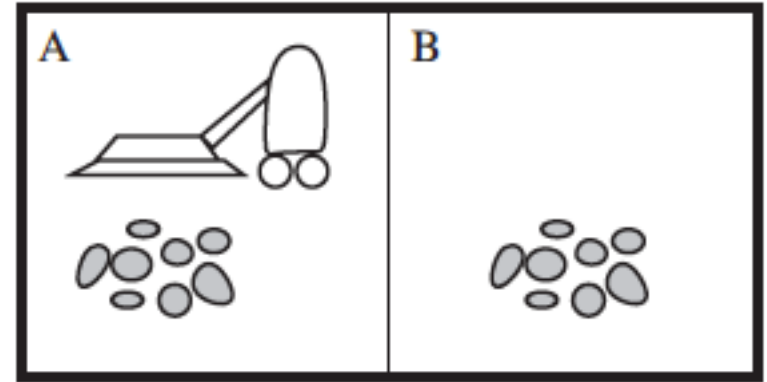
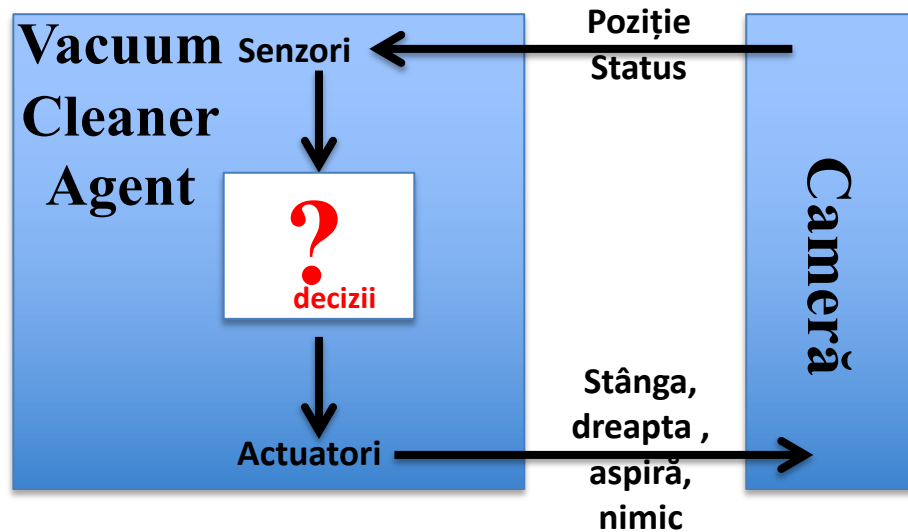


*Ciclul percepte – acțiuni*

# Agenți inteligenți și mediile în care aceștia funcționează



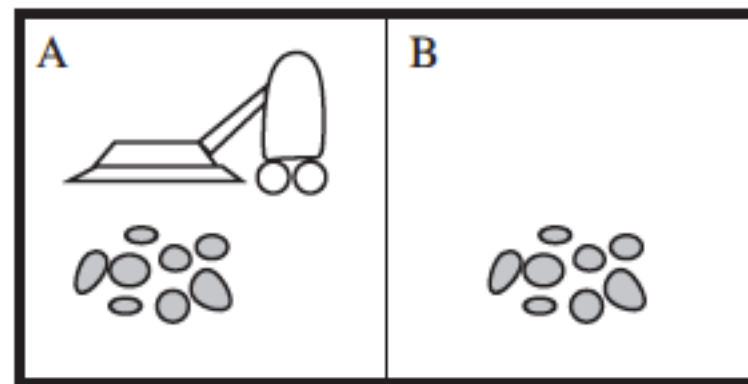
# Un agent aspirator



- Aspiratorul percepe poziția (A sau B) și starea (curat sau murdar)
- Acțiuni: mergi în stânga, mergi în dreapta, aspiră, nimic.
- Consideră următoarea funcție  $f: P^* \rightarrow A$ , dacă poziția curentă este murdară atunci aspiră, altfel mută-te la cealaltă poziție
- Putem tabela funcția  $f$

# Un agent aspirator

Secvența de percepte	Acțiuni
[A, curat]	Dreapta
[A, murdar]	Aspira
[B, curat]	Stanga
[B, murdar]	Aspira
[A, curat], [B,curat]	Stanga
[A, curat], [B, murdar]	Aspira
...	...



- Aspiratorul percepe poziția (A sau B) și starea (curat sau murdar)
- Acțiuni: mergi în stânga, mergi în dreapta, aspiră, nimic.
- Consideră următoarea funcție  $f: P^* \rightarrow A$ , dacă poziția curentă este murdară atunci aspiră, altfel mută-te la cealaltă poziție
- Putem tabela funcția  $f$



# Proprietăți ale mediilor în care funcționează agenții

- **Complet observabil vs parțial observabil:** la orice moment în timp mediul este complet observabil dacă senzorii agentului îi dau acces complet la starea mediului la fiecare moment și îi oferă toate aspectele relevante în luarea deciziei optime.
  - șah vs poker
- **Determinist vs stochastic:** dacă următoarea stare a mediului este în întregime determinată de starea curentă și de acțiunile agentului, mediul este determinist.
  - șah vs poker

# Proprietăți ale mediilor în care funcționează agenții

- **Discret vs continuu**: dacă numărul de percepții și de acțiuni ale agentului sunt finite atunci mediul este discret.
  - șah vs condusul unei mașini
- **Episodic vs secvențial**: într-un mediu episodic, experiența agentului este împărțită în episoade independente. Un episod constă în percepție urmată de o singură acțiune. Episoadele următoare nu depind de acțiunile episoadelor anterioare.
  - corectarea unui test grilă vs șah

# Proprietăți ale mediilor în care funcționează agenții

- **Static vs dinamic**: dacă mediul se schimbă cât timp agentul decide ce acțiuni să facă mediul este dinamic.
  - rezolvarea unui rebus vs condusul unei mașini
- **Agent vs multiagent**: dacă sunt mai mulți agenți mediul este multiagent (sisteme multiagent competitive vs sisteme multiagent cooperative).
  - corectarea unui test grilă vs șah

# Jocul de dame



Complet observabil vs parțial observabil

Determinist vs stochastic

Discret vs continuu

Episodic vs secvențial

Static vs dinamic

Agent vs multiagent

Complet observabil

Determinist

Discret

Secvențial

Static

Multiagent competitiv

# Jocul de poker



Complet observabil vs parțial observabil

Determinist vs stochastic

Discret vs continuu

Episodic vs secvențial

Static vs dinamic

Agent vs multiagent

Parțial observabil

Stochastic

Discret

Secvențial

Static

Multiagent competitiv

# 8 - puzzle

1	2	3
8		4
7	6	5

Complet observabil vs parțial observabil

Determinist vs stochastic

Discret vs continuu

Episodic vs secvențial

Static vs dinamic

Agent vs multiagent

Complet observabil

Determinist

Discret

Secvențial

Static

Agent

# Mașină autonomă



Complet observabil vs parțial observabil

Determinist vs stochastic

Discret vs continuu

Episodic vs secvențial

Static vs dinamic

Agent vs multiagent

Parțial observabil

Stochastic

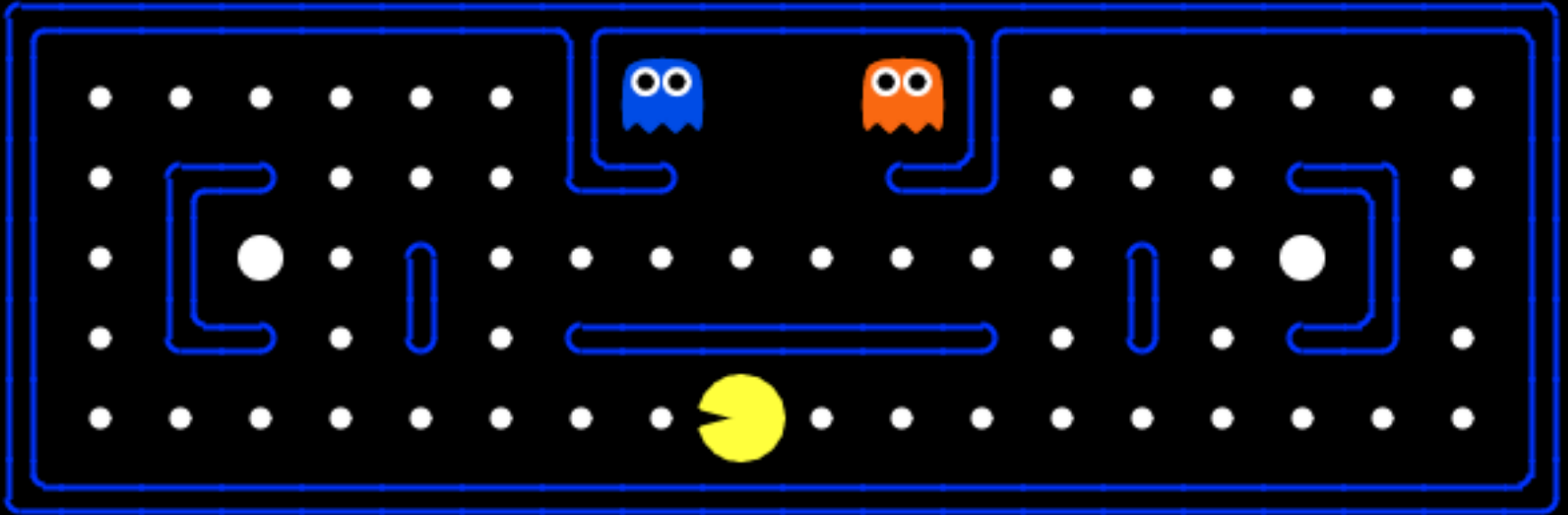
Continuu

Secvențial

Dinamic

Multiagent cooperativ

# PAC MAN



**SCORE: 0**

Complet observabil vs parțial observabil

Determinist vs stochastic

Discret vs continuu

Episodic vs secvențial

Static vs dinamic

Agent vs multiagent

Complet observabil

Stochastic

Discret

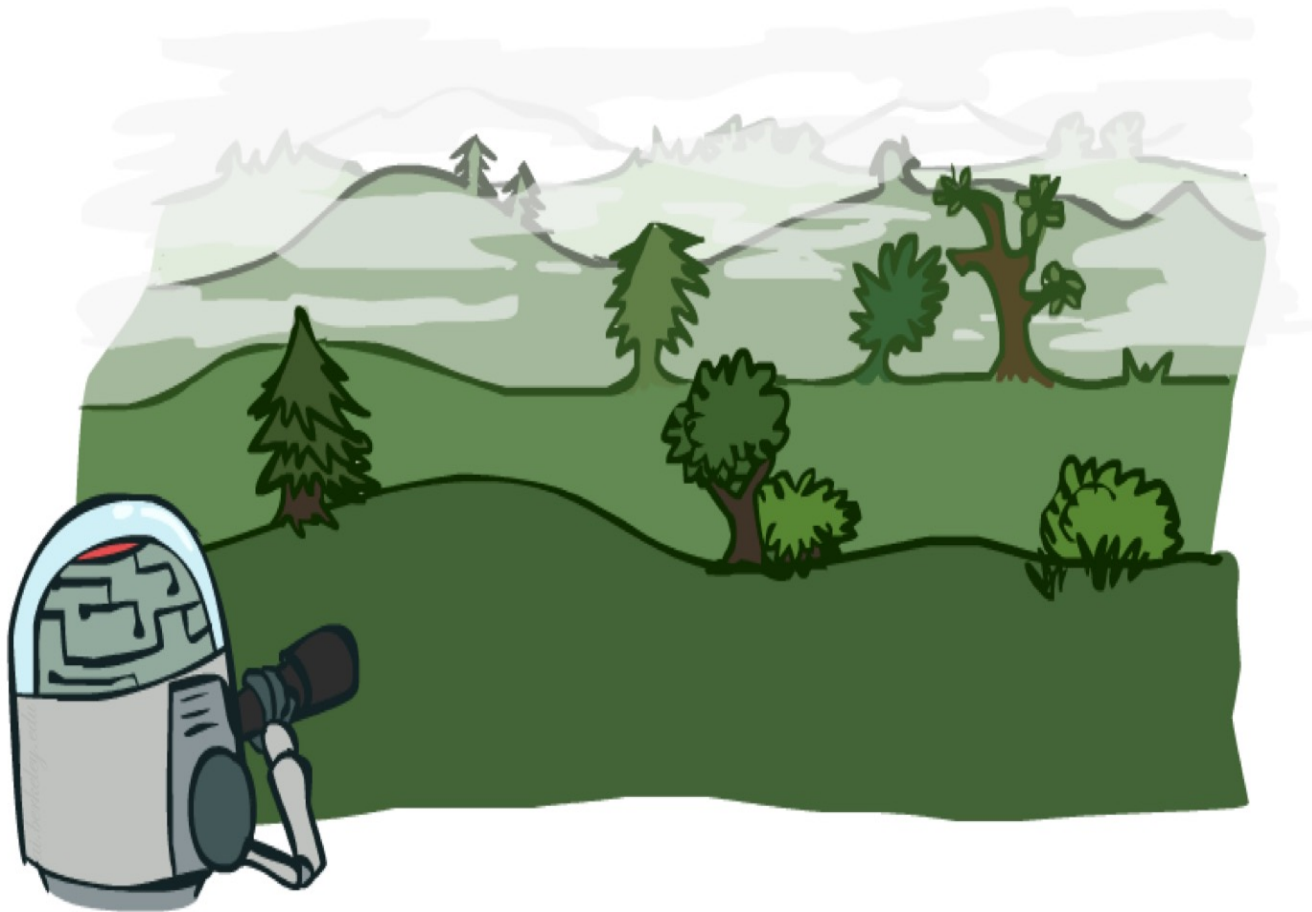
Secvențial

Dinamic

Multiagent competitiv



# Probleme de căutare



# Rezolvarea problemelor prin căutare

- primul pas: formularea scopului
  - un scop este o mulțime de stări ale universului în care vrem să ajungem
- al doilea pas: formularea problemei
  - ce acțiuni (operatori) poate face agentul dintr-o anumită stare
  - stare = configurație a lumii înconjurătoare (a universului problemei)
  - acțiunile generează tranziția între stări ale universului
  - o acțiune se poate defini printr-o funcție succesor:  $S(x)$  = mulțimea stărilor în care se poate ajunge din  $x$  printr-o unică acțiune
  - ce stări trebuiesc luate în considerare
- algoritm de căutare
  - primește ca input o problemă
  - întoarce ca output o soluție sub forma unei succesiuni de acțiuni

# Formularea unei probleme

- Spațiul stărilor
  - stare = configurație a lumii înconjurătoare
  - mulțimea stărilor în care se poate ajunge plecând din starea inițială, prin intermediul oricărei secvențe de acțiuni
  - ce stări trebuie luate în considerare
  - stare inițială
  - stare scop
- Acțiuni
  - ce acțiuni (operatori) poate face agentul dintr-o anumită stare
  - acțiunile generează tranziția între stări ale universului
  - uneori acțiunile au un cost asociat
- Funcția succesori
  - $SUCCESSOR(x, a)$  = mulțimea stărilor în care se poate ajunge din starea  $x$  prin acțiunea  $a$

# Problema 8-puzzle

## Descrierea problemei

Pe o tablă  $3 \times 3$  se găsesc 8 piese, numerotate de la 1 la 8. La fiecare moment o singură piesă se poate mișca cu o poziție, pe orizontală sau verticală, în limitele cadrului tablei, în locul rămas liber (poziția roșie). Se dă o configurație inițială și una finală a tablei. Trebuie să se găsească secvența de mutări care să aducă piesele din configurația inițială în cea finală.

2		3
1	8	4
7	6	5

Stare inițială

[2, X, 3, 1, 8, 4, 7, 6, 5]



1	2	3
8		4
7	6	5

Stare scop

[1, 2, 3, 8, X, 4, 7, 6, 5]

# Formularea problemei 8-puzzle

- Spațiul stărilor

- stare = configurație a pătratului  $3 \times 3$
- $9!/2$  stări posibile
- stare inițială
- stare scop

2		3
1	8	4
7	6	5

Stare inițială

[2, X, 3, 1, 8, 4, 7, 6, 5]

1	2	3
8		4
7	6	5

Stare scop

[1, 2, 3, 8, X, 4, 7, 6, 5]

- Acțiuni

- STÂNGA, JOS, DREAPTA, SUS
- fiecare acțiune (mutare) are cost = 1

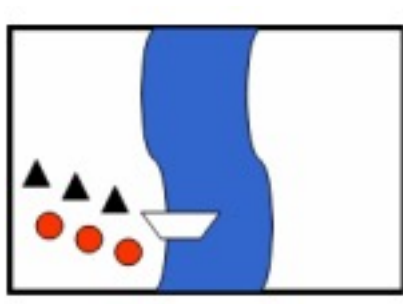
- Funcția succesori

- $SUCCESSOR([X, a, b, c, d, e, f, g, h], STANGA) = \text{NU EXISTA}$
- $SUCCESSOR([X, a, b, c, d, e, f, g, h], SUS) = \text{NU EXISTA}$
- $SUCCESSOR([X, a, b, c, d, e, f, g, h], DREAPTA) = [a, X, b, c, d, e, f, g, h]$
- $SUCCESSOR([X, a, b, c, d, e, f, g, h], JOS) = [c, a, b, X, d, e, f, g, h]$
- ...

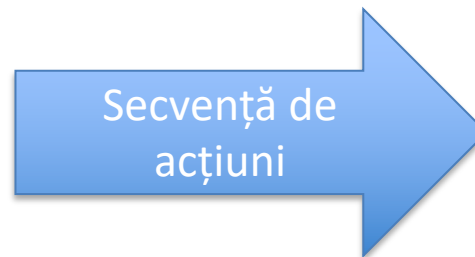
# Problema misionarilor și a canibalilor

## Descrierea problemei

**Trei misionari** și **trei canibali** se află la marginea unui râu, cu scopul de a trece pe celălalt mal. Ei au la dispoziție o barcă de două persoane. Dacă la un moment dat, pe un mal sau pe celălalt, numărul canibalilor este mai mare decât cel al misionarilor, canibalii îi vor mânca pe misionari. Problema constă în a afla cum pot trece cele 6 persoane în deplină siguranță de pe un mal pe celălalt😊. Barca nu merge singură, este nevoie de cel puțin o persoană în barcă.



Stare inițială

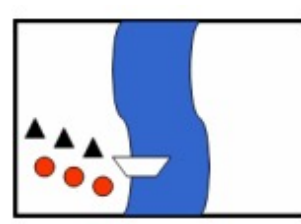


Stare scop

# Formularea problemei misionarilor și a canibalilor

- Spațiul stărilor

- triplet  $(m, c, b)$ :  $m$  = numărul de misionari pe malul stâng,  $c$  = numărul de canibali pe malul stâng,  $b$  = prezența bărcii (0 sau 1) pe malul stâng
- stare inițială:  $(3,3,1)$
- stare scop:  $(0,0,0)$



Stare inițială



Stare scop

- Acțiuni

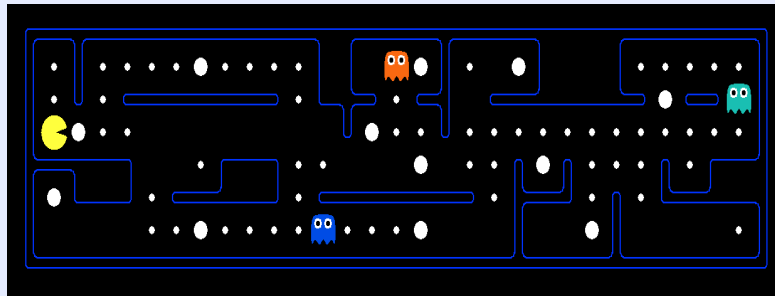
- pot duce de pe un mal către celălalt mal 1 sau 2 misionari, 1 sau 2 canibali, 1 misionar + 1 canibal ( $1m, 2m, 1c, 2c, 1m+1c$ )

- Funcția succesor

- $\text{Succesor}((m, c, b), 1c) = (m, c + 1 - 2b, 1 - b), \dots$
- $\text{Succesor}((m, c, b), 1m) = (m + 1 - 2b, c, 1 - b), \dots$
- unele stări sunt invalide  $(2,3,0)$  sau ilegale  $(0,0,1)$

# Ce conține spațiul stărilor?

O **stare a lumii** include fiecare detaliu al mediului



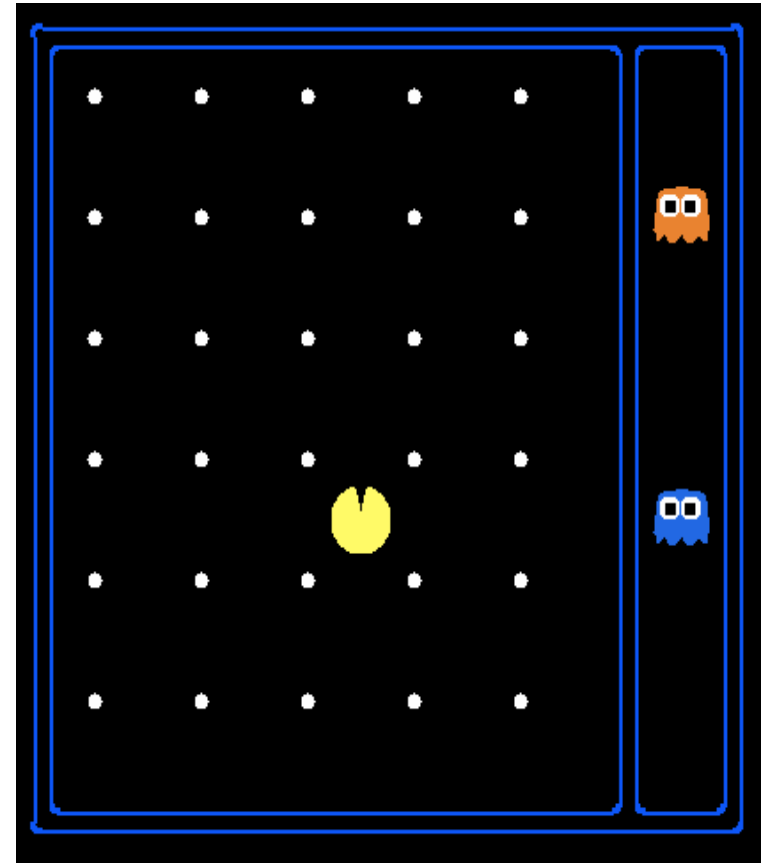
O **stare pentru căutare** păstrează numai detaliile relevante planificării (abstractizare)

- Problema: vreau să ajung într-un anumit punct (Pathing)
  - stări: poziția  $(x,y)$
  - acțiuni: Nord Sud Est Vest
  - funcție succesor: actualizează numai poziția
  - test scop:  $(x,y)=END$  ?
- Problemă: Mănâncă toate punctele albe (Eat-all-dots)
  - stări:  $\{(x,y), \text{var. booleană pentru puncte albe}\}$
  - acțiuni: Nord Sud Est Vest
  - succesor: actualizează poziția + posibil variabila booleană
  - test scop: toate variabilele booleene sunt false

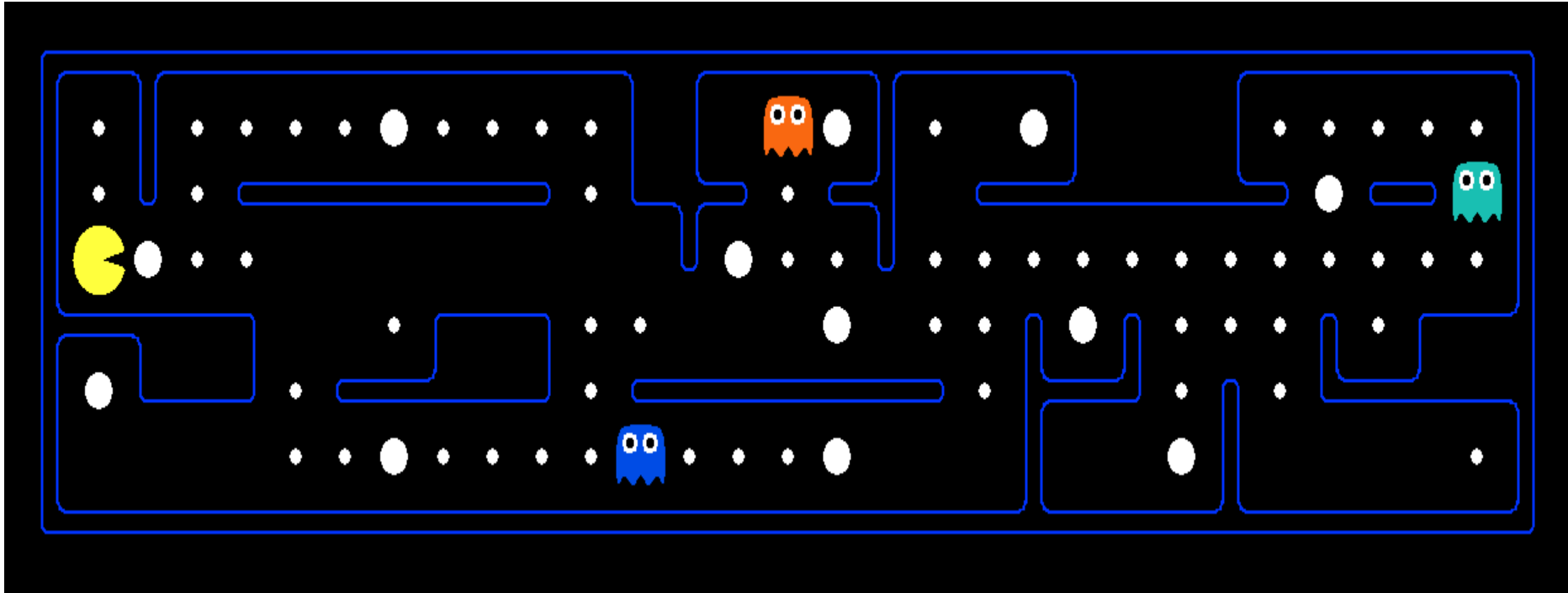


# Dimensiunea spațiului stărilor

- Stările lumii:
  - pozițiile agentului: 120
  - numărul de puncte albe: 30
  - poziții fantome: 12
  - orientarea agentului: NSEW
- Câte:
  - stări ale lumii?  
 $120 \times (2^{30}) \times (12^2) \times 4$
  - stări pentru problema Pathing?  
120
  - stări pentru problema Eat-all-dots?  
 $120 \times (2^{30})$



# Safe Passage Problem



- Problemă: mănâncă toate punctele iar în tot acest timp fantomele trebuie să fie speriate
- Ce trebuie să specificăm pentru o stare?
  - (poziția agentului, var. booleană pentru puncte albe, var. booleană pentru putere puncte albe, timp rămas ca fantomele să fie speriate)

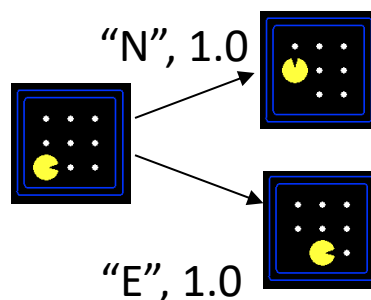
# Probleme de căutare - sumar

- O **problemă de căutare** cuprinde:

- un spațiu al stărilor



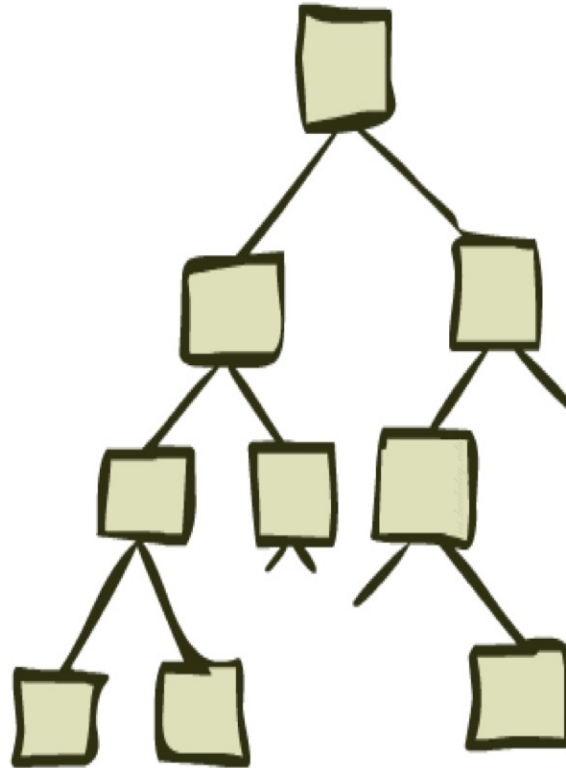
- o funcție succesor  
(cu acțiuni, costuri)



- o stare inițială și un scop care poate fi verificat

- o **soluție** este o secvență de acțiuni (un plan) care transformă starea inițială într-o stare scop

# Graful de stări și arborele de căutare

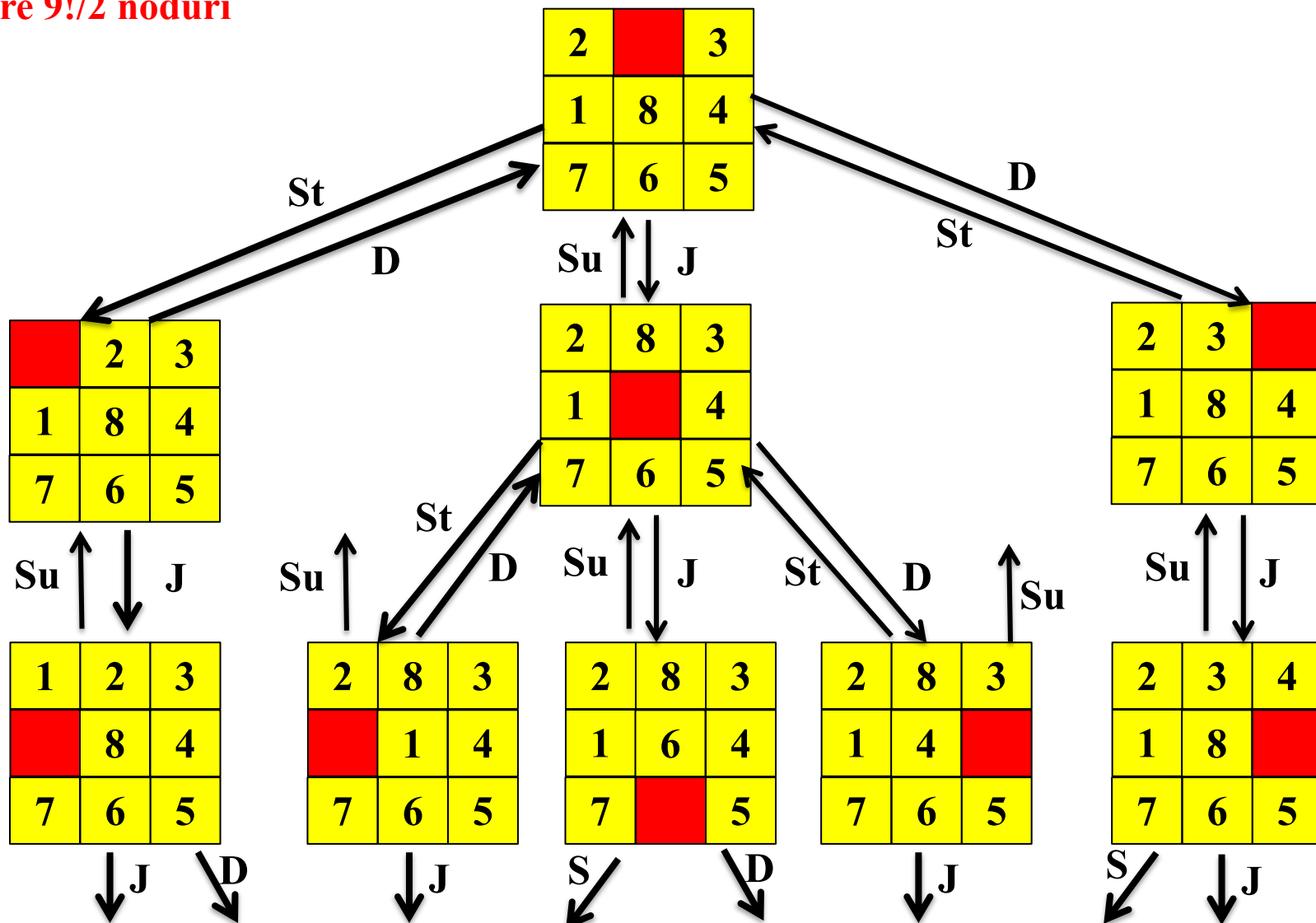


# Graful de stări al unei probleme

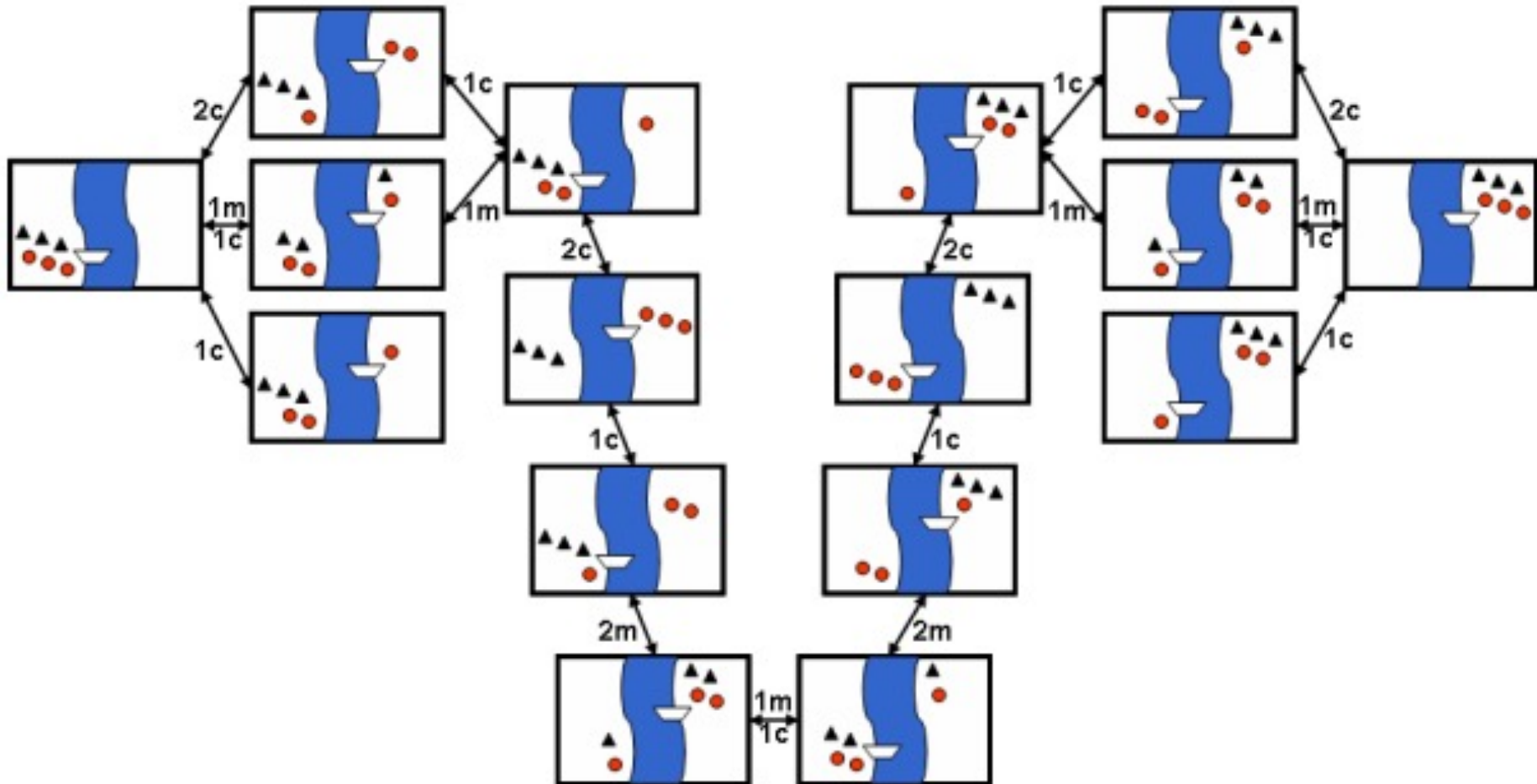
- Graf orientat care conține toate stările lumii (o stare = un nod)
- Fiecare stare apare o singură dată
- Arcele reprezintă acțiuni
  - modelează funcția succesor
  - în ce stări putem ajunge din starea curentă realizând o anumită acțiune
  - deseori avem arc de la starea A la starea B și invers (acțiuni simetrice)

# Graful de stări al problemei 8 - puzzle

Are  $9!/2$  noduri



# Graful de stări al problemei misionarilor și a canibalilor



# Arborele de căutare al unei probleme

- Tot un graf orientat, dar organizat pe nivele:
  - nivelul 0 = rădăcina (starea inițială)
  - nivelul  $i$  = nod obținut prin traversarea a  $i$  arce de la rădăcină din graful de stări
- Un nod desemnează o soluție parțială
- Un arc reprezintă un pas în construirea unei soluții
- Nodurile se pot repeta (spre deosebire de graful stărilor)
  - pot ajunge într-o stare urmând drumuri diferite
- Vrem să găsim un drum (= soluție) în graful de stări de la starea inițială la o stare scop (stare finală).



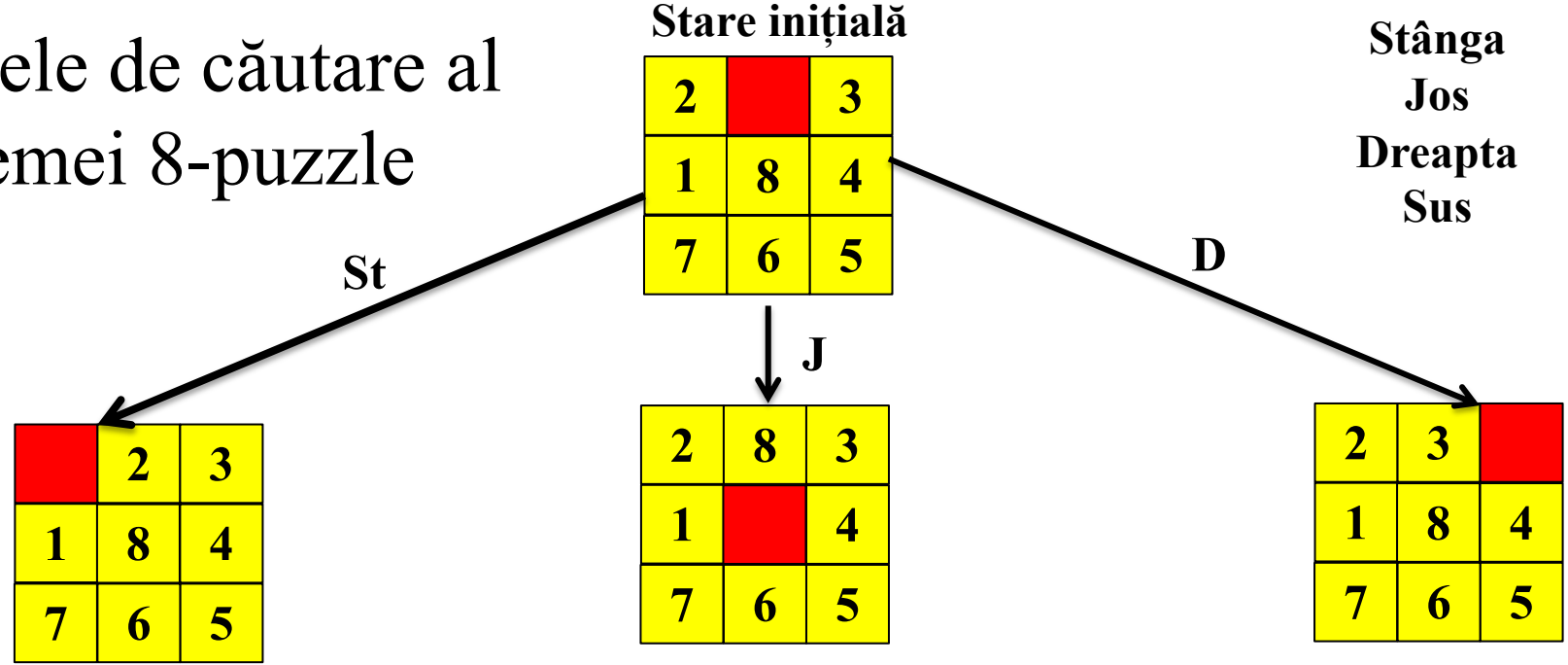
# Arborele de căutare al problemei 8-puzzle

**Stare inițială**

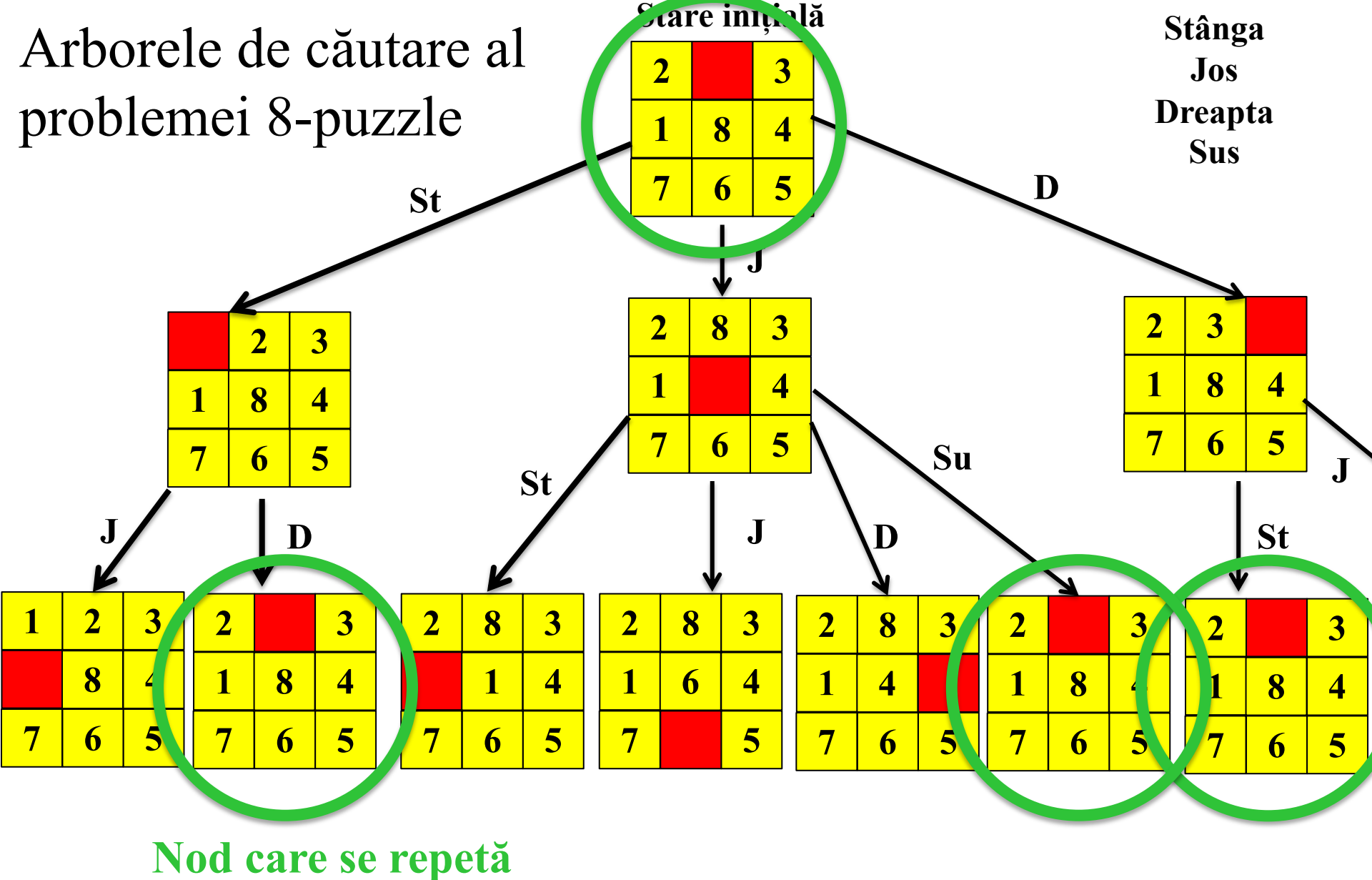
<b>2</b>		<b>3</b>
<b>1</b>	<b>8</b>	<b>4</b>
<b>7</b>	<b>6</b>	<b>5</b>

**Stânga**  
**Jos**  
**Dreapta**  
**Sus**

# Arborele de căutare al problemei 8-puzzle

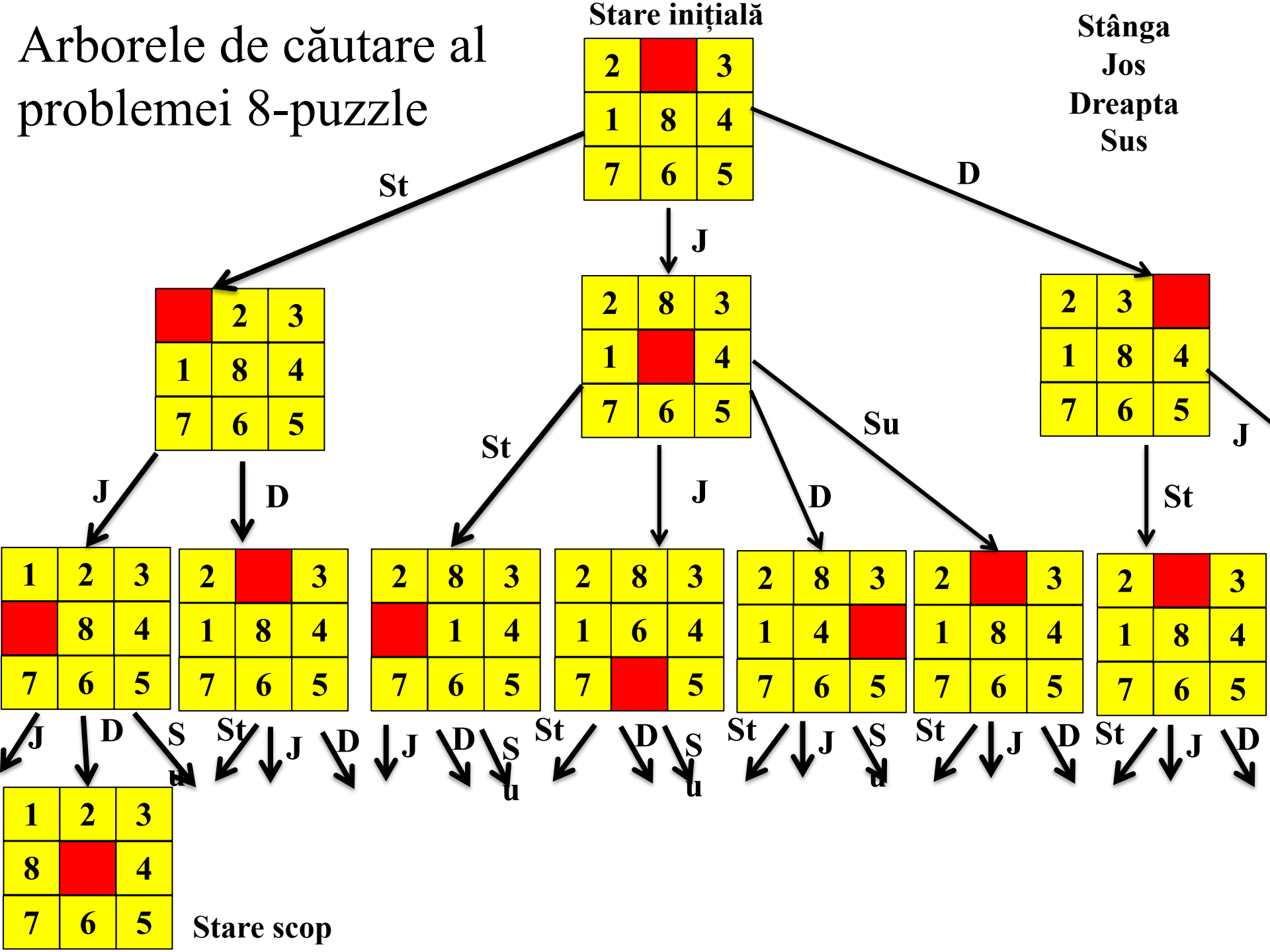


# Arborele de căutare al problemei 8-puzzle



## Nod care se repetă

# Arborele de căutare al problemei 8-puzzle



# Arborele de căutare al problemei misionarilor și canibalilor

1c

1m

1m+1c

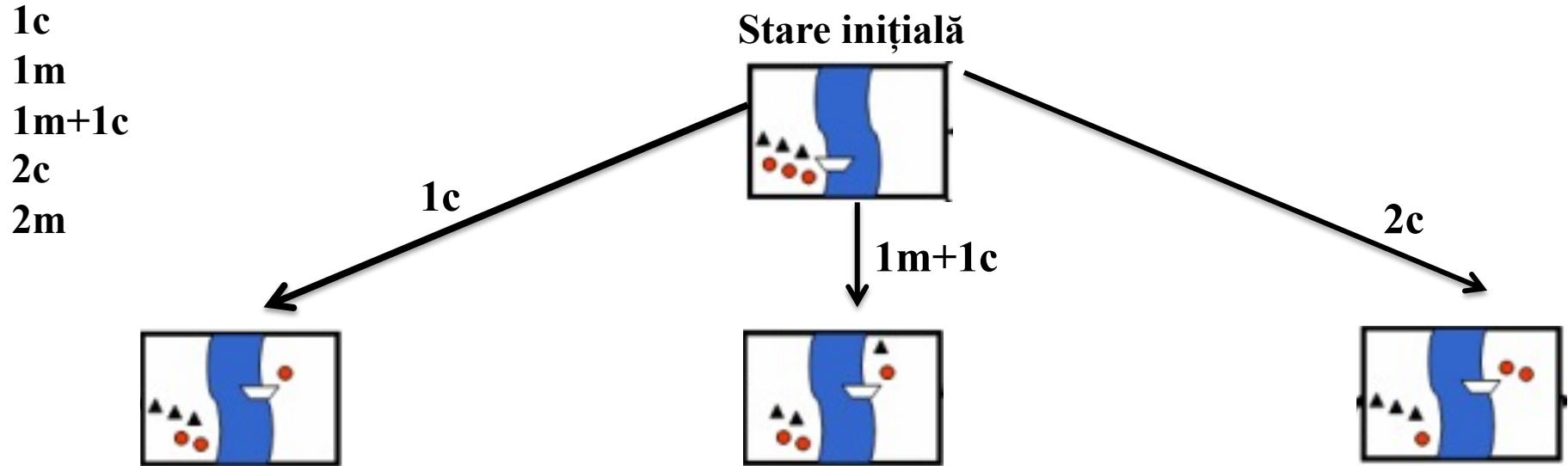
2c

2m

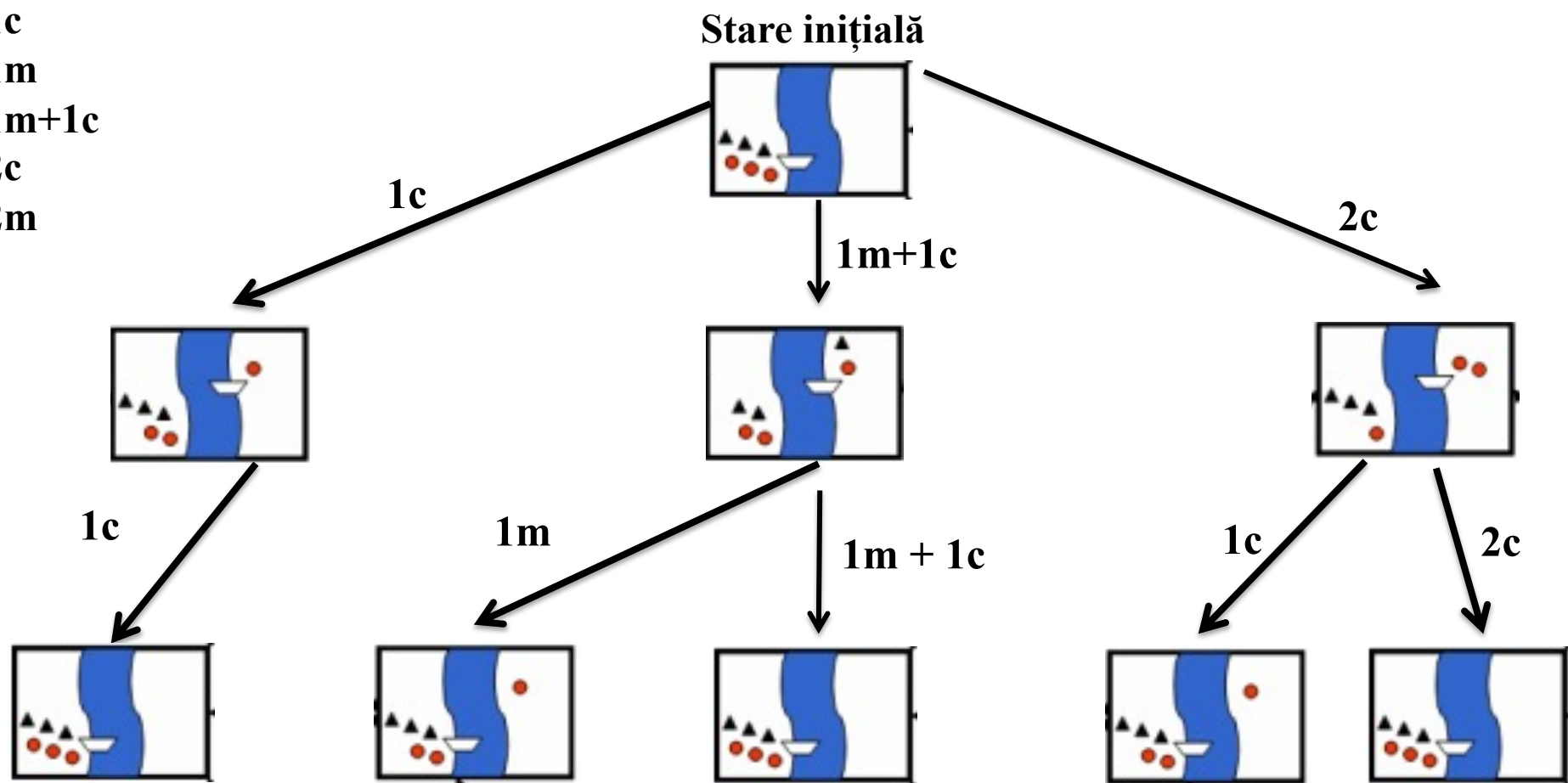
Stare inițială



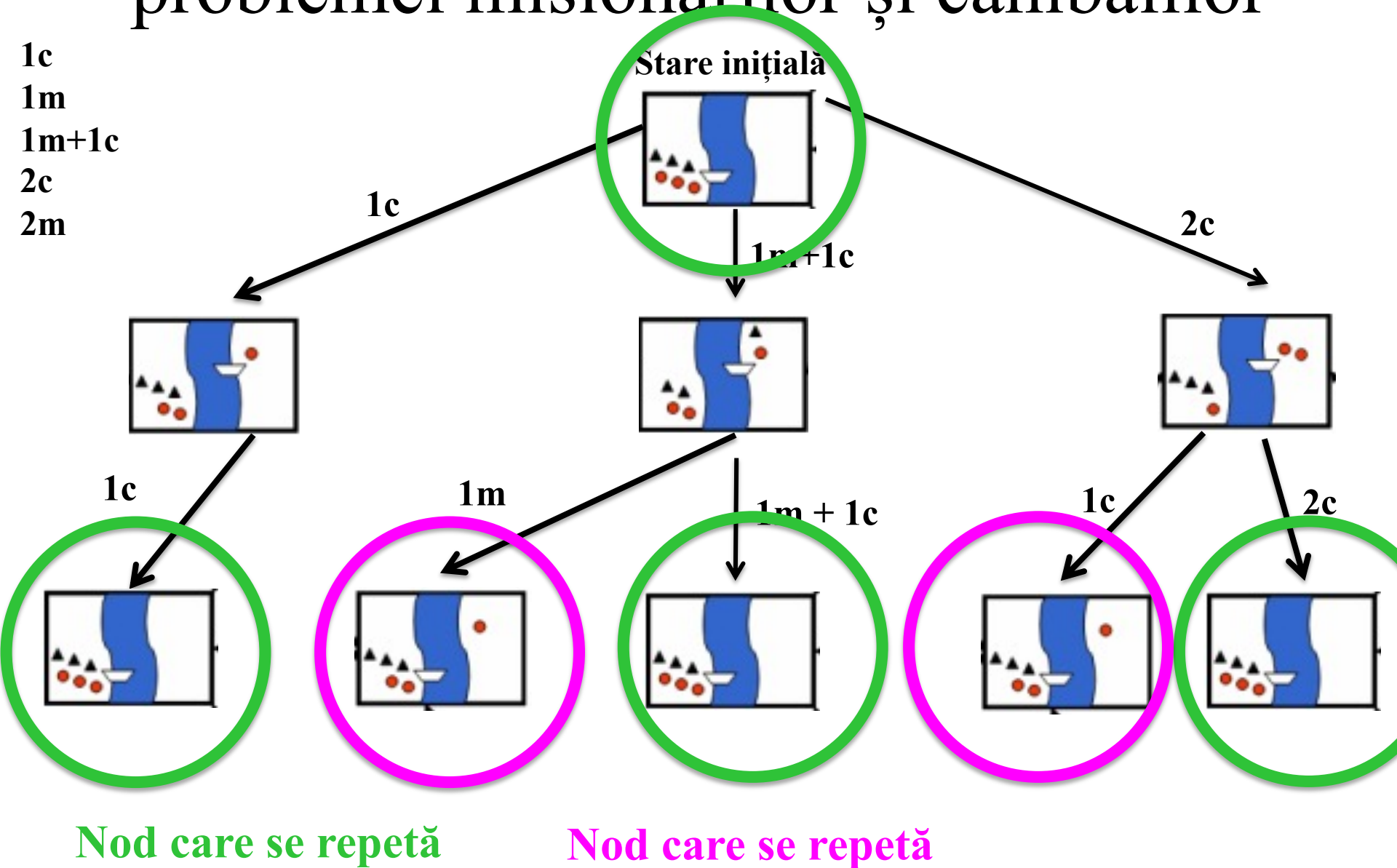
# Arborele de căutare al problemei misionarilor și canibalilor



# Arborele de căutare al problemei misionarilor și canibalilor

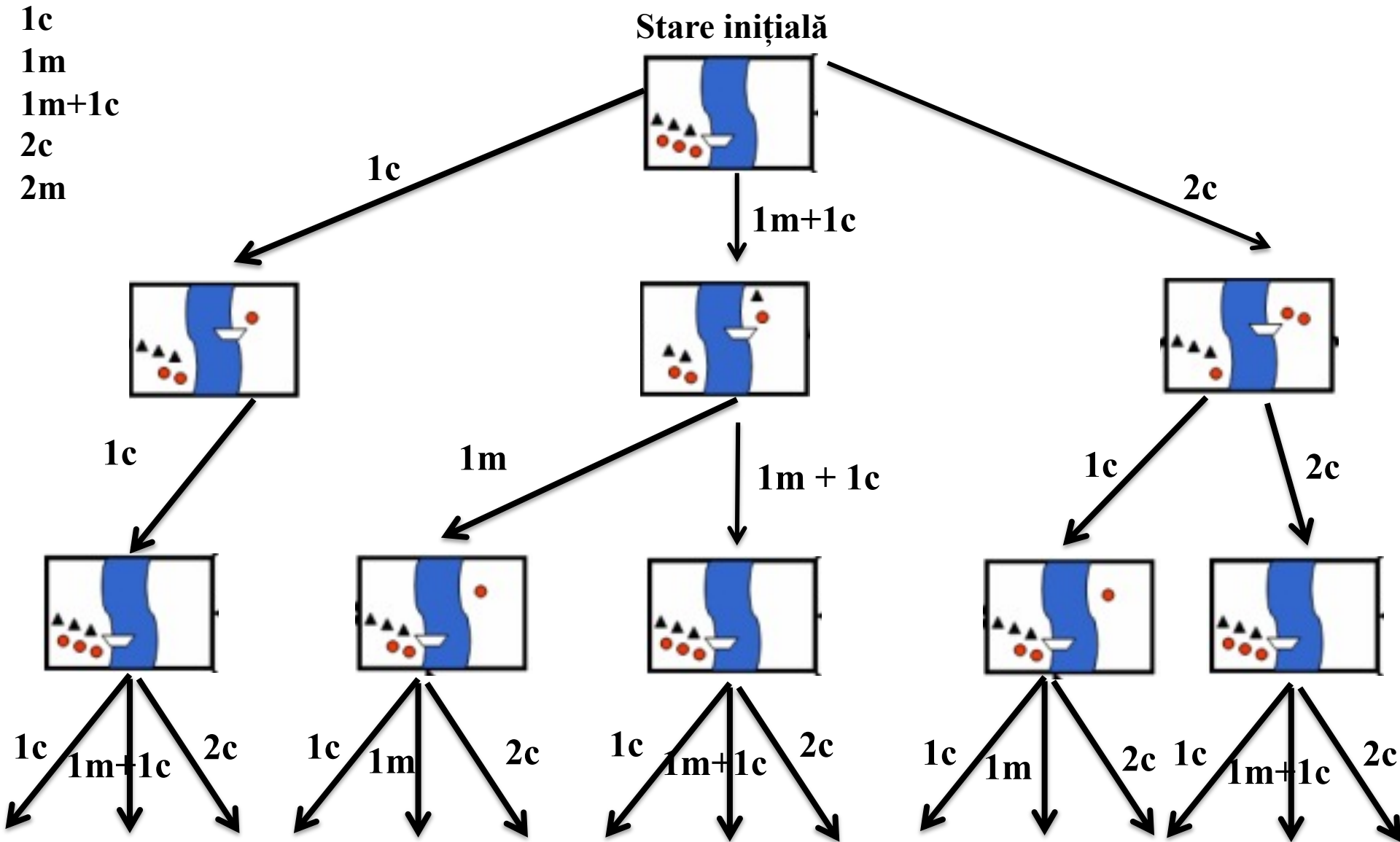


# Arborele de căutare al problemei misionarilor și canibalilor



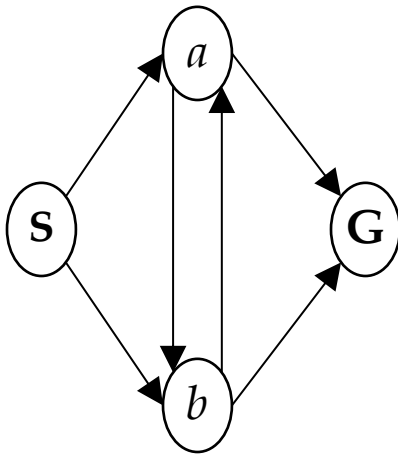


# Arborele de căutare al problemei misionarilor și canibalilor



# Graful de stări vs. Arborele de căutare

Considerăm graful cu 4 stări de mai jos:

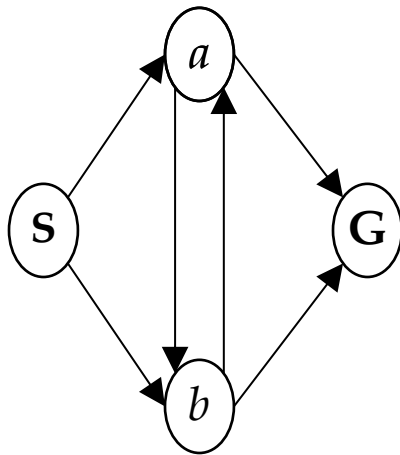


Cât de mare este arborele de căutare (pornim din S)

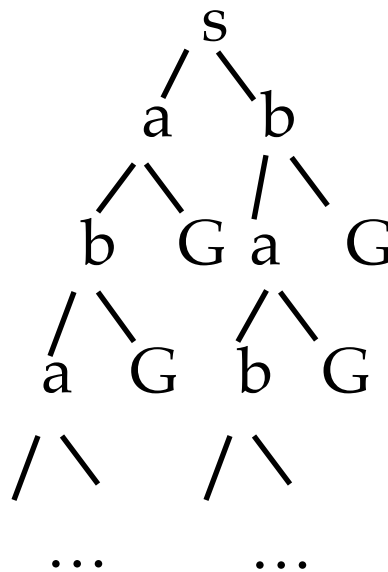


# Graful de stări vs. Arborele de căutare

Considerăm graful cu 4 stări de mai jos:



Cât de mare este arborele de căutare (pornim din S)



Important: foarte multe structuri repetitive în arborele de căutare