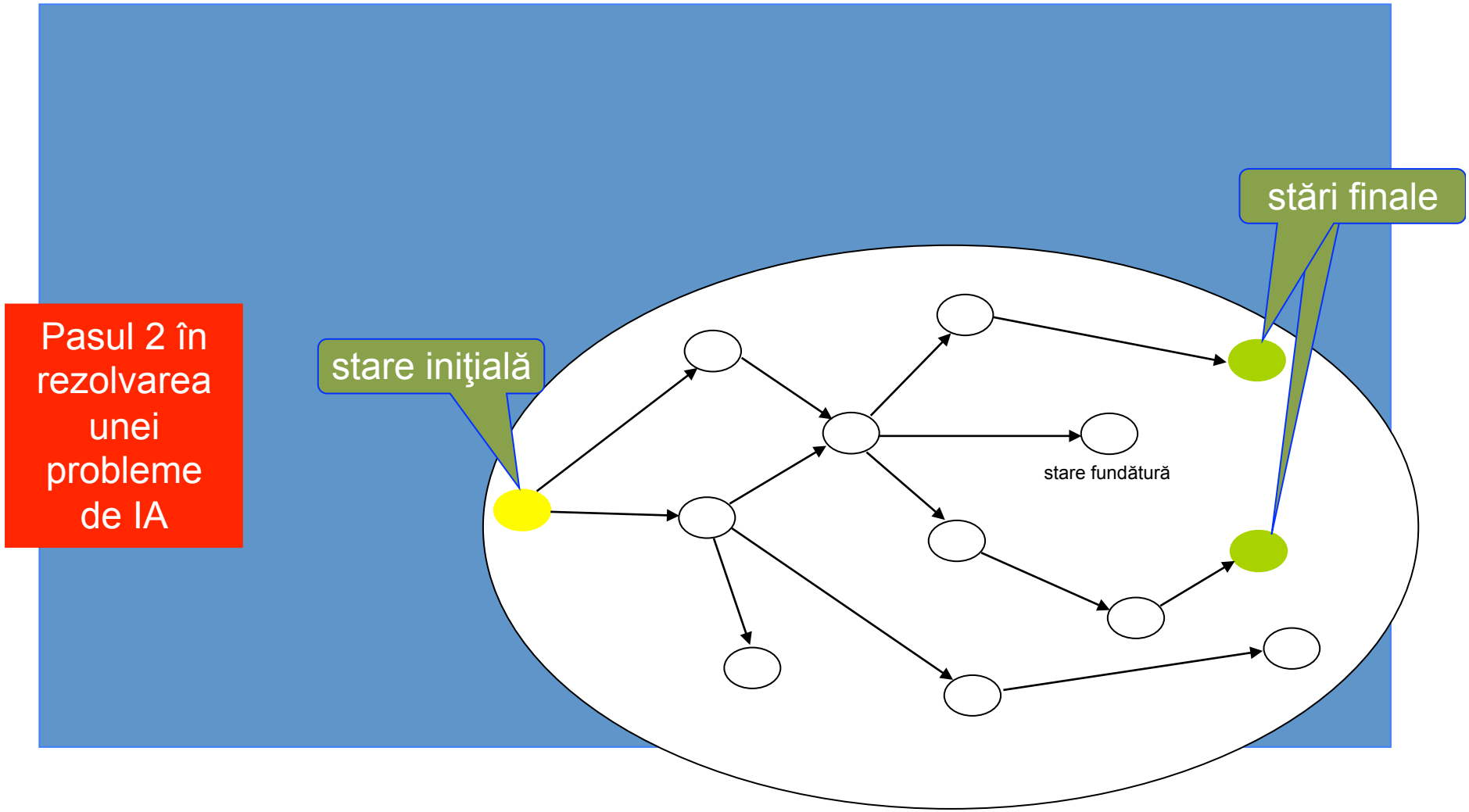
Structura unui motor de inferență bazat pe reguli

Curs 2

Cele 5 cerinţe în modelarea unei probleme de IA

- Pasul 1 — Diferenţiază problema generală de instanţele ei
- Pasul 2 — Recunoaşte o stare şi apreciază dimensiunea spaţiului stărilor
- Pasul 3 — Găseşte cea mai adecvată reprezentare a stărilor
- Pasul 4 — Reprezintă tranziţiile dintre stări
- Pasul 5 — Alege o strategie de control

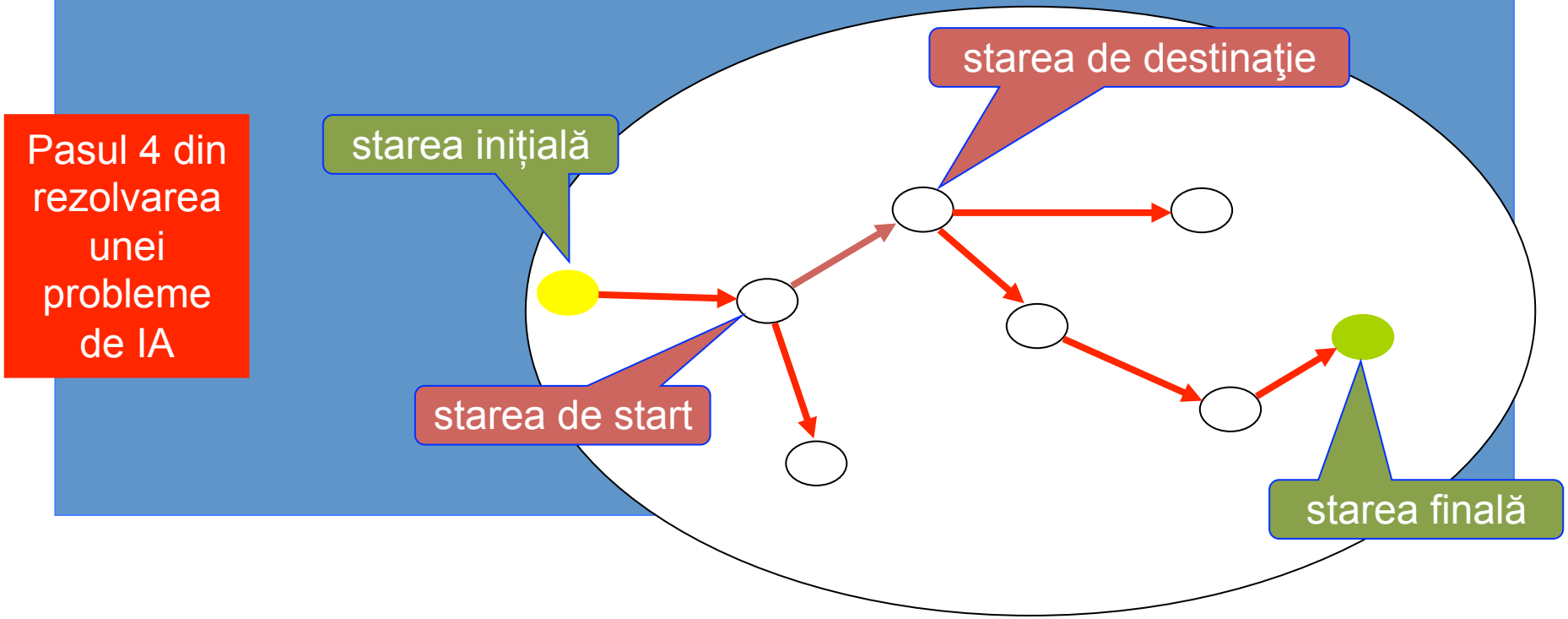
Stări, spaţiul stărilor

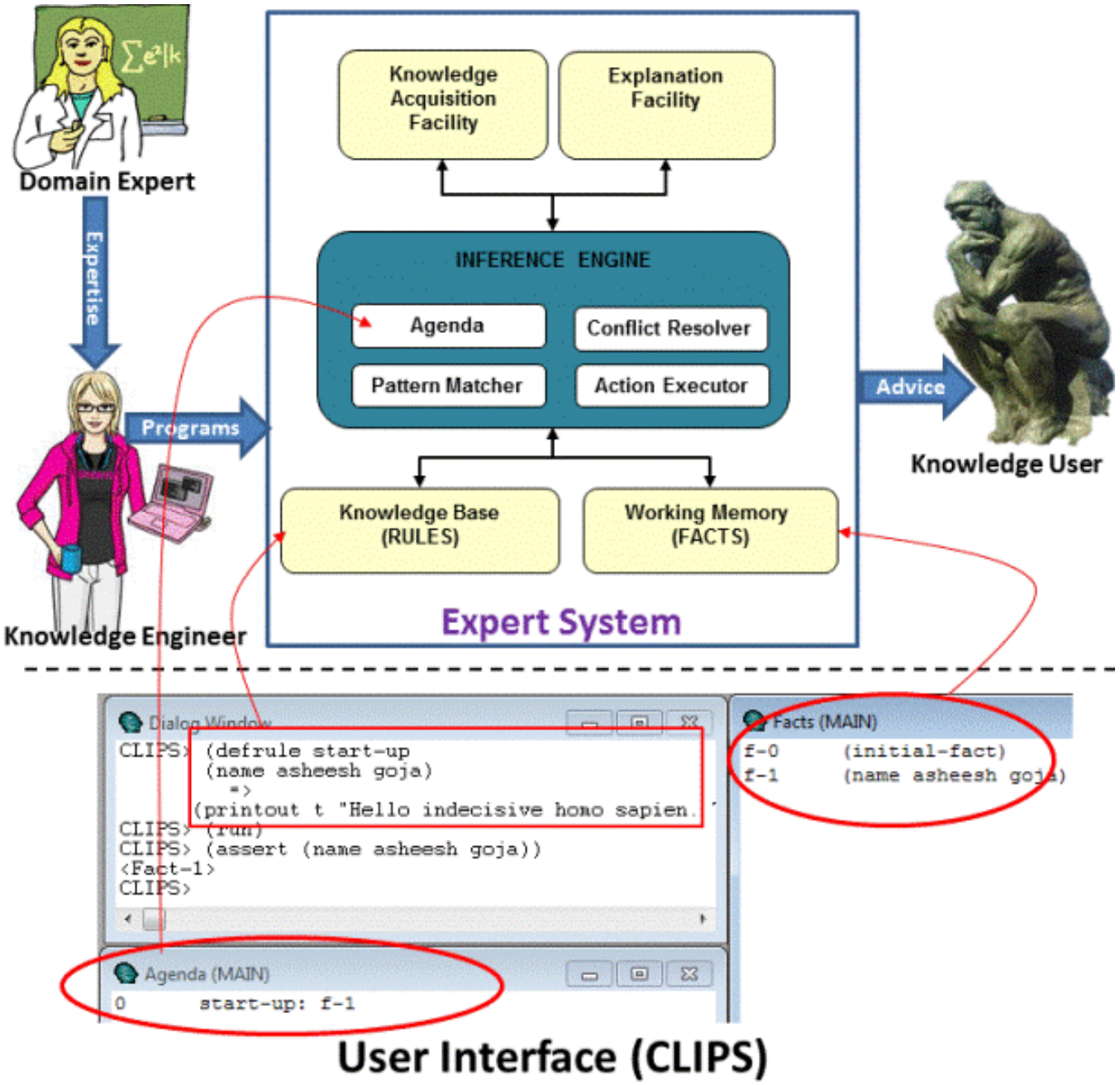


Reprezentarea tranziţiilor dintre stări

Un operator (regulă) verifică condiţii şi produce transformări în stare

```
if <condiţii> then <acţiuni>
```





User Interface (CLIPS)

Componentele motorului de SE

- Baza de cunoștințe (fapte)
- Baza de reguli
- Sistemul de control
- Agenda

Baza de cunoștințe

- Două tipuri de cunoştinţe:
  - declarații asupra structurii obiectelor (numele generice ale obiectelor, configurația de atribute şi tipurile lor de valori)
  - declarații asupra conținutului informațional al obiectelor (numele ce le individualizează, valorile atributelor) => **fapte**
    - [proprietate = valoare]
    - [proprietate = multi-valoare]

Viața cunoștințelor

- Permanente
  - declarații de tipuri
  - anumite fapte
- Volatile
  - se schimbă în timpul rulării

Reprezentarea faptelor
Baza de fapte (BF)

- Cu unicitate: nu pot exista mai multe fapte identice!
- Două implicații:
  - maniera de operare a sistemului de control: nu permite includerea simultană în bază a două fapte identice;
  - păstrarea consistenței informaționale a bazei => o responsabilitate a programatorului: baza să nu cuprindă informații contradictorii.

Exemplu

- Un număr complex: 80 + 100i
  - [80, 100]
  - [real=80, imaginar=100]

- Dacă în bază trebuie memorate două numere identice?
  - [complex=c1, real=80, imaginar=100]
  - [complex=c2, real=80, imaginar=100]

Reguli

- Codifică transformările ce trebuie operate asupra obiectelor din baza de cunoştințe

  <nume regulă>: [<comentariu>]

  dacă <condiții> atunci <acțiuni>

- Exemplu:

  diagnostic_meningită:

  DACĂ pacientul are gâtul țeapăn,

  ȘI pacientul are temperatură mare,

  SI pacientul are dese pierderi de cunoştință,

  ATUNCI pacientul e suspect de meningită

Aplicarea (“aprinderea”) regulilor

- Pentru a putea fi aplicată, în bază trebuie să existe:
  - [pacient = Ionescu, mobilitate_gât = mică]
  - [pacient = Ionescu, temperatură = mare]
  - [pacient = Ionescu, pierderi_cunoştinţă = frecvente]
- Rezultatul:
  - [pacient = Ionescu, diagnostic_posibil = meningită]

Variabile

- Apar în definițiile de reguli.
- Domeniul lexical: limitat la o regulă.
- Exemplu:

diagnostic_meningită:

dacă

[pacient = *X*, mobilitate_gât = mică]
[pacient = *X*, temperatură = mare]
[pacient = *X*, pierderi_cunoştinţă = frecvente]

atunci

introdu în bază
[pacient = *X*, diagnostic_posibil = meningită]

Șabloane, legări de variabile

- Condițiile din părțile stângi ale regulilor în care apar variabile
- La ce servesc?
  - indică un model pentru un fapt din BF care ar trebui “să se potrivească” cu şablonul;
  - dacă potrivirea reușește, provoacă “legarea” variabilelor la valori.

Exemplu

- Suma a două numere complexe (variabilele în *italic*):

regula adunare_numere_complexe:

dacă

[complex = *c1*, real = *a1*, imaginar = *b1*]

[complex = *c2*, real = *a2*, imaginar = *b2*] şi

$c1 \neq c2$

atunci

introdu în bază numărul complex [complex = *c*, real = *a1*+*a2*, imaginar = *b1*+*b2*]

Confruntarea de șabloane

- Când partea stângă a unei reguli se confruntă cu BF:
  1. nu există o combinație de fapte din bază care să verifice condițiile,
  2. există exact un mod în care faptele din bază verifică condițiile
  3. există mai multe moduri în care acestea să le verifice.

Exemplu

- Confruntarea dintre regula adunare_numere_complexe și o BC conținând numai faptele:

  [complex = z1, real = 200, imaginar = 150] şi

  [complex = z2, real = 300, imaginar = 151]

  provoacă 2 legări:

  c1 → z1, *a1* → 200, *b1* → 150, *c2* → z2, *a2* → 300, *b2* → 151

  sau

  c1 → z2, *a1* → 300, *b1* → 151, *c2* → z1, *a2* → 200, *b2* → 150

Agenda

- Structura de date care memorează la fiecare moment instanțele regulilor.
- Aceste instanțe sunt dispuse într-o listă.
- Instanța de regulă aflată pe prima poziție este cea care va fi utilizată la aprinderea regulii.

Faptele sunt însoțite de indecși în BF

- Exemplu:

  f1: [pacient = Ionescu, mobilitate_gât = mică]

  f2: [pacient = Ionescu, temperatură = mare]

  f3: [pacient = Ionescu, pierderi_cunoştință = frecvente]

Instanță a unei reguli

- Constă din:
  - numele regulii +
  - indecșii faptelor ce verifică condițiile regulii +
  - legările variabilelor la valori
- Notație:

  <nume_regulă; f1, ... fn; *v1*→a1, ... *vk*→ak>
- Exemplu:

  <diagnostic_meningită; f1, f2, f3; *X*→Ionescu>

Criterii care dictează ordinea instanțelor regulilor din agendă

- **Prioritatea declarată a regulilor**
- **Strategia de control**

  – Urmând aceste două criterii, instanțele regulilor ce-şi satisfac condițiile la un moment dat sunt întâi ordonate în agendă în ordinea descrescătoare a priorităților declarate, iar cele de priorități egale, în ordinea dată de strategia de control.

Regulile au priorități declarate

<nume regulă>: [<comentariu>]

prioritate = *p*

dacă <condiții> atunci <acțiuni>

- Exemplu:

diagnostic_meningită:

prioritate = 10

DACĂ pacientul are gâtul țeapăn,

ȘI pacientul are temperatură mare,

SI pacientul are dese pierderi de cunoştință,

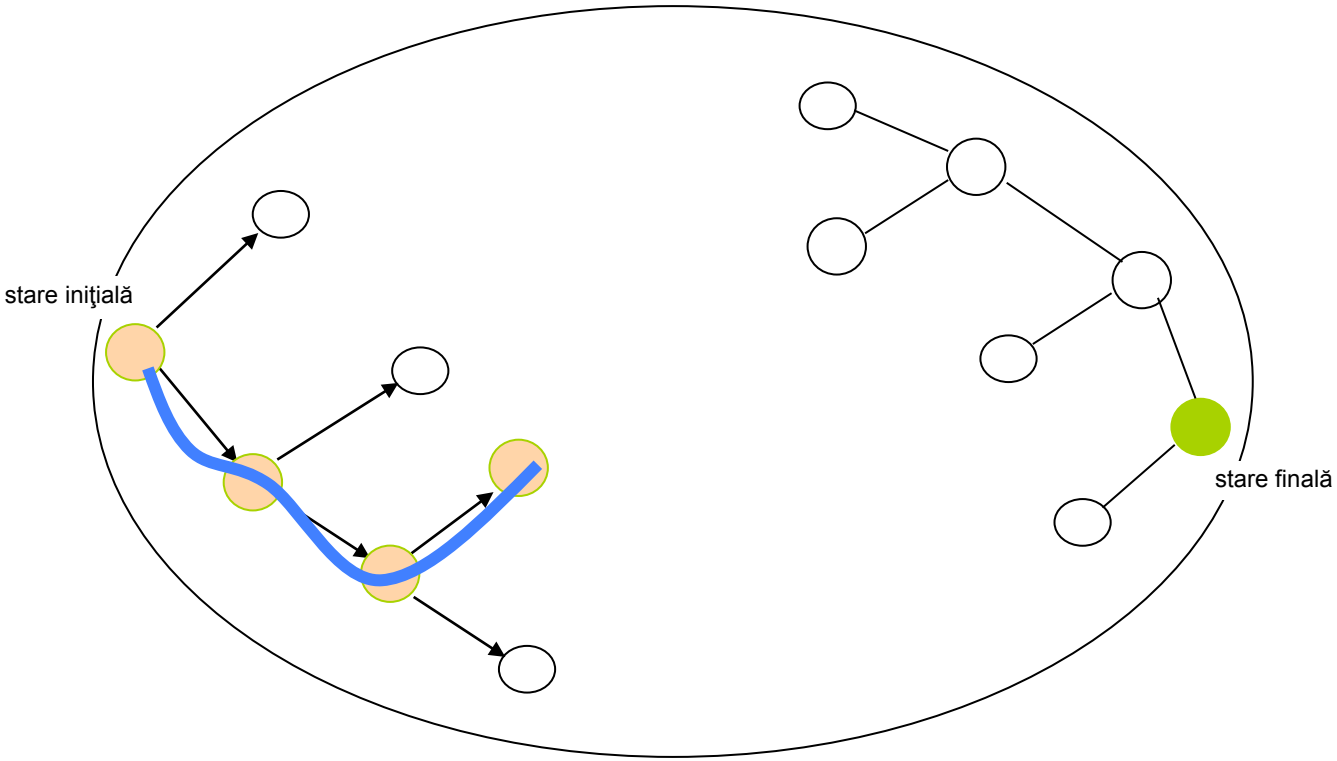
ATUNCI pacientul e suspect de meningită

Strategii de control

- Înainte
- Înapoi
- Bidirecțională

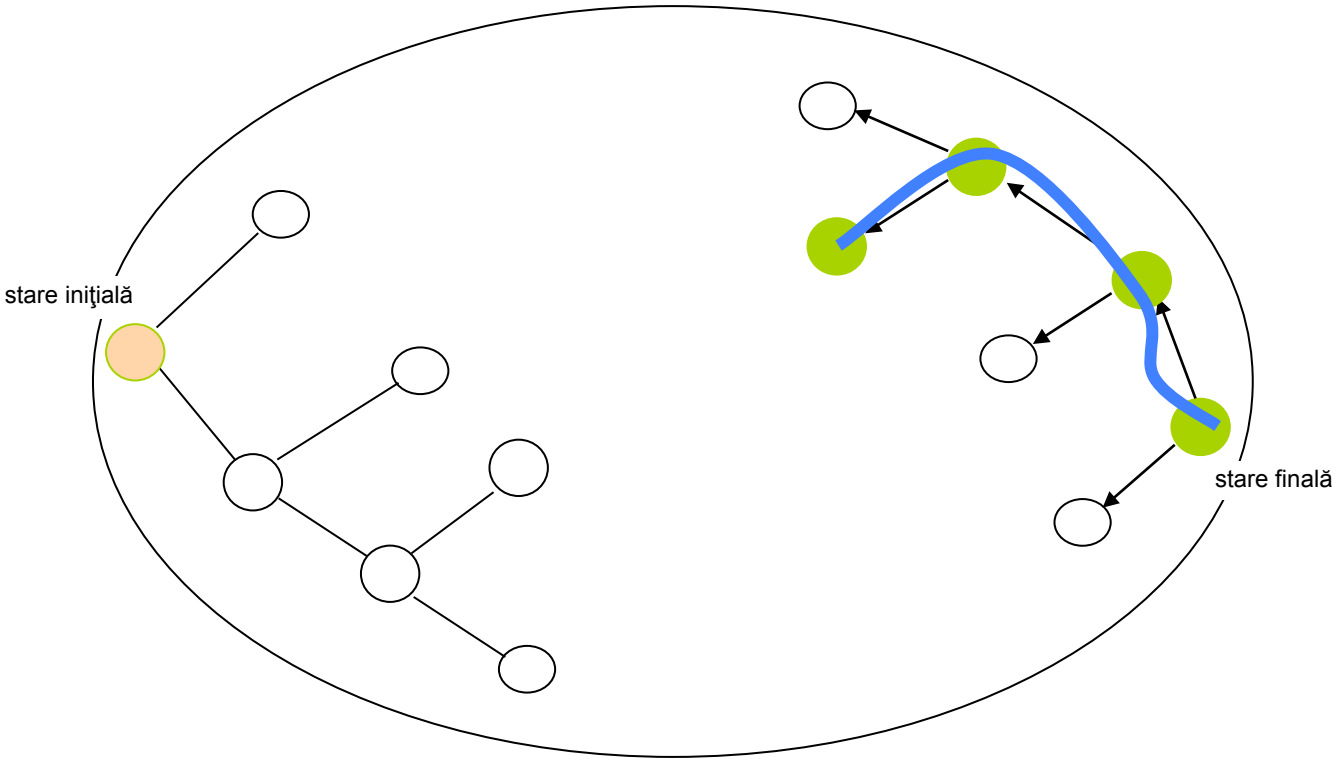
Căutare în spațiul stărilor

- Căutare înainte



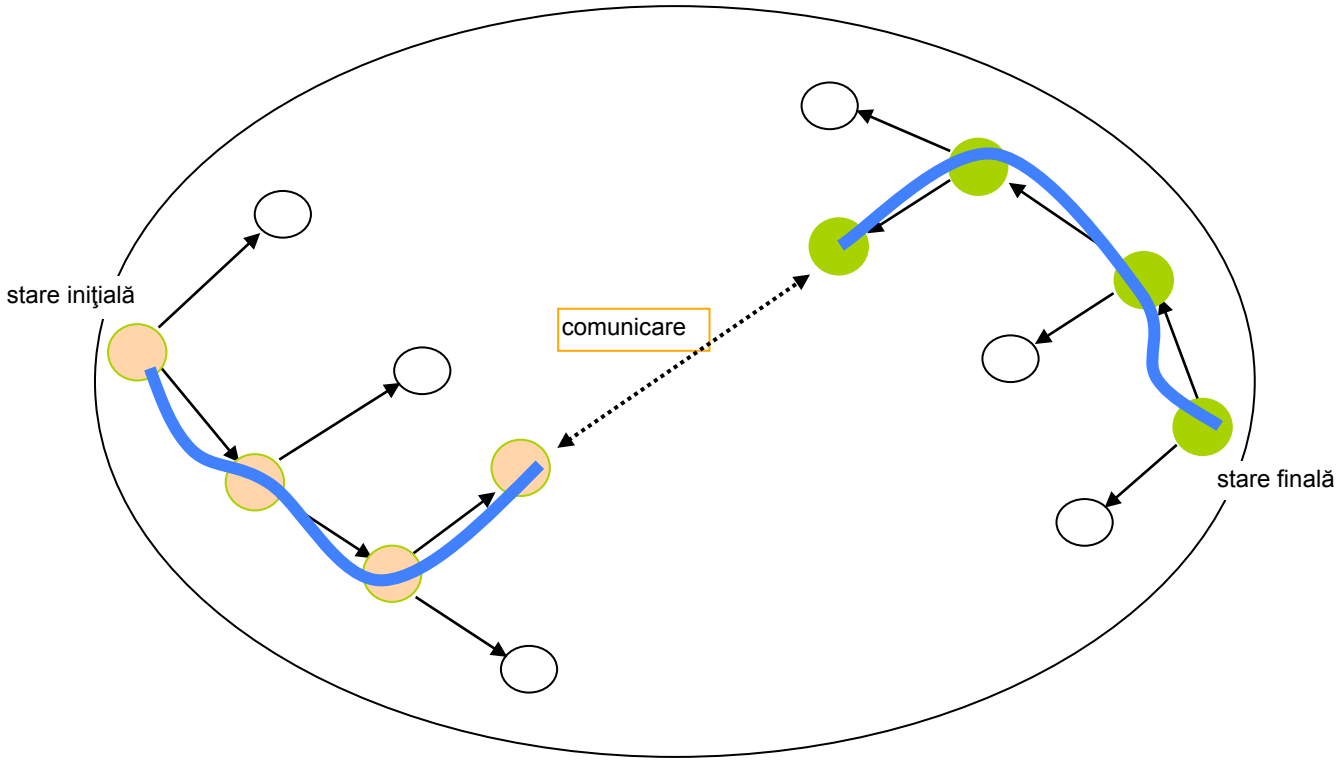
Căutare în spațiul stărilor

- Căutare înapoi

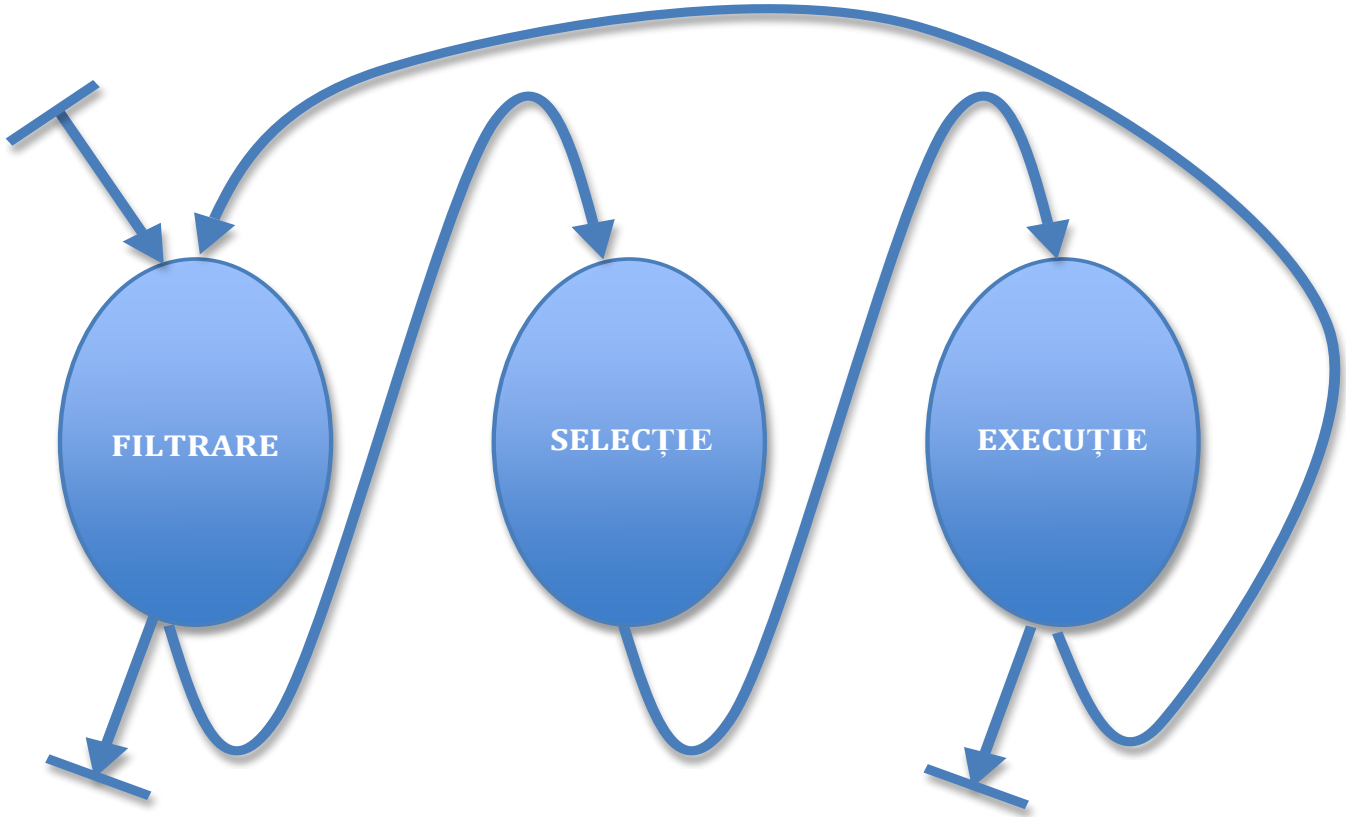


Căutare în spațiul stărilor

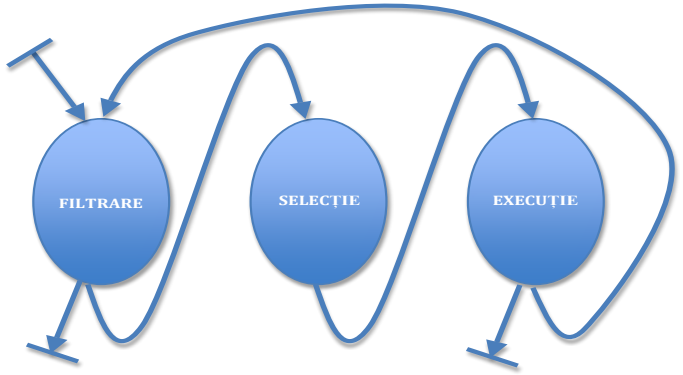
- Căutare bidirecţională sincronă



Fazele unui motor de SE cu înlănțuire înainte



Faza de Filtrare

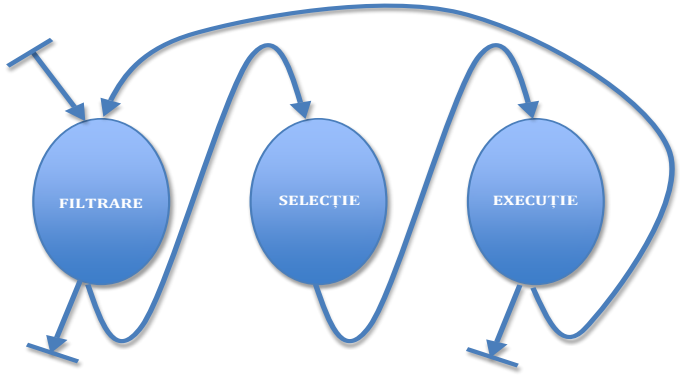


- Se determină mulțimea tuturor instanțelor de reguli filtrate (*MIRF*) corespunzătoare regulilor din baza de reguli (*BR*) care îşi pot satisface condițiile pe faptele din baza de fapte (*BF*).
- Dacă nici o regulă nu a putut fi filtrată, atunci motorul se opreşte.
- Dacă există cel puțin o instanță de regulă filtrată, atunci se trece în faza următoare.

Refractabilitate

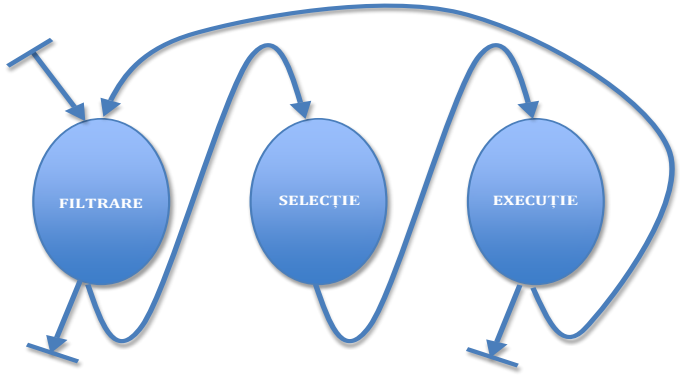
- Proprietate care se manifestă în faza de filtrare:
  - o regulă nu este filtrată mai mult de o singură dată pe un set anume de fapte.
- Fără această proprietate, sistemele expert ar fi angrenate adesea în bucle triviale ce ar apărea ori de câte ori acțiunile părții drepte ale unei reguli nu ar produce modificări în bază.

Faza de Selecție



- Se selectează o instanță de regulă $R \in MIRF$. Dacă *MIRF* conține mai mult de o singură instanță, atunci selecția se realizează prin aplicarea uneia or a mai multor **strategii de conflict**, după care se trece în faza următoare.

Faza de Execuție



- Se execută partea de acțiuni a regulii *R*, cu rezultat asupra bazei de fapte.
- Se revine în faza de filtrare, sau…
- … dacă comanda este de HALT, motorul se oprește!