



2. Cautare cu actiuni nedeterministe

Problema aspiratorului determinist

- Locatii A,B care pot fi curate (C) sau murdare (M)
- Actiuni agent: **St**, **Dr**, **Aspira**, (**nimic**)
- 2×2^2 stari posibile (2×2^n)
- $M, M, \text{Agent}^A \rightarrow_{\text{Dr}} M, M, \text{Agent}^B$
- $M, M, \text{Agent}^A \rightarrow_{\text{St}} M, M, \text{Agent}^A$
- $M, M, \text{Agent}^A \rightarrow_{\text{Aspira}} C, M, \text{Agent}^A$
- Stare initiala (M, M, Agent^A)
- Plan = [**Aspira**, **Dr**, **Aspira**]



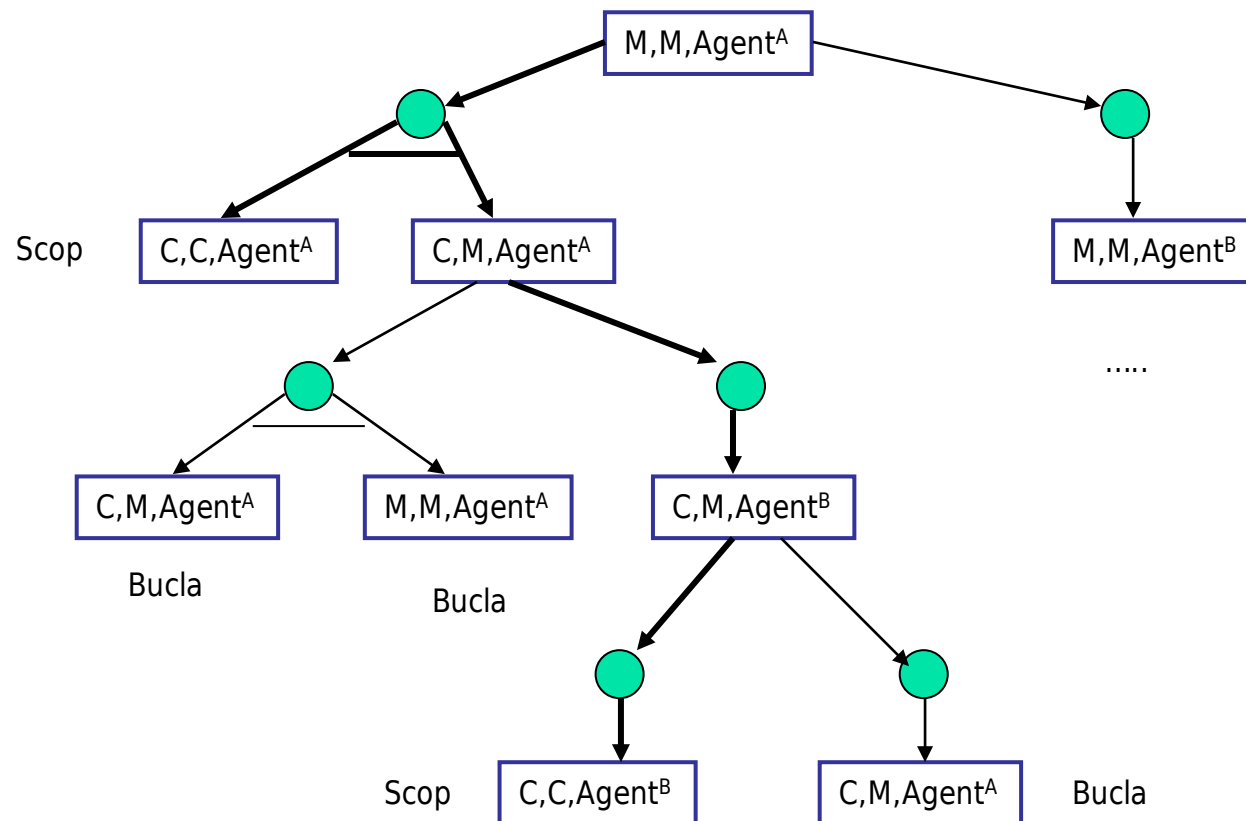
Problema aspiratorului nedeterminist

Aspira nedeterminist

- daca **Aspira** in M atunci (C) sau (C si C patrat alaturat)
- daca **Aspira** in C atunci (C) sau (M)
- Stare initiala (M,M, Agent^A)
- Plan contingent =
 [**Aspira**,
 daca Stare = (C,M, Agent^A) **atunci** **Dr, Aspira**
 altfel nimic]
- Planul – arbore SI/SAU

Problema aspiratorului nedeterminist

- Solutie – un arbore SI/SAU:
 - stare scop in fiecare frunza
 - o actiune dintr-o ramura a unui nod SAU
 - toate actiunile din ramurile unui nod SI





Plan contingent

Algoritm Plan: **Determina graf SI/SAU de actiuni**

1. Inspec-SAUS(S_i ,[])

/* intoarce plan contingent sau INSUCCES */

Inspec-SAUS(S, Cale)

1. **daca** S este stare finala

atunci intoarce Planul vid

2. **daca** $S \in \text{Cale}$ **atunci intoarce** INSUCCES

3. **pentru** fiecare actiune A_i posibil de executat din S **executa**

 3.1 Plan \leftarrow Inspec-SI(Stari(S, A_i), [S|Cale])

 3.2 **daca** Plan \neq INSUCCES **atunci intoarce** [A_i |Plan]

4. **intoarce** INSUCCES

sfarsit



Plan contingent

Inspec-SI(Stari, Cale)

1. **pentru** fiecare $S_i \in \text{Stari}$ **executa**
 - 1.1 $\text{Plan}_i \leftarrow \text{Inspec-SAU}(S_i, \text{Cale})$
 - 1.2 **daca** $\text{Plan}_i = \text{INSUCCES}$
atunci intoarce INSUCCES
2. **intoarce**
[if S_1 then Plan_1 else ...if S_{n-1} then Plan_{n-1} else Plan_n]
sfarsit