Paradigme de Programare

S.l. dr. ing. Andrei Olaru
slides: Mihnea Muraru si Andrei Olaru

Catedra de Calculatoare

2013 – 2013, semestrul 2

Cursul 11

Maşina algoritmică Markov

Cuprins

Introducere

Maşina algoritmică Markov

- Model de calculabilitate efectivă, echivalent cu Maşina Turing şi Calculul Lambda;
- Principiul de funcţionare: *pattern matching* şi substituţie;
- Fundamentul teoretic al paradigmei asociative şi al limbajelor bazate pe reguli (de forma *dacă-atunci*).

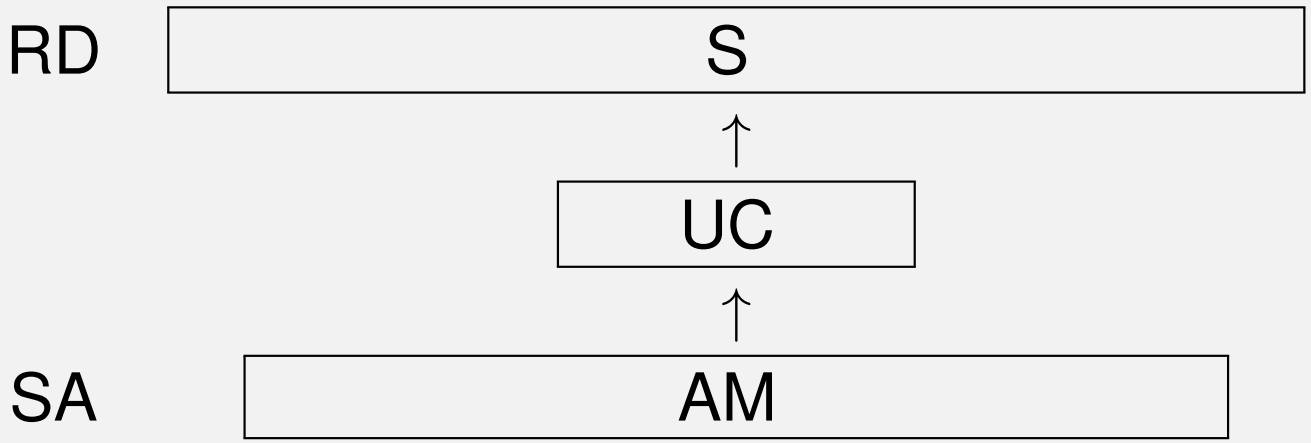
Paradigma asociativă

Caracteristici

- Potrivită mai ales în cazul problemelor ce nu admit o soluţie precisă algoritmică (ieftină);
- Codificarea cunoştinţelor specifice unui domeniu şi aplicarea lor într-o manieră euristică;
- Descrierea proprietăţilor soluţiei, prin contrast cu paşii care trebuie realizaţi pentru obţinerea acesteia (*ce* trebuie obţinut vs. *cum*);
- Absenţa unui flux explicit de control, deciziile fiind determinate, implicit, de cunoştinţele valabile la un anumit moment $\rightarrow$ *data-driven control*.

Maşina algoritmică Markov

Structură



- Registrul de date, RD, cu secvenţa de simboluri, S
- Unitatea de control, UC
- Spaţiul de stocare a algoritmului, SA, ce conţine algoritmul Markov, AM

Registrul de date

Spațiul de lucru al mașinii

- Nemărginit la dreapta
- Simboluri din alfabetul $A_b \cup A_l$:
  - A_b: alfabetul de bază
  - A_l: alfabetul local / de lucru
  - $A_b \cap A_l = \emptyset$
- Şirurile iniţial şi final formate doar cu simboluri din A_b;
- Simbolurile din A_l utilizabili exclusiv în timpul execuţiei;
- Şirul de simboluri posibil vid.

Reguli

Algoritmul după care lucrează mașina

- Unitatea de bază a unui algoritm Markov → regula asociativă de substituţie:

  `şablon identificare (LHS)` → `şablon substituţie (RHS)`

- Exemplu: `ag1c -> ac`
- Şabloanele → secvenţe de simboluri:
  - constante: simboluri din A_b
  - variabile locale: simboluri din A_l
  - variabile generice: simboluri speciale, din mulţimea G, legaţi la simboluri din A_b
- Dacă `RHS` este "." → regulă terminală, ce încheie execuţia maşinii.

Variabile generice

- De obicei, notate cu `g`, urmat de un indice;
- Mulţimea valorilor pe care le poate lua o variabilă $\rightarrow$ domeniul variabilei – `Dom(g)`;
- Legate la exact un simbol la un moment dat;
- Durata de viață $\rightarrow$ timpul aplicării regulii;
- Utilizabile în `RHS` doar în cazul apariţiei în `LHS`.

Algoritmi

Conținn programele

- Mulţimi ordonate de reguli, îmbogăţite cu declaraţii:
  - de partiţionare a mulţimii A_b
  - de variabile generice

Exemplul 40.1.

Eliminarea din mulțimea A simbolurilor ce aparţin mulţimii M:

```
1  setDiff1(A, B); A g1; B g2;
2        ag2 -> a;
3        ag1 -> g1a;
4        a -> .;
5        -> a;
6  end
```

```
1  setDiff2(A, B); B g2;
2        g2 -> ;
3        -> .;
4  end
```

- $\mathtt{A}, \mathtt{B} \subseteq A_b$
- $\mathtt{g}_1$, $\mathtt{g}_2 \rightarrow$ variabile generice
- $\mathtt{a}$ nedeclarată $\rightarrow$ variabilă locală ($\mathtt{a} \in A_l$)

Reguli

Aplicabilitate

Definiția 40.2 (Aplicabilitatea unei reguli).

Regula $r : a_1 \ldots a_n \to b_1 \ldots b_m$ este aplicabilă dacă şi numai dacă există un subşir $c_1 \ldots c_n$, în RD, astfel încât $\forall i = \overline{1, n}$ exact 1 condiţie din cele de mai jos este îndeplinită:

- $a_i \in A_b \wedge a_i = c_i$
- $a_i \in A_l \wedge a_i = c_i$
- $a_i \in G \wedge$
  $(\forall j = \overline{1, n} . a_j = a_i \Rightarrow c_j \in Dom(a_i) \wedge c_j = c_i)$,

i.e. variabila a_i este legată la o valoare unică, obţinută prin potrivirea dintre şablon şi subşir.

Definiția 40.3 (Aplicarea unei reguli).

Aplicarea regulii

$r: a_1 \ldots a_n \to b_1 \ldots b_m$ asupra unui subşir

$s: c_1 \ldots c_n$, în raport cu care este aplicabilă, constă în substituirea lui s prin subşirul $q_1 \ldots q_m$, calculat astfel:

- $b_i \in A_b \Rightarrow q_i = b_i$
- $b_i \in A_l \Rightarrow q_i = b_i$
- $b_i \in G \wedge (\exists\, j = \overline{1, n} \,.\, b_i = a_j) \Rightarrow q_i = c_j$

Exemplu de aplicare

Exemplul 40.4.

- A_b= {1, 2, 3}
- A_1= {x, y}
- Dom(g_1) = {2}
- Dom(g_2) = A_b
- S = 1111112x2y31111
- r : $1g_1xg_1yg_2 \rightarrow 1g_2x$

S = 11111 1 2 x 2 y 3 1111

r = : 1 g_1 x g_1 y g_2 $\rightarrow$ $1g_2x$

S' = 1111113x1111

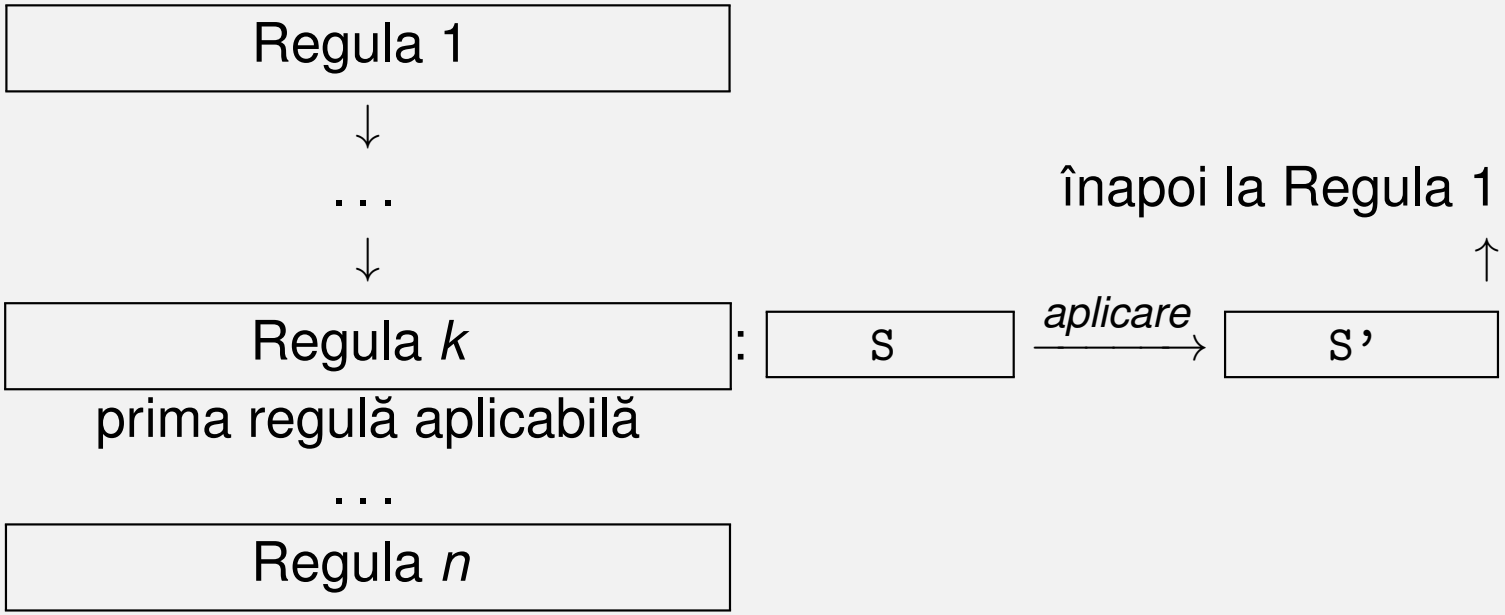
Aplicabilitate vs. aplicare

Comentarii

- Aplicabilitatea
  - unei reguli pentru mai multe subşiruri;
  - mai multor reguli pentru acelaşi subşir.
- La un anumit moment, aplicarea propriu-zisă a unei singure reguli asupra unui singur subşir;
- Nedeterminism inerent, ce trebuie exploatat, sau rezolvat
- Convenţie care poate fi făcută:
  - aplicarea primei reguli aplicabile, asupra
  - celui mai din stânga subşir asupra căreia este aplicabilă

Unitatea de control

Funcționare



Unitatea de control

Etape

- Analogia cu o sită pe mai multe nivele, ce corespund regulilor;
- Secvenţialitatea testării aplicabilităţii, nu a aplicării propriu-zise!
- Etape:
  1. determinarea primei reguli *aplicabile*
  2. aplicarea acesteia
  3. actualizarea RD
  4. salt la pasul 1

Unitatea de control

Algoritm

control(S, *Rules*)

1. $i \leftarrow 1$; $n \leftarrow |Rules|$; $status \leftarrow$ RUNNING
2. **while** $i \leq n$ **and** $status =$ RUNNING
3. $\quad r \leftarrow Rules[i]$
4. $\quad$ **if** *isApplicable*(S, r) **then**
5. $\quad\quad S \leftarrow fire(S, r)$
6. $\quad\quad$ **if** *isTerminal*(r) **then**
7. $\quad\quad\quad status \leftarrow$ TERMINATED
8. $\quad\quad$ **else**
9. $\quad\quad\quad i \leftarrow 1$
10. $\quad$ **else**
11. $\quad\quad i \leftarrow i + 1$
12. **if** $status =$ TERMINATED **then**
13. $\quad$ **return** S
14. **else** *error*(“Execution blocked”)

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2        ag1g2 -> g2ag1;
3        ag1 -> bg1;
4        abg1 -> g1a;
5        a -> .;
6        -> a;
7  end
```

- DOP

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP}$

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP} \xrightarrow{2} \texttt{OaDP}$

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP} \xrightarrow{2} \texttt{OaDP} \xrightarrow{2} \texttt{OPaD}$

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP} \xrightarrow{2} \texttt{OaDP} \xrightarrow{2} \texttt{OPaD} \xrightarrow{3} \texttt{OPbD}$

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP} \xrightarrow{2} \texttt{OaDP} \xrightarrow{2} \texttt{OPaD} \xrightarrow{3} \texttt{OPbD} \xrightarrow{6} \texttt{aOPbD}$

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP} \xrightarrow{2} \texttt{OaDP} \xrightarrow{2} \texttt{OPaD} \xrightarrow{3} \texttt{OPbD} \xrightarrow{6} \texttt{aOPbD} \xrightarrow{2} \texttt{PaObD}$

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP} \xrightarrow{2} \texttt{OaDP} \xrightarrow{2} \texttt{OPaD} \xrightarrow{3} \texttt{OPbD} \xrightarrow{6} \texttt{aOPbD} \xrightarrow{2} \texttt{PaObD} \xrightarrow{3} \texttt{PbObD}$

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP} \xrightarrow{2} \texttt{OaDP} \xrightarrow{2} \texttt{OPaD} \xrightarrow{3} \texttt{OPbD} \xrightarrow{6} \texttt{aOPbD} \xrightarrow{2} \texttt{PaObD} \xrightarrow{3} \texttt{PbObD} \xrightarrow{6} \texttt{aPbObD}$

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP} \xrightarrow{2} \texttt{OaDP} \xrightarrow{2} \texttt{OPaD} \xrightarrow{3} \texttt{OPbD} \xrightarrow{6} \texttt{aOPbD}$
  $\xrightarrow{2} \texttt{PaObD} \xrightarrow{3} \texttt{PbObD} \xrightarrow{6} \texttt{aPbObD} \xrightarrow{3} \texttt{bPbObD}$

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP} \xrightarrow{2} \texttt{OaDP} \xrightarrow{2} \texttt{OPaD} \xrightarrow{3} \texttt{OPbD} \xrightarrow{6} \texttt{aOPbD}$
  $\xrightarrow{2} \texttt{PaObD} \xrightarrow{3} \texttt{PbObD} \xrightarrow{6} \texttt{aPbObD} \xrightarrow{3} \texttt{bPbObD} \xrightarrow{6} \texttt{abPbObD}$

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP} \xrightarrow{2} \texttt{OaDP} \xrightarrow{2} \texttt{OPaD} \xrightarrow{3} \texttt{OPbD} \xrightarrow{6} \texttt{aOPbD} \xrightarrow{2} \texttt{PaObD} \xrightarrow{3} \texttt{PbObD} \xrightarrow{6} \texttt{aPbObD} \xrightarrow{3} \texttt{bPbObD} \xrightarrow{6} \texttt{abPbObD} \xrightarrow{4} \texttt{PabObD}$

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP} \xrightarrow{2} \texttt{OaDP} \xrightarrow{2} \texttt{OPaD} \xrightarrow{3} \texttt{OPbD} \xrightarrow{6} \texttt{aOPbD}$
  $\xrightarrow{2} \texttt{PaObD} \xrightarrow{3} \texttt{PbObD} \xrightarrow{6} \texttt{aPbObD} \xrightarrow{3} \texttt{bPbObD} \xrightarrow{6} \texttt{abPbObD}$
  $\xrightarrow{4} \texttt{PabObD} \xrightarrow{4} \texttt{POabD}$

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP} \xrightarrow{2} \texttt{OaDP} \xrightarrow{2} \texttt{OPaD} \xrightarrow{3} \texttt{OPbD} \xrightarrow{6} \texttt{aOPbD}$
  $\xrightarrow{2} \texttt{PaObD} \xrightarrow{3} \texttt{PbObD} \xrightarrow{6} \texttt{aPbObD} \xrightarrow{3} \texttt{bPbObD} \xrightarrow{6} \texttt{abPbObD}$
  $\xrightarrow{4} \texttt{PabObD} \xrightarrow{4} \texttt{POabD} \xrightarrow{4} \texttt{PODa}$

Un exemplu

Inversarea intrării

- Ideea: mutarea, pe rând, a fiecărui element, în poziţia corespunzătoare, prin interschimbarea elementelor adiacente

```
1  Reverse(A); A g1, g2;
2       ag1g2 -> g2ag1;
3       ag1 -> bg1;
4       abg1 -> g1a;
5       a -> .;
6       -> a;
7  end
```

- $\texttt{DOP} \xrightarrow{6} \texttt{aDOP} \xrightarrow{2} \texttt{OaDP} \xrightarrow{2} \texttt{OPaD} \xrightarrow{3} \texttt{OPbD} \xrightarrow{6} \texttt{aOPbD}$
  $\xrightarrow{2} \texttt{PaObD} \xrightarrow{3} \texttt{PbObD} \xrightarrow{6} \texttt{aPbObD} \xrightarrow{3} \texttt{bPbObD} \xrightarrow{6} \texttt{abPbObD}$
  $\xrightarrow{4} \texttt{PabObD} \xrightarrow{4} \texttt{POabD} \xrightarrow{4} \texttt{PODa} \xrightarrow{5} .$

Sfârșitul cursului 11

Ce am învățat

· Ce este și cum funcționează mașina algoritmică Markov: structură, variabile, reguli, algoritmul unității de control.