ALGORITMI DE GENERARE A MULŢIMII FUNCŢIILOR SURJECTIVE CU DOMENIU ŞI CODOMENIU FINIT

mat. Ion Florea

Universitatea Transilvania, Braşov

Rezumat:

Lucrarea prezintă o metodă de determinare a mulţimii funcţiilor surjective cu domeniu şi codomeniu mulţimi finite, bazată pe metoda generală de elaborare a algoritmilor Backtraking.

Cuvinte cheie: algoritm, corectitudine algoritm, echivalenţă, mulţime.

1. Introducere

Problema găsirii unor algoritmi de generare a unor mulţimi remarcabile constituie un subiect abordat în literatura de specialitate. În [1] sunt prezentaţi o serie întreagă de asemenea algoritmi, printre care şi algoritmi de generare a produsului cartezian a unui număr finit de mulţimi, a mulţimii aranjamentelor şi a mulţimii permutărilor, deci [2] algoritmi de generare a mulţimii tuturor funcţiilor, respectiv a mulţimii funcţiilor injective, a mulţimii funcţiilor bijective, evident cu domeniu şi codomeniu mulţimi finite.

In continuare, vom prezenta un algoritm de generare a mulţimii tuturor funcţiilor surjective cu domeniu şi codomeniu mulţimi finite, precum şi consideraţiile matematice care il fundamentează.

2. Echivalenţa între mulţimea funcţiilor surjective cu domeniu şi codomeniu mulţimi finite şi o mulţime de matrici

Definiţie. 1 *Fie $m, n \in Z$: $m, n \geq 1$: $M = 1, ..., m$, $N = 1, ..., n$; S_{mn} mulţimea tuturor funcţiilor surjective, definite pe M, cu valori în N: SR_{mn} mulţimea de matrici caracterizată de următoarele proprietăţi:*

i. oricare ar fi $S = (s_{ij})_{i \in M, j \in N}: s_{ij} \in 0, 1$;

ii. oricare ar fi $i \in M$ există un unic $j \in N$ astfel încât $s_{ij} = 1$;

iii. oricare ar fi $j \in N$ există $i \in M$ astfel încât $s_{ij} = 1$.

Observaţie. Fie $f : M \to N$ surjectivă. Atunci $m \geq n$. Proprietatea este cunoscută şi uşor de demonstrat.

Teorema. 1 *Mulţimile S_{mn} şi SR_{mn} sunt cardinal echivalente.*

Demonstraţie. Fie $h : S_{mn} \to SR_{mn}$ definită astfel: $h(f) = S$; $S = (s_{ij})_{i \in M, j \in N}$: $s_{ij} = 1$ dacă $f(i) = j, = 0$ altfel.

Funcţia h este corect definită.
Fie $f \in S_{mn}$: definim $S = (s_{ij})_{i \in M, j \in N}$ astfel: oricare ar fi $i \in M, j \in N, s_{ij} = 1$, dacă $f(i) = j, 0$ altfel. Evident $s_{ij} \in 0, 1$ deci S verifică i). Deoarece f este funcţie pentru fiecare $i \in M$ există un unic $j \in N$ astfel incat $f(i) = j$, adică oricare ar fi $i \in M$ există un unic $j \in N$ astfel incat $s_{ij} = 1$, deci S verifică ii). Deoarece f este funcţie surjectivă, oricare ar fi $j \in N$ există $i \in M$ astfel incat $f(i) = j$, adică oricare ar fi $j \in N$ există $i \in M$ astfel incat $s_{ij} = 1$, deci S verifică iii). Pentru a incheia, observăm că unei funcţii nu-i poate corespunde decat o singură matrice, ceea ce este evident din modul cum am definit matricea corespunzatoare funcţiei.

Funcţia h este injectivă.
Fie $f, g \in S_{mn}$, f diferită de g; există $j, k \in N$, j diferit de k astfel incat $j = f(i)$ şi $k = g(i)$. Fie $S = u(f)$ şi $T = g(f)$, $S = (s_{ij})_{i \in M, j \in N}$, $T = (t_{ij})_{i \in M, j \in N}$: din definiţia mulţimii SR_{mn} (vezi ii)) rezultă $s_{ij} = 1$ şi $t_{ij} = 0$, respectiv $s_{ik} = 0$ şi $t_{ik} = 1$, deci S este diferită de T.

Funcţia h este surjectivă.
Fie $S \in S_{mn}$: din definiţie (vezi ii)) rezultă că oricare ar fi $i \in M$ există un unic $j \in N$ astfel incat $s_{ij} = 1$. Definim $f(i) = j$. Evident f este corect definită şi $h(f) = S$.

3. Algoritm de generare a mulţimii matricilor, echivalentă cu mulţimea funcţiilor surjective

Din teorema 1 rezultă că a genera mulţimea funcţiilor surjective definite pe M cu valori in N este echivalent cu a genera mulţimea de matrici SR_{mn}. Pentru aceasta vom folosi metoda generală Backtracking (vezi [3]).

Spaţiul soluţiilor este mulţimea A, definită ca produsul cartezian al mulţimilor A_1, ..., A_m; $A_1 = \ldots = A_m$; oricare ar fi $i \in M$ mulţimea A_i se defineşte astfel:

$A_i = \{(a_1, \ldots, a_n)/a_t \in \{0,1\}$, oricare ar fi $t \in N$; există un unic $t \in N$ astfel încât $a_t = 1\}$.

Elementul iniţial pentru fiecare mulţime $A_i, i \in M$, este vectorul nul $(0, \ldots 0)$ cu n componente.

Succesorul unui element din mulţimea $A_i, i \in M$ se defineşte astfel:

- elementul $(0, \ldots, 0, 1)$ nu are succesor;
- succesorul elementului $(a_1, \ldots, a_n)$ astfel încât $a_k = 1, k = 1, \ldots, n-1$ este elementul $(a_1, \ldots, a_n)$ pentru care $a_{k+1} = 1$.

Condiţiile de continuare se exprimă pornind de la observaţia ca oricare matrice din SR_{mn}, generată linie cu linie, nu are nici o coloană nulă; la fiecare pas $k, k = 1, \ldots, m$, când se generează o linie condiţiile de continuare se exprimă astfel: numărul coloanelor nule din matricea formată de linia de la pasul k cu liniile deja generate la paşii anteriori trebuie să fie cel mult $m - k$.

Algoritmul este descris de programul de mai jos scris în limbajul TurboPascal, care, totodată afişează pentru fiecare matrice generată funcţia surjectivă corespunzatoare.

```
program surjective;
uses crt;
const dim_max = 20;
type mat=array[1 ...dim_max, 1 ...dim_max]
of 0 ...1;
var k, nrf, m, n:integer;
    as, ev:boolean;
    s:mat;
procedure init(k, n:integer; var s:mat);
```

Iniţializează cu vectorul nul linia k a matricii date

```
var i:integer;
begin
for i:=1 to n do
s[k,i]:=0;
end;
procedure succesor (var as:boolean; var
s:mat; k,n:integer);
```

Procedura genereazā succesorul liniei k, dacă există, caz în care variabila as ("are succesor") primeşte valoarea true

```
var i:integer;
begin
i:=1;
while (s[k,i]=0) and (i<n) do
    i:=i+1;
if i<n then
    begin
    as:=true;
    s[k,i]:=0;
    s[k,i+1]:=1;
    end
    else
    if s[k,n]=1 then as:=false
          else
          begin
          as:=true;
          s[k,1]:=1;
          end
end;
procedure valid (var ev:boolean; s:mat;
m,k,n:integer);
```

Procedura verifică dacă linia k generată verifică condiţiile de continuare, caz în care variabila ev (" E valid") primeşte valoarea true. În cn determinăm nr. coloanelor nule din cele k linii generate

```
var i,j,sc,cn:integer;
begin
cn:=0;
for j:=1 to n do
    begin
    sc:=0;
    for i:=1 to k do
       sc:=sc+s[i,j];
    if sc=0 then cn:=cn+1;
    end;
if cn <= m - k then ev:=true
    else ev:=false;
end;
function soluţie (m,k:integer):boolean;
```

După ce am generat şi coloana m, am generat o soluţie, funcţia returnând valoarea true dacă k a primit valoarea m

```
begin
soluţie:=(k=m)
end;
procedure tipar (m,n:integer);
```

Afişează funcţia corespunzatoare matricii generate

```
var i,j:integer;
begin
for i:=1 to m do
    for j:=1 to n do
    if s[i,j]=1 then write ('f(', i, ')=
', j, ' ');
```

```
   end;
```

Programul principal

```
begin
clrscr;
repeat
writeln( " nr.el.ale domeniului:  ");
read(m);
writeln( " nr.el.ale codomeniului:  ");
read(n);
until m=n;
writeln(" FUNCTIILE SURJECTIVE SUNT ");
k:=1;
init(1,n,s);
nrf:=0;
while k>0 do
begin
   repeat
   succesor (as,s,k,n);
   if as then valid (ev,s,m,k,n);
   until (not as) or (as and ev);
   if as then
       if soluţie (m,k) then
              begin
              nrf:=nrf+1;
              writeln;
              write (" Funcţia nr:
', nrf, ' ");
              tipar (m,n);
              if nrf mod 24=0 then
                  delay (10000);
                  end
              else
              begin
              k:=k+1;
              init(k,n,s)
              end
       else
       k:=k-1;
end;
repeat until keypressed;
end
```

4. Corectitudinea algoritmului

Lema. 1 *Fie $A = (a_{ij})_{i\in M, j\in N}$ astfel încât oricare ar fi $i \in M$ există un unic $j \in N$ pentru care $a_{ij} = 1$. Atunci numărul coloanelor diferite de vectorul nul.este mai mic sau egal cu numărul liniilor.*

Demonstratie. Vom demonstra afirmaţia din enunţ prin inducţie după m. Pentru $m = 1$ afirmaţia este evidentă. Presupunem afirmaţia adevarată pentru o matrice cu m linii şi o demonstrăm pentru o matrice cu $m+1$ linii. Considerăm matricea formată cu primele m linii; conform ipotezei de inducţie, numărul coloanelor nenule este k, cu $k \leq m$. Fie $j_1, \ldots, j_k$ indicii coloanelor nenule şi fie j astfel încât $a_{m+1,j} = 1$. Dacă j este diferit de j_t, pentru $t = 1, \ldots, k$, atunci numărul coloanelor nenule este $k+1$ şi $k+1 \leq m+1$, altfel numărul coloanelor nenule rămâne tot k, iar $k \leq m < m+1$, ceea ce termină demonstraţia.

Teorema. 2 *Sunt adevărate următoarele afirmaţii:*

i. Algoritmul generează toate funcţiile surjective.

ii. Algoritmul nu generează decât funcţii surjective.

iii. Orice funcţie surjectivă este generată o singură dată.

Demonstratie. i) Presupunem prin absurd că există $S = (s_{ij})_{i\in M, j\in N}, S \in S_{mn}$ care nu este generată prin algoritmul prezentat. Atunci există $k; k = 1, \ldots, m$; astfel la pasul k să nu fie îndeplinite condiţiile de continuare pentru matricea considerată, deci numărul coloanelor nule din matricea formată cu primele k linii ale matricii date este mai mare strict decât $m - k$, adică numărul coloanelor nenule este mai mic strict decat $n - m + k$.

Conform lemei, în matricea formată cu liniile $k+1, \ldots, m$ ale matricii considerate, există cel mult $m - k$ coloane nenule, deci în matricea S numărul coloanelor nenule este mai mic strict decat $n - m + k + m - k$, egal cu n, deci în matricea S există cel puţin o coloană nulă, ceea ce constituie o contradicţie.

ii) Fie $n_k; k = 1, \ldots, m$; numărul coloanelor nule din matricea formată din primele k linii generate. Conform condiţiilor' de continuare $n_k \leq m - k$, deci pentru $k = m$, cand se generează o soluţie, $n_m \leq m - m$, deci $n_m = 0$, deci $S \in S_{mn}$.

iii) Faptul că orice funcţie surjectivă este generată o singură dată rezultă din insuşi conţinutul metodei Backtracking (vezi [3]).

Bibliografie

1. LIVOVSCHI, L., GEORGESCU, H.: Sinteza şi analiza algoritmilor, Editura Ştiinţifică şi Enciclopedică, Bucureşti, 1981.
2. TOMESCU, I.: Introducere in combinatorică, Editura Tehnică, Bucureşti, 1972.
3. HOROWITZ, E., SAHNI, S.: Foundamentals of Computer Algorithms, Academic Press, N.Y., 1978.