Metoda Backtracking

*Acest **curs** prezinta **Metoda Backtracking**.*
In acest PDF poti vizualiza cuprinsul si bibliografia (daca sunt disponibile) si aproximativ doua pagini din documentul original.
*Arhiva completa de pe site contine un fisier, intr-un numar total de **7 pagini**.*
Fisierele documentului original au urmatoarele extensii: doc.

Extras

Prezentarea tehnicii Backtracking

Aceasta tehnica se foloseste în rezolvarea problemelor care îndeplinesc simultan urmatoarele conditii:

- solutia lor poate fi pusa sub forma unui vector S=x1,x2, ...,xn, cu x1 € A1,

x2 € A2 ...,xn € An

- multimile A1, A2 , ..., An sunt multimi finite, iar elementele lor se considera ca se afla într-o relatie de ordine bine stabilita;

- nu se dispune de o alta metoda de rezolvare, mai rapida

- x1 x2 ..., xn pot fi la rândul lor vectori;

- A1, A2 ..., An pot coincide.

La întâlnirea unei astfel de probleme, daca nu cunoastem aceasta tehnica, suntem tentati sa generam toate elementele produsului cartezian A1,A2 ...,An si fiecare element sa fie testat daca este solutie. Rezolvând problema în acest mod, timpul de executie este atât de mare, încât poate fi considerat infinit, algoritmul neavând nici o valoare practica.

De exemplu, daca dorim sa generam toate permutarile unei multimi finite A, nu are rost sa generam produsul cartezian AxAx.....xA, pentru ca apoi, sa testam, pentru fiecare element al acestuia, daca este sau nu permutare (nu are rost sa generam 1.1,1.....1, pentru ca apoi sa constatam ca nu am obtinut o permutare, când de la a doua cifra 1 ne puteam da seama ca cifrele nu sunt distincte).

Tehnica Backtracking are la baza un principiu extrem de simplu:

- se construieste solutia pas cu pas: x1, x2 ...,xn

- daca se constata ca, pentru o valoare aleasa, nu avem cum sa ajungem la solutie, se renunta la acea valoare si se reia cautarea din punctul în care am ramas.

Concret:

- se alege primul element x, ce apartine lui A;

- presupunând generate elementele x1,x2 ...,xk , apartinând multimilor A1,

A2 ...,Ak , se alege (daca exista) xk+1 , primul element disponibil din multimea Ak+1 , apar doua posibilitati :

1) Nu s-a gasit un astfel de element, caz în care caz în care se reia cautarea considerând generate elementele x1,x2 ...,xk+1 , iar aceasta se reia de la urmatorul element al multimii Ak ramas netestat;

2) A fost gasit, caz în care se testeaza daca acesta îndeplineste anumite conditii de continuare aparând astfel doua posibilitati:

- îndeplineste, caz în care se testeaza daca s-a ajuns la solutie si apar din nou doua posibilitati

- s-a ajuns la solutie, se tipareste solutia si se reia algoritmul considerând generate elementele x1,x2 ...,xk , (se cauta în continuare, un alt element al multimii Ak , ramas netestat);

- nu s-a ajuns la solutie, caz în care se reia algoritmul considerând generate elementele x1,x2 ...,xk , si se cauta un prim element xk+2 € Ak.

- nu le îndeplineste caz în care se reia algoritmul considerând generate elementele x1,x2 ...,xk , iar elementul xk-1 se cauta între elementele multimii A, ramase netestate.

Algoritmii se termina atunci când nu exista nici un element x1 € A1 netestat.

Observatie: tehnica Backtracking are ca rezultat obtinerea tuturor solutiilor problemei. În cazul în care se cere o sigura solutie se poate forta oprirea, atunci când a fost gasita.

Am aratat ca orice solutie se genereaza sub forma de vector. Vom considera ca generarea solutiilor se face intr-o stiva. Astfel, x1 € A1, se va gasi pe primul nivel al stivei, x2 € A2 se va gasi pe al doilea nivel al stivei,... xk € Ak se va gasi pe nivelul k al stivei. În acest fel, stiva (notata ST) va arata astfel:

ST

Nivelul k+1 al stivei trebuie initializat (pentru a alege, în ordine, elementele multimii k+1). Initializarea trebuie facuta cu o valoare aflata (în relatia de ordine considerata, pentru multimea Ak+1) înaintea tuturor valorilor posibile din multime. De exemplu, pentru generarea permutarilor multimii {1,2.....n}, orice nivel al stivei va lua valori de la 1 la n. Initializarea unui nivel (oarecare) se face cu valoarea 0. Procedura de initializare o vom numi INIT si va avea doi parametri: k (nivelul care trebuie initializat si ST (stiva)).

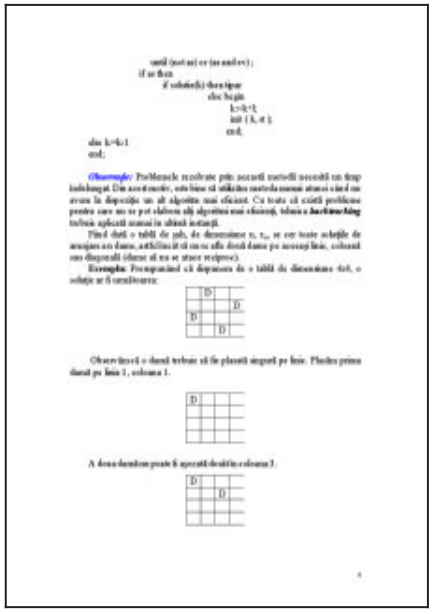
Gasirea urmatorului element al multimii Ak (element care a fost netestat) se face cu ajutorul procedurii SUCCESOR (AS,ST,K). Parametrul AS (am succesor) este o variabila booleana. În situatia în care am gasit elementul, acesta este pus în stiva si AS ia valoarea TRUE, contrar (nu a ramas un element netestat) AS ia valoarea FALSE..

Odata ales un element, trebuie vazut daca acesta îndeplineste conditiile de continuare (altfel spus, daca elementul este valid). Acest test se face cu ajutorul procedurii VALID (EV,ST,K).

Testul daca s-a ajuns sau nu la solutia finala se face cu ajutorul functiei

. . . . . .
. . . . . .
. . . . . .

Imagini din documentul complet:



Mai multe detalii se gasesc in [pagina documentului](#) din [Biblioteca.RegieLive.ro](#)