PROGRAMAREA ASISTATĂ DE CALCULATOR A TRASEULUI UNEI NAVE DE TRANSPORT MARITIM

dr. ing. Gheorghe Bordea — *Institutul de Marină Civilă Constanța*

conf. Dr. Alexandru Tertisco — *Universitatea* Politehnica *București*

ing. Alexandru Boicea — *Universitatea* Politehnica *București*

Rezumat. În lucrare, se consideră o rețea de transport, exprimată printr-un graf orientat G (N, A) în care mulțimea nodurilor $N = \{1, 2, ..., n\}$ reprezintă porturile, iar multimea arcelor A, dată de relația definită pe produsul cartezian $A \subseteq N \bullet N$. Aceasta exprimă căile de comunicație, existente între porțiunile cuprinse în rețea. Pe rețeaua de transport G pot fi definite trasee închise, care reprezintă cicluri elementare de deplasare a navei dintr-un port oarecare i cu revenire în acelasi port, după ce a tranzitat alte porturi $j \neq i$, $i, j \in N$. Mulțimea tuturor ciclurilor elementare $B = \{B1 ... Bk\}$ este caracteristică rețelei date G. Pot fi determinate traseele închise mai complicate, care sunt numite cicluri neelementare. Acestea sunt construite ca o combinație liniară de cicluri elementare. În lucrare, se propune o metodă grafo-analitică și este elaborat algoritmul determinării pe calculator a tuturor ciclurilor elementare, care aparțin rețelei și, pe aceasta bază, a traseului neelementar, caracterizat de tranzitarea anumitor porturi de mai multe ori în cadrul aceluiași ciclu (care reprezintă, de fapt, traseul dorit).

Cuvinte cheie: sisteme de evenimente discrete, rețele de transport, proiectare asistată de calculator.

1. Determinarea ciclurilor elementare ale unei rețele de transport maritim

Porturile din rețeaua de transport sunt asociate stărilor pe care le poate tranzita nava sau în care se poate afla la începutul ori la sfârșitul traseului parcurs.

În aceasta sectiune, se prezintă o metodă grafică, de determinare a ciclurilor elementare, și se întocmește algoritmul corespunzător metodei prezentate. Ciclurile elementare sunt trasee închise, în cadrul cărora nava tranzitează oricare port de pe traseu, o singură dată.

Metoda se aplică pentru un sistem cu evenimente discrete, dat prin relația stărilor *R*, și constă în eliminarea succesivă a nodurilor grafului, asociate porturilor din rețeaua de transport.

Atât metoda prezentată, cât și algoritmul, se aplică la determinarea ciclurilor elementare în cazul unui exemplu ce constă dintr-un sistem cu evenimente discrete, pentru care se cunoaște relația stărilor, exprimate prin graful asociat rețelei de transport.

La baza metodei grafice de determinare a ciclurilor elementare stau următoarele reguli și observații:

- într-un graf, două noduri pot fi legate: printr-un arc, când acestea sunt în relație directă, sau printr-un lanț, când nodurile respective sunt în relație indirectă, prin tranzitivitate;
- dacă nodurilor grafului li se atașează numere întregi, prin eliminarea unui nod se poate obține un subgraf în care lanțurile formate din două arce, ce realizau legăturile dintre celelalte noduri și treceau prin nodul desființat, se pot echivala cu arce cărora li se atașează mulțimi formate din numerele atașate nodurilor grafului prin care trecea inițial lanțul echivalat;
- un lanț format dintr-un arc ce intră în nodul desființat și un arc care iese din nodul desființat, arce cărora anterior li s-au atașat mulțimi, poate fi echivalat cu un arc numai dacă mulțimile atașate celor două arce ce intră în componența lanțului sunt disjuncte; în cazul în care cele două mulțimi nu sunt disjuncte, înseamnă că lanțul ce trebuie echivalat trece de două ori prin nodul desființat și, deci, el nu trebuie luat în considerare pentru că nu contribuie la obținerea unui ciclu elementar;
- echivalarea unui lanț format din două arce, cărora anterior li s-au atașat mulțimi, se face cu un arc căruia i se atașează o mulțime compusă din: elementele mulțimii atașate arcului ce intră în nodul desființat, numărul atașat nodului desființat și elementele mulțimii atașate arcului ce iese din nodul desființat; respectarea acestei reguli face ca, în final, numerele atașate nodurilor prin care trece un ciclu elementar să fie obținute în ordinea în care ciclul traversează nodurile respective;
- dacă după una sau mai multe echivalări se ajunge la situația că unui nod i se atașează o buclă, atunci aceasta reprezintă un ciclu elementar ce trece prin nodul respectiv și nodurile mulțimii asociate buclei;
- buclele atașate nodurilor se elimină deoarece ele nu pot intra în componența unui alt ciclu elementar.

Pentru exemplificarea metodei, se consideră un sistem cu evenimente discrete, caracterizat de șase stări ce aparțin unei singure clase de echivalență, sistem pentru care relația stărilor este dată prin graful reprezentat în figura 1.

Pentru sistemul considerat, se vor determina ciclurile elementare, prin metoda grafică prezentată mai sus.

Aplicarea metodei necesită numerotarea stărilor. Evident, pentru ca eliminarea stărilor să aibă loc în ordinea crescătoare a numerelor asociate lor, se face numerotarea acestora în ordinea în care se dorește eliminarea lor grafică. Se numerotează stările (porturile) sistemului din exemplu cu numere de la 1 la 6.

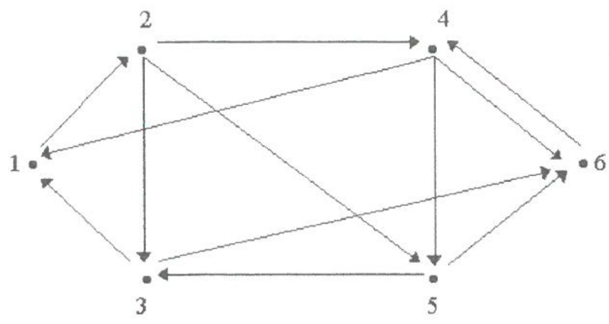

Figura 1. Graful rețelei de transport

Referitor la nodul 1, se constată că prin acesta trec lanțurile formate din câte două arce: 3→ 1→ 2 și 4→ 1→ 2, ce leagă nodurile 3 și, respectiv 4, de nodul 2.

Fiecare din cele două lanțuri se echivalează cu câte un arc căruia i se asociază mulțimea [1], corespunzător nodului desființat. Se obține graful din figura 2.

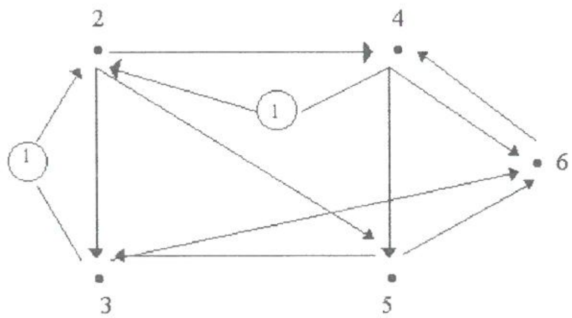

Figura 2. Graful rezultat după eliminarea nodului 1

Prin nodul 2 trec lanțurile ce leagă nodul 3 de nodurile 3, 4 și, respectiv, 5:

- 3→ [1]→ 2→ 3,
- 3→ [1]→ 2→ 4 și
- 3→ [1]→ 2→ 5,

precum și lanțurile ce leagă nodul 4 de nodurile 3, 4 și, respectiv, 5:

- 4→ [1]→ 2→ 3,
- 4→ [1]→ 2→ 4 și
- 4→ [1]→ 2→ 5.

Desființând nodul 2, lanțurile enumerate mai sus sunt echivalate cu arce caroara li se asociază mulțimea [1] [2]. Aceasta înseamnă că arcele rezultate reprezintă echivalări ale unor lanțuri care, în graful inițial din figura 1, sunt formate din câte trei arce și trec prin nodurile 1 și 2, parcurse în aceasta ordine. Se obține graful din figura 3.

Buclele atașate nodurilor 3 și 4, obținute în urma echivalărilor, corespund ciclurilor elementare B_1=[1, 2, 3] și B_2=[1, 2, 4].

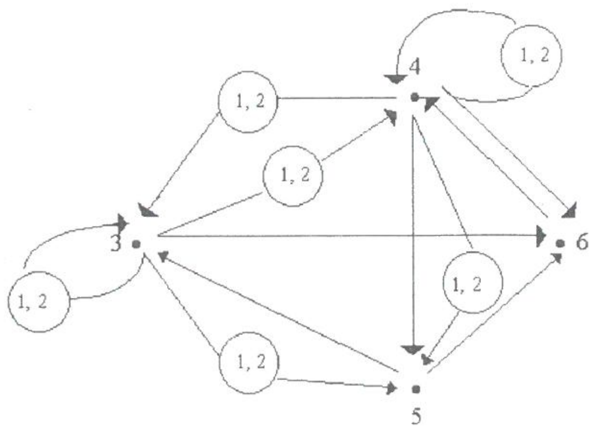

Figura 3. Graful rezultat după eliminarea nodurilor 1 și 2

Parcurgerea de către cicluri a nodurilor grafului se face în ordinea în care numerele atașate acestora apar în mulțimile B_1, respectiv B_2.

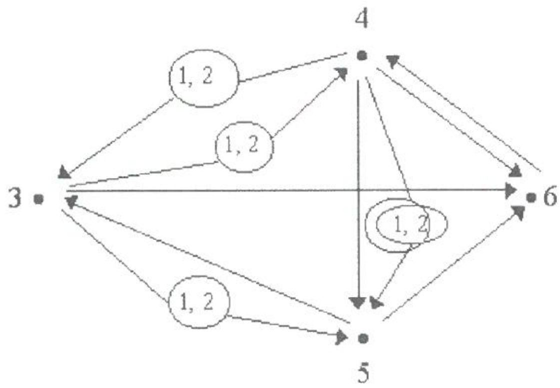

Figura 4. Eliminarea buclelor atașate nodurilor 3 și 4

Deoarece o buclă nu poate să facă parte dintr-un alt ciclu elementar, buclele atașate nodurilor 3 si 4 se elimină și rezultă graful din figura 4.Prin nodul 3 trec lanțuri ce leagă nodul 4 de nodurile4, 5 și, respectiv, 6:

- 4→ [1, 2]→ 3→ [1,2]→ 4,
- 4→ [1, 2]→ 3→ [1,2]→ 5 și
- 4→ [1, 2]→ 3→ 6,

precum și lanțuri ce leagă nodul 5 de nodurile 5, 4 și, respectiv, 6.

- 5→ 3→ [1, 2]→ 5,
- 5→ 3→ [1, 2]→ 4,
- 5→ 3→ 6.

Se observă că unele lanțuri (de exemplu 4→ [1, 2]→ 3→ [1, 2]→ 5) sau bucle (de exemplu 4→ [1, 2]→ 3→ [1, 2]→ 4) nu trebuie luate în considerare deoarece trec de mai multe ori prin unele din nodurile grafului și, deci, nu pot intra în componența unor cicluri elementare.

Desființînd nodul 3, după echivalările corespunzătoatre, se obține graful din figura 5.

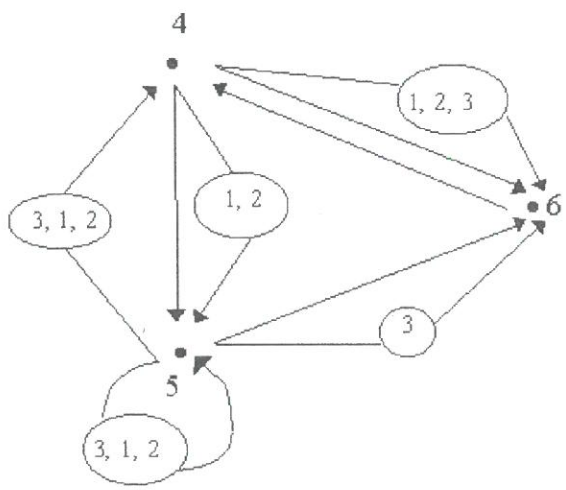

Figura 5. Eliminarea nodului 3

Nodul 5 al acestui graf conţine o buclă. Se obţine astfel ciclul elementar B_3=[3, 1, 2, 5], corespunzător buclei ataşate nodului 5. Se elimină bucla ataşată nodului 5 şi se obţine graful din figura 6.

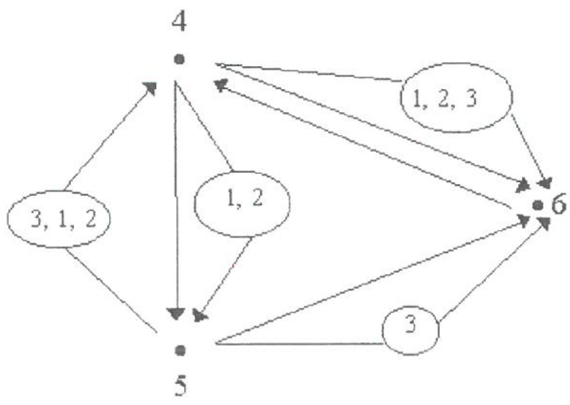

Figura 6. Eliminarea buclei nodului 5

Prin nodul 4 trec lanţuri ce leagă nodul 5 de nodurile 5 şi respectiv, 6:

- 5→ [3, 1, 2]→ 4→ 5,
- 5→ [3, 1, 2]→ 4→ [1, 2]→ 5,
- 5→ [3, 1, 2]→ 4→ 6 şi
- 5→ [3, 1, 2]→ 4→ [1, 2, 3]→ 6,

precum şi lanţuri ce leagă nodul 6 de nodurile 5 şi, respectiv, 6.

- 6→ 4→ 5,
- 6→ 4→ [1, 2]→ 5,
- 6→ 4→ 6 şi
- 6→ 4→ [1, 2, 3]→ 6.

Eliminând nodul 4, rezultă graful din figura 7.

Lanţurile 5→ [3, 1, 2]→ 4→ [1, 2]→ 5 şi 5→ [3, 1, 2]→ 4→ [1, 2, 3]→ 6 nu au fost luate în considerare deoarece trec de mai multe ori prin unele noduri ale grafului. Ele nu pot intra în componenţa ciclurilorlementar.

Corespunzător buclei nodului 5 şi celor două bucle ataşate nodului 6, se obţin încă trei cicluri elementar.

B_4=[4, 6], B_5=[4, 1, 2, 3, 6], B_6=[3, 1, 2, 4, 5].

Se elimină buclele şi se obţine graful din figura 7.

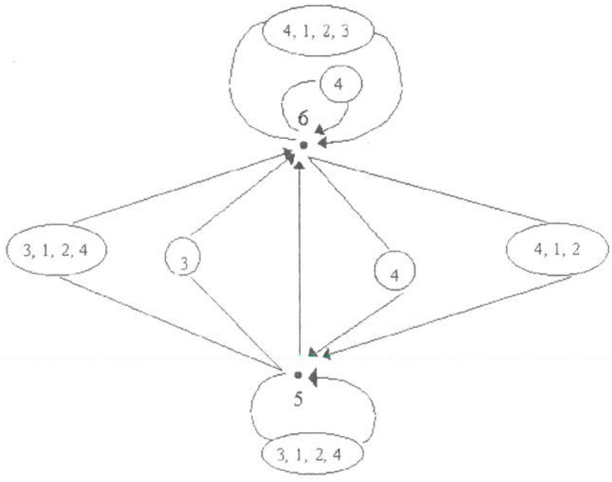

Figura 7. Graful rezultat după eliminarea nodului 5

Prin nodul 5 trec lanţuri ce leagă nodul 6 de nodul 6:

- 6→ [4]→ 5→ 6,

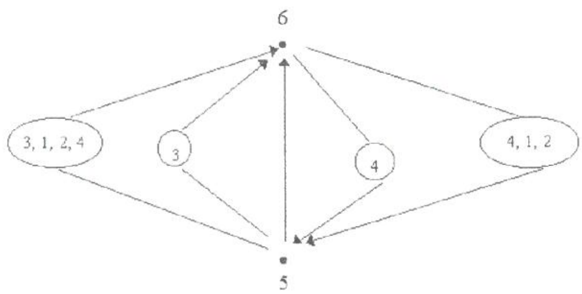

Figura 8. Graful rezultat după eliminarea buclelor nodurilor 5 şi 6

- 6→ [4]→ 5→ [3]→ 6,
- 6→ [4]→ 5→ [3, 1, 2, 4]→ 6 (trece de două ori prin nodul 4),
- 6→ [4, 1, 2]→ 6,
- 6→ [4, 1, 2]→ 5→ [3]→ 6 şi
- 6→ [4, 1, 2]→ 5→ [3, 1, 2, 4]→ 6 (trece de două ori prin nodurile 1, 2 şi 4).

Eliminând nodul 5, se obtine graful echivalent din figura 9, graf care conţine un singur nod şi patru bucle.

Corespunzător buclelor acestui graf, rezultă încă patru cicluri elementare:

B_7=[4, 5, 6],
B_8=[4, 1, 2, 5, 6],
B_9=[4, 5, 3, 6],
B_{10}=[4, 1, 2, 3, 5, 6].

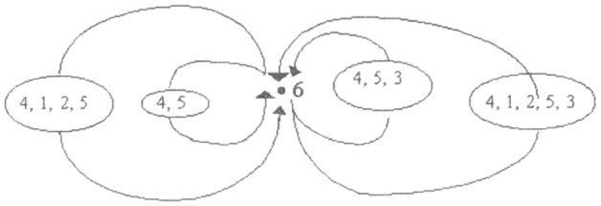

Figura 9. Graful rezultat după eliminarea nodului 5

În total, pentru sistemul studiat, se obţin zece cicluri elementare:

B_1=[1, 2, 3],
B_2=[1, 2, 4],
B_3=[3, 1, 2, 5],
B_4=[4, 6],
B_5=[4, 1, 2, 3, 6],
B_6=[3, 1, 2, 4, 5],
B_7=[4, 5, 6],
B_8=[4, 1, 2, 5, 6],
B_9=[4, 5, 3, 6],
B_{10}=[4, 1, 2, 3, 5, 6].

2. Algoritm de determinare a ciclurilor elementare

Algoritmul de determinare a ciclurilor elementare utilizează matricea relaţiei stărilor şi se bazează pe regulile metodei prezentate în paragraful anterior.

Plecând de la matricea [R], algoritmul construieşte un tablou *T*, ale cărui elemente sunt de tip mulţime şi în care fiecărei stări a sistemului îi coprespunde o linie şi, respectiv, o coloană.

În tablou se elimină succesiv stare cu stare, respectiv liniile şi coloanele asociate acestora. După eliminarea unei stări, se modifică corespunzător şi elementele tabloului, situate pe liniile şi coloanele asociate stărilor aflate în relaţie cu starea eliminată.

Pentru prezentarea algoritmului, se consideră că sistemul cu evenimente discrete este caracterizat de *n* stări. Dacă $I=\{1,2,\ldots,n\}$ reprezintă mulţimea indicilor stărilor sistemului, iar r_{ij}, unde $i,j \in I$, reprezintă elementele matricei [R], asociată relaţiei stărilor, determinarea ciclurilor elementare se poate face parcurgând următoarele etape de calcul:

Pasul 1: $s \leftarrow n$;

Pasul 2: Se construieşte tabloul *T* în care, pentru fiecare $i,j \in I$, la intersecţia liniei *i* cu coloana *j* (adică pe poziţia (i,j)) se înscrie mulţimea vidă $\varnothing$, numai dacă $r_{ij}=1$;

Pasul 3: Dacă există o valoare $i \in \{1,2,.,s\}$, astfel încât în tabloul *T*, pe poziţia (i,i), se află cel puţin o submulţime a lui *I*, vidă sau nevidă, atunci există cel puţin un ciclu ce pleacă din starea *i* şi ajunge în starea *i*:

- numărul ciclurilor este dat de numărul submulţimilor aflate pe poziţia (i,i);
- fiecare submulţime conţine indicii stărilor prin care trece un ciclu, ce pleacă din starea *i* şi ajunge în starea *i*;
- un ciclu se obţine reunind mulţimea {i} cu o mulţime de pe poziţia (i,i).

După determinarea ciclurilor, mulţimile ce au intrat în costituirea lor, situate pe poziţiile (i,i), sunt eliminate din tabloul *T*.

Atunci când $s=n$, ciclurile elementare trec printr-o singură stare. În acest caz, pe graful stărilor, acestora le corespund bucle ataşate nodurilor grafului.

Pasul 4: Dacă în tabloul *T* pentru $i, j \in \{1,2,.,s\}$, pe poziţiile (i,s) şi (s,j) există cel puţin câte o submulţime, atunci pe poziţia (i,j) se înscriu submulţimile rezultate prin reuniunea:

- fiecărei submulţimi de pe poziţia (i,s) cu
- submulţimea {s} şi cu
- fiecare submulţime de pe poziţia (s,j),

numai dacă submulţimea de pe poziţia (i,s) şi, respectiv, submulţimea reunită, de pe poziţia (s,j), sunt disjuncte;

Pasul 5: Din tabloul *T*, se elimină linia *s* şi coloana *s*;

Pasul 6: $s \leftarrow s-1$;

Pasul 7: Dacă $s>1$, calculul se continuă cu pasul 3;

Pasul 8: Stop.

Se aplică algoritmul prezentat mai sus pentru determinarea ciclurilor elementare corespunzătoare sistemului din exemplul figurii 1.

Plecând de la graful stărilor, se construieşte matricea asociată relaţiei stărilor [R]. Aceasta este reprezentată în figurile 10.

Conform algoritmului, rezultă:

Pasul 1: $s=6$;

Pasul 2: Se obţine tabloul *T* din figura 10;

	1	2	3	4	5	6
1		1				
2			1	1	1	
3	1					1
4	1				1	1
5			1			1
6				1		

Figura 10. Matricea [R]

Pasul 3: Nu există submulţimi situate pe diagonala principală a tabloului *T*;

Pasul 4: Pe poziţiile (3,6) şi (6,4) se află mulţimi vide. Rezultă că pe poziţia (3,4) trebuie să se adauge mulţimea $\{6\}=\varnothing \cup \{6\} \cup \varnothing$.

Cum şi pe perechile de poziţii: ((4,6), (6,4)) şi ((5,6), (6,4)) se află mulţimi, se adaugă mulţimea {6} pe fiecare din poziţiile (4,4) şi (5,4);

	1	2	3	4	5	6
1		$\emptyset$				
2			$\emptyset$	$\emptyset$	$\emptyset$	
3	$\emptyset$					$\emptyset$
4	$\emptyset$				$\emptyset$	$\emptyset$
5			$\emptyset$			$\emptyset$
6				$\emptyset$		

Figura 11. Tabloul *T*, pentru *s*=6

Pasul 5: Se elimină linia 6 şi coloana 6; rezultă tabloul *T* din figura 12;

	1	2	3	4	5
1		$\emptyset$			
2			$\emptyset$	$\emptyset$	$\emptyset$
3	$\emptyset$			$\{6\}$	
4	$\emptyset$			**$\{6\}$**	$\emptyset$
5			$\emptyset$	$\{6\}$	

Figura 12. Tabloul *T*, pentru *s*=5

Pasul 6: s=5;

Pasul 7: Se continuă cu pasul 3;

Pasul 3: Cum pe poziţia (4,4) se află mulţimea $\{6\}$, aceasta corespunde unui ciclu elementar. Se obţine ciclul $B_1=\{4,6\}=\{4\}\cup\{6\}$, după care mulţimea de pe poziţia (4,4) se elimină;

Pasul 4: În acest caz:

- pe fiecare din poziţiile (2,3) si (4,3) se adaugă mulţimea $\{5\}$, iar
- pe fiecare din poziţiile (2,4) si (4,4) se adaugă mulţimea $\{5,6\}=\emptyset\cup\{5\}\cup\{6\}$;

Pasul 5: Se elimină linia 5 si coloana 5; rezultă tabloul *T* din figura13;

Pasul 6: s=4;

Pasul 7: Se continuă cu pasul 3;

Pasul 3: Cum pe poziţia (4,4) se află mulţimea $\{5,6\}$, rezultă un ciclu elementar: B_2=[4,5,6]. De pe

	1	2	3	4
1		$\emptyset$		
2			$\emptyset$, $\{5\}$	$\emptyset$,$\{5,6\}$
3	$\emptyset$			$\{6\}$
4	$\emptyset$		$\{5\}$	**$\{5,6\}$**

Figura 13. Tabloul *T*, pentru *s*=4

poziţia (4,4), se elimină mulţimea $\{5,6\}$;

Pasul 4:În acest caz:

- pe poziţia (2,1) se adaugă mulţimile: $\{4\}=\emptyset\cup\{4\}\cup\emptyset$ şi $\{5,6,4\}=\{5,6\}\cup\{4\}\cup\emptyset$;
- pe poziţia (2,3) se adaugă mulţimea $\{4,5\}=\emptyset\cup\{4\}\cup\{5\}$;
- pe poziţia (2,3) nu se adaugă mulţimea $\{5,6\}\cup\{6\}\cup\{4\}$ deoarece mulţimile $\{5,6\}$ şi $\{5\}$ nu sunt disjuncte;
- pe poziţia (3,1) se adaugă mulţimea $\{6,4\}=\{6\}\cup\{4\}\cup\emptyset$;
- pe poziţia (3,3) se adaugă mulţimea $\{6,4,5\}=\{6\}\cup\{4\}\cup\{5\}$;

Pasul 5: Se elimină linia 4 şi coloana 4. Rezultă tabloul *T* din figura 14;

	1	2	3
1		$\emptyset$	
2	$\{4\}$ $\{5,6,4\}$		$\emptyset$, $\{5\}\{4,5\}$
3	$\emptyset\{6,4\}$		**$\{6,4,5\}$**

Figura 14. Tabloul *T*, pentru *s*=3

Pasul 6: s=3;

Pasul 7: Se continuă cu pasul 3;

Pasul 3: Cum pe poziţia (3,3) se află mulţimea $\{6,4,5\}$, se obţine încă un ciclu elementar $B_3=\{3,6.4,5\}$.

Se elimină mulţimea de pe poziţia (3,3).

	1	2
1		$\emptyset$
2	$\{4\}$, $\{5,6,4\}$, $\{3\}$, $\{3,6,4\}$, $\{5,3\}$, $\{5,3,6,4\}$, $\{4,5,3\}$	

Figura 15. Tabloul *T*, pentru *s*=4

Continuând efectuarea operaţiilor conform algoritmului, pentru s=2 se obţine tabloul din figura 15 cu o linie şi o coloană.

În cazul acestui tablou, pe diagonala principală nu sunt situate mulţimi deci, nu rezultă cicluri elementare.

Pentru s=1 se obţine tabloul din figura 16, cu o linie şi o coloană. În acest caz, toate submulţimile sunt situate pe poziţia (1,1).

	1
1	**$\{2,4\}$, $\{2,5,6,4\}$, $\{2,3\}$, $\{2,5,3\}$, $\{2,4,5,3\}$, $\{2,3,6,4\}$, $\{2,5,3,6,4\}$**

Figura 16.Tabloul *T*, pentru

Conform tabloului din figura 16, mai rezultă şapte cicluri elementare:

$B_4=\{1,2,4\}$,
$B_5=\{1,2,5,6,4\}$,
$B_6=\{1,2,3\}$,
$B_7=\{1,2,5,3\}$,
$B_8=\{1,2,4,5,3\}$,
$B_9=\{1,2,3,6,4\}$,
$B_{10}=\{1,2.5.3,6,4\}$.

În concluzie, sistemului analizat îi corespund zece cicluri elementare:

$B_1=\{4, 6\}$,
$B_2=\{4, 5, 6\}$,
$B_3=\{3, 6, 4, 5\}$,
$B_4=\{1, 2, 4\}$,
$B_5=\{1, 2, 5, 6, 4\}$,
$B_6=\{1, 2, 3\}$,
$B_7=\{1, 2, 5, 3\}$,
$B_8=\{1, 2, 4, 5, 3\}$,
$B_9=\{1, 2, 3, 6, 4\}$,
$B_{10}=\{1, 2, 5, 3, 6, 4\}$.

Se constată că mulțimea ciclurilor elementare obținută prin metoda grafică coincide cu mulțimea ciclurilor elementare obținută în urma parcurgerii algoritmului.

3. Determinarea unui ciclu care să tranzitez anumite stări ale sistemului de un număr impus de ori

Se asociază fiecărui ciclu elementar $B_i \in B$ un vector de dimensiune n

$v^i=(v_1^i, v_2^i, ..., v_n^i)$, (1)

unde $v_j^i=1$, dacă ciclul B_i trece prin starea x^j, sau $v_j^i=0$, în caz contrar.

Dacă unui ciclu oarecare:

$C=\alpha_1 B_1+\alpha_2 B_2+...+\alpha_t B_t$, (2)

unde $\alpha_i \in N$, $i\in\{1, 2,..., t\}$, i se asociază un vector v:

$v=(v_1, v_2, ..., v_n)$, (3)

rezultă relația vectorială:

$v=\alpha_1 v^1 +\alpha_2 v^2 +...+\alpha_t v^t$ (4)

Fiecare componentă v_i, a vectorului v, arată de câte ori ciclul C trece prin starea x^i, unde $i\in\{1, 2,..., n\}$. Este evident că vectorul v conține numai informații cu privire la numărul de treceri ale ciclului C prin fiecare stare a sistemului. El nu conține informații cu privire la legăturile relaționale dintre stări.

Din relația (4), între componentele vectorului v și componentele vectorilor v^i, $i\in\{1, 2,..., t\}$, rezultă următoarele relații liniare:

$v_j=\alpha_1 v^1_j +\alpha_2 v^2_j +...+\alpha_t v^t_j$ (5)

unde $j\in\{1, 2,..., n\}$.

Cu ajutorul relațiilor (23) se poate determina un ciclu care să treacă prin fiecare stare a unui sistem dat de un număr impus de ori, adică:

- prin starea x^1 de β_1 ori,
- prin starea x^2 de β_2 ori,...,
- prin starea x^n de β_n ori,

atunci când se cunosc vectorii asociați ciclurilor ementare.

Rezolvarea unei astfel de probleme este echivalentă cu determinarea constantelor $\alpha_1, \alpha_2, ..., \alpha_t$, astfel încât:

$v=\alpha_1 v^1+\alpha_2 v^2+...+\alpha_t v^t=(\beta_1, \beta_2, ..., \beta_n)$, (6)

Ecuația vectorială (6) este echivalentă cu un sistem de cel mult n ecuații liniare cu coeficienți întregi pozitivi.

Sistemul rezultat poate fi compatibil sau incompatibil, iar în cazul în care este compatibil, el poate avea una sau mai multe soluții.

Pentru exemplificare, se presupune că graful din figura 1 corespunde unui sistem de transport maritim: fiecare nod corespunde unui port maritim, iar fiecare arc reprezintă o legatură comercială dintre două porturi ale rețelei de transport.

Dacă se cere să se determine drumul pe care trebuie să-l urmeze o navă comercială pentru a încărca și (sau) descărca mărfuri:

- de patru ori în portul 1,
- de patru ori în portul 2,
- de trei ori în portul 3,
- de trei ori în portul 4,
- o dată în portul 5 și
- de două ori în portul 6.

O soluție a problemei constă în determinarea, cu ajutorul relației (6), a unui ciclu neelementar caracterizat de vectorul $v=(4,4,3,3,1,2)$.

Pentru graful din figura. 1, $S=\{1, 2,..., 6\}$, $n=6$, iar ciclurile elementare, determinate în cadrul paragrafului 2, sunt:

$B_1=\{1, 2, 3\}$,
$B_2=\{1, 2, 4\}$,
$B_3=\{3, 1, 2, 5\}$,
$B_4=\{4, 6\}$,
$B_5=\{4, 1, 2, 3, 6\}$,
$B_6=\{3, 1, 2, 4, 5\}$,
$B_7=\{4, 5, 6\}$,
$B_8=\{4, 1, 2, 5, 6\}$,
$B_9=\{4, 5, 3, 6\}$,
$B_{10}=\{4, 1, 2, 3, 5, 6\}$.

Evident, numărul ciclurilor elementare este $t=10$, iar mulțimea ciclurilor elementare este $B=\{B_1, B_2, ..., B_{10}\}$.

Vectori asociați ciclurilor elementare din mulțimea B rezultă:

$v^1=(1,1,1,0,0,0)$,
$v^2=(1,1,0,1,0,0)$,
$v^3=(1,1,1,0,1,0)$,
$v^4=(0,0,0,1,0,1)$,
$v^5=(1,1,1,1,0,1)$,
$v^6=(1,1,1,1,1,0)$,
$v^7=(0,0,0,1,1,1)$,
$v^8=(1,1,0,1,1,1)$,
$v^9=(0,0,1,1,1,1)$,
$v^{10}=(1,1,1,1,1,1)$.

Ecuația vectorială (24), în acest caz particular, devine:

$\alpha_1 v^1+\alpha_2 v^2+...+\alpha_{10} v^{10}=(4,4,3,3,1,2)$, (7)

și este echivalentă cu următorul sistem:

$$\begin{cases} a_1 + a_2 + a_3 + a_5 + a_6 + a_8 + a_{10} = 4 \\ a_1 + a_2 + a_3 + a_5 + a_6 + a_8 + a_{10} = 4 \\ a_1 + a_3 + a_5 + a_6 + a_9 + a_{10} = 3 \\ a_2 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9 + a_{10} = 3 \\ a_3 + a_6 + a_7 + a_8 + a_9 + a_{10} = 1 \\ a_4 + a_5 + a_7 + a_8 + a_9 + a_{10} = 2 \end{cases} \tag{8}$$

Se constată că sistemul are numai cinci ecuații diferite, primele două ecuații fiind identice.

Rezolvând sistemul, se obține soluția:

$\alpha_1 = \alpha_2 = \alpha_5 = \alpha_{10} = 1$,

$\alpha_3 = \alpha_4 = \alpha_6 = \alpha_7 = \alpha_8 = \alpha_9 = 0$,

deci un ciclu neelementar ce satisface condițiile problemei este:

$$C = B_1 + B_2 + B_5 + B_{10}. \tag{9}$$

4. Concluzii

Lucrarea a avut ca obiectiv studiul posibilităților algoritmizării determinării ciclice de deplasare a unei nave printr-o submulțime de posturi dintr-o rețea de transport maritim.Originalitatea rezolvarii problemei constă în faptul că rețeaua de transport este descrisă prin mulțimea ciclurilor elementare, posibile în rețeaua de transport. În acest sens, se propune o metodă grafică și un algoritm care permite determinarea pe calculator a ciclurilor elementare dintr-o rețea de transport, dată prin mulțimea posturilor și a traseelor admisibile de legătura dintre posturi.

Mulțimea ciclurilor elementare care descrie rețeaua permite determinarea unui traseu ciclic neelementar al navei, ca o combinație liniară de cicluri elementare. În aceste condiții, s-a propus un algoritm algebric pentru determinarea unui ciclu care să tranziteze anumite posturi de un anumit număr impus de ori. Această metodă a permis înlocuirea metodelor de căutare în grafuri prin metode algebrice existente pentru construirea traseelor ciclice condiționate ale navelor în rețeaua dată.

Bibliografie

1. GH. BORDEA, M. TERTIŞCO, J.CULIŢĂ: Binary Petri Nets in the Backstrom - Klein State Space. În: Proc. of the CSCS-11, Bucharest, Mai 1997. pp.252-254.
2. M. TERTIŞCO, J.CULIŢĂ: The Plan Existence Problem for a Discrete Event System. În: Proc. of IFAC, 4th SSC' 97, Bucharest, October 1997, pp. 204-207.

LINGUASTAT

Corpora Processing Platform

Target Users

- current users: researchers in NL area, faculty teaching Corpus Linguistics courses, language resources developers, students Computer Science.
- potential users: lexicographers, grammarians, students in linguistics

Short Description

LINGUASTAT is a collection of corpora processing tools (language independent) and several language resources for Romanian. Most of the tools were developed within the European projects MULTEXT-EAST and TELRI. Among the available tools there are a text tokenizer (a modified version of the MTSeg-developed by LPL Aix-en-Provence), a 3-gram tagger (an enhanced version of the QTag developed at the University of Birmingham), a "recovering" module (mapping from an internally-used small tagset (89 tags) onto a large tagset (611 tags) as found in morpho-syntactic lexicons), a rule-based disambiguator for the "statistically hard" ambiguities. LINGUASTAT accepts as input either plain texts or SGML annotated texts (it is CES/TEI aware)

The environment also contains some Romanian-specific utilities such as diacritics insertion (either ISO-LAT2 or SGML entities), hyphenation, as well as some language models for different language registers (fiction, news, etc). A training corpus (more than 300,000 tagged words, hand validated), a large dictionary of more than 450,000 wordforms entries (EAGLES compliant) and more than 15,000,000-words texts from various language registers (fiction, news, technical literature, philosophy, etc.) are available.

The system is under continuous development (for instance a concordancer, a lemmatizer, a combined classifier method for disambiguation and several statistical utilities will be soon included). The system, implemented in a client-server architecture (Java, C, Perl and TCL/TK), is in use at Research Institute for Informatics, Bucharest and the Romanian Academy as an in-house tool for various corpora-based projects, but soon LINGUASTAT will be installed as a free service on the web.

Technical Requirements

The system runs under UNIX operating systems (it was tested under both SOLARIS and LINUX).