BOOTSTRAPPING FINITE AUTOMATS AND GRAMMARS IN MODELLING AN ECONOMIC PROCESS

IOSIF MIRCEA NEAMȚU
Department of Computer Science
Lucian Blaga University
5-7 Ioan Ratiu Str, Sibiu
ROMANIA

IOAN POP
Department of Computer Science
Lucian Blaga University
5-7 Ioan Ratiu Str, Sibiu
ROMANIA

Abstract: Modelling and simulating economic processes is a discipline that lies on the border to mathematics and informathics and is all about fundamenting the manager's decision in efficiency conditions for the manufacturer, with the help of flexible economic-mathematic models and with the possiblitity of using simulation techniques.
Within the simulation activity, three key- elements are involved: the real system, the model (pattern), the computer and two relations: relation and simulation patterns.
„The real system" represents the system apprehended with human senses.
„The real model" represents the replaced real system which corresponds, basically, to the demands of the initial real system.
„The abstract model" achieves the crossing-over from the „real system" to the „real model"; it reproduces the real system by breaking it down into elementary components and astablishing bonds among those.
For the demonstration of those above, we take into consideration the following problem:
For a business to carry out its functional conditions at the moment of Romania's adherance to the EU, within an economic section , it needs to accomplish a goal that can be finalized by achieving time/quality indicators withind a well preestablished order.

Key-Words: Modeling, Simulating, Process, Event, Mode, Automat, Lines

1. Introduction

Let's consider a graphic Γ finit, oriented, valuated and without any circuits, that patterns a research problem because a plan of activities is needed, meaning, a complex of lime and space limited assignments.
A certain state in realizing the project is called *event* and it is represented by a tip of the graphic; also, any section of the project has a beginning and an end in distinctive events that are time-consuming, called *activity* and it is represented by an arch.
The needed time for an activity represented by an arch is the value of that certain arch and it is called *operative time.* The graphic also displays an *entrance event* – the tip x_0 and a *closing event* – the tip x_n.
We establish the graphics of a program and make sure that it does not have any circuits. We are interested in the time of the establishment of the activity, meaning the length of the established program. This length can't be inferior to the sum of the operative times taken from the most adverse path from x_0 to x_n, meaning that the result between these two points is the maximum sum of operative times. This path is called *critical path (*we need to take into consideration that several paths can occur).
The graphic is called *valued graphic* if a function v: U → R exists, so that for any $u = (x_i, x_j) \in U$, $v(u) = v(xi, xj) \geq 0$, i, j = 1.. n. The number v(u) is the distance between the tips x_i and x_j and is called the *the value of the arch* u and can also stand for the costs or duration in time.
If d is a path in the valued graphic Γ, the sum of it's arches values is called *the value of the path d.*

2. Case study

For Γ = (X, U) a oriented finite graphic with the *events* (knots) $X=\{x_0,x_1,x_2,x_3,x_4,x_5,x_6\}$ and that consumes a duration of time called *activity* represented by an arch, so that:

$$U = \{(x, y) \mid x \in X, y \in T(x)\}, \text{ if } x\in X, \text{ then } T(x) \subset X.$$

U represents the multitude of activities (of the arches), u=(x,y) is an arch of the graphic, with the *initial extremity* (the origin) of the arch u, and the *final extremity* of u is:

U={(0,1), (0,2), (0,3), (1,2), (1,4), (2,4), (2,5), (2,6), (3,2), (3,5), (4,5), (4,6), (6,6)};

T is an application of X on the multitude of its parts:

$T(x_0)=\{x_1,x_2,x_4\}$, $T(x_1)=\{x_2,x_4\}$, $T(x_3)=\{x_4,x_5,x_6\}$, $T(x_3)=\{x_2,x_5\}$, $T(x_4)=\{x_5,x_6\}$, $T(x_0)=\{x_6\}$,

The times needed to accomplish every goal v(u) (characteristically for browsing the arches):
Where: $v : U \rightarrow R$, so that for any $u = (x_i, x_j) \cup U$, $v(u) = v(x_i, x_j) \geq 0$, $0 \leq i,j \leq 6$;

$v(x_0,x_1)=3$, $v(x_0,x_2)=5$, $v(x_0,x_3)=4$, $v(x_1,x_2)=6$, $v(x_1,x_4)=7$, $v(x_2,x_4)=2$, $v(x_2,x_5)=7$, $v(x_2,x_6)=10$, $v(x_3,x_2)=3$, $v(x_3,x_5)=9$, $v(x_4,x_5)=5$, $v(x_4,x_6)=3$, $v(x_5,x_6)=4$.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6
x_0	0	**3**	**5**	**4**	0	0	0
x_1	0	0	**6**	0	**7**	0	0
x_2	0	0	0	0	**2**	**7**	**10**
x_3	0	0	**3**	0	0	**9**	0
x_4	0	0	0	0	0	**5**	**3**
x_5	0	0	0	0	0	0	**4**
x_6	0	0	0	0	0	0	0

By considering, as being known, the lengths of every accomplishing phase in weeks- the values of the arches; we would like to find out:

1) The shortest time for accomplishing the goal
2) The times: the shortest and the longest of every operation from achieving the goal
3) Establishing the way of controlling the achievement of the plan in order to avoid or minimize potential delays.

1) We determine the matrix of the arches A:

The graphics Γ = (X, U) with $X=\{x_0,x_1,x_2,x_3,x_4,x_5,x_6\}$.

The matrix A = (aij), $0 \leq i, j \leq 6$, with:

$$a_{ij} = \begin{cases} 1, \text{dacă există arcul } (x_i, x_j) \\ 0, \text{în cazul contrar} \end{cases}$$

It is called the *arch's matrix* or the *matrix of the direct connections*. It determines the graphic in a unique way and it establishes a new way of defining a graphic.

A	x_0	x_1	x_2	x_3	x_4	x_5	x_6
x_0	0	**1**	**1**	**1**	0	0	0
x_1	0	0	**1**	0	**1**	0	0
x_2	0	0	0	0	**1**	**1**	**1**
x_3	0	0	**1**	0	0	**1**	0
x_4	0	0	0	0	0	**1**	**1**
x_5	0	0	0	0	0	0	**1**
x_6	0	0	0	0	0	0	0

To see if the graphic has any circuits, we determine the matrix D (*the path matrix* or the *matrix of total connections*) with $D = (d_{ij})$, $1 \leq i, j \leq n$:

$$d_{ij} = \begin{cases} 1, \text{dacă există cel puțin un drum de la } x_i \text{ la } x_j \\ 0, \text{în cazul contrar} \end{cases}$$

The path matrix can determine many graphics. This way, the graphics, although different, have the same path matrix.

The D matrix marks out the existence of circuits in the considered graphic. According to the theorem, if: $d_{ii} = 1$, the graphic has a path that starts from x_i and returns to x_i, meaning that the graphic has circuits. In this case, the graphic has no circuits because all elements of the main diagonal from D are equal to 0.

In this case (the graphic has no circuits), we attach to D the column of touching exponents $p(x_i)$, i = 0..6, in order to obtain the triangular matrix D'.

D	x_0	x_1	x_2	x_3	x_4	x_5	x_6	$p(x_i)$
x_0	0	**1**	**1**	**1**	**1**	**1**	**1**	**6**
x_1	0	0	**1**	0	**1**	**1**	**1**	**4**
x_2	0	0	0	0	**1**	**1**	**1**	**3**
x_3	0	0	**1**	0	**1**	**1**	**1**	**4**
x_4	0	0	0	0	0	**1**	**1**	**2**
x_5	0	0	0	0	0	0	**1**	**1**
x_6	0	0	0	0	0	0	0	**0**

If in the matrix D of a finite graphic with n tips and no circuits, we align the lines and columns in a decreasing manner after the *touching exponents of the tips*, we obtain a new matrix D'; called the *superior triangular matrix* of the D matrix:

$D' = (d'_{ij})$, $1\leq i, j \leq n$, with all elements equal to 1 above the main diagonal.
In deed, by marking $x'_1, \ldots, x'_n$ the order of the lines (columns) in D' and considering $d'_{ij} = 1$, $i < j$, it

results that there is at least one path from x'_i to x'_j, meaning that the tips touched by x'_j will also be touched by x'_I; $p(x'_i) > p(x'_j)$ that indicates the alignment of the line I in front of the line j, therefore $d'_{ij} = 1$ will be found above the main diagonal in D'.
In our case, the graphic has no circuits and D' is obtained from D by changing the order between x_2 and x_3, both on the lines and on the columns in the following way:

D'	x_0	x_1	x_3	x_2	x_4	x_5	x_6	$p(x_i)$
x_0	0	**1**	**1**	**1**	**1**	**1**	**1**	**6**
x_1	0	0	0	**1**	**1**	**1**	**1**	**4**
x_3	0	0	0	**1**	**1**	**1**	**1**	**4**
x_2	0	0	0	0	**1**	**1**	**1**	**3**
x_4	0	0	0	0	0	**1**	**1**	**2**
x_5	0	0	0	0	0	0	**1**	**1**
x_6	0	0	0	0	0	0	0	**0**

Γ = (X, U) is an oriented graphic, with $X=\{x_0,x_1,x_2,x_3,x_4,x_5,x_6\}$ and U the amount of its aches.

The graphic Γ is called *valued graphic* if a function $v : U \rightarrow R$ exists, so that for any $u = (x_i, x_j) \in U$, $v(u) = v(x_i, x_j) \geq 0$, i, j = 1..6.
The number v(u) is called the *arch's value* u and can stand for, in our case, the needed time for achieving the goal between the tips x_i and x_j.
If *d* is a path in the *valued graphic* Γ, the sum of the values of its arches will be called *the value of the path* d.
Henceforth we will determine the *minimum value* of the paths from any tip of the graphic Γ to a certain tip x_n. The matrix $V = (v_{ij})$, $1 \leq i, j \leq 6$, with:

$$v_{ij} = \begin{cases} 0 & , i = j \\ v(x_i, x_j), \text{ dacă există arcul } (x_i, x_j) \\ \infty & , \text{ dacă nu există arcul } (x_i, x_j). \end{cases}$$

We mark by $m^{(k)}_m$ the minimum value of the paths between x_i and x_n, paths built up from maximum k arches, and by m_m the minimum value of paths from x_i to x_n, regardless from the number of arches.

This is the V matrix with the order of tips from D':

V	x_0	x_1	x_3	x_2	x_4	x_5	x_6
x_0	0	**3**	**4**	**5**	-∞	-∞	-∞
x_1	-∞	0	-∞	**6**	**7**	-∞	-∞
x_3	-∞	-∞	0	**3**	-∞	**9**	-∞
x_2	-∞	-∞	-∞	0	**2**	**7**	**10**
x_4	-∞	-∞	-∞	-∞	0	**5**	**3**
x_5	-∞	-∞	-∞	-∞	-∞	0	**4**
x_6	-∞	-∞	-∞	-∞	-∞	-∞	0

We determine the matrix $M^{(2)} = V * V$, where the elements:

$$m^2_{i,j} = \max(v_{ik} + v_{kj}),\ i,j = 0..6.$$

in which, for example, the element $m^2_{4,6}$ is obtained by transposing the line of x_4 from V to the column of x_6; then we sum up the gained columns and retain the highest value. So that:
$m^2_{4,6}$=max(-∞+(-∞), -∞+(-∞), -∞+(-∞), -∞+10, 0 + 3, 5+4, 3+0) = 9,

$M^{(2)}$	x_0	x_1	x_3	x_2	x_4	x_5	x_6
x_0	0	3	4	9	10	3	15
x_1	-∞	0	-∞	6	8	13	16
x_3	-∞	-∞	0	3	5	10	13
x_2	-∞	-∞	-∞	0	2	7	11
x_4	-∞	-∞	-∞	-∞	0	5	**9**
x_5	-∞	-∞	-∞	-∞	-∞	0	4
x_6	-∞	-∞	-∞	-∞	-∞	-∞	0

Then we calculate $M^{(3)} = M^{(2)} * V$, $M^{(4)} = M^{(3)} * V$, a.s.o. until $M^{(k+1)} = M^{(k)}$ and the algorithm ends; in the base of the sentence that demonstrates that for any graphic Γ with n tips, the maximum length of an elementary path is equal to n - 1, a $k \in N$ exists so that:

$$m^{(k)}_m = m^{(k+1)}_m,\ 1 \leq k \leq n,\ \text{when: } m^{(k)}_m = m_m .$$

We calculate $M^{(4)} = M^{(3)} * V$ and observe that $M^{(3)} = M^{(4)}$, so that $M^{(3)} = M$, where M is the *matrix of maximum values* of the paths between any two tips pf the graphics, *paths formed from any two arches.*
Observation: to accelerate the calculation speed, we can determine the matrix $M^{(k)}$ in the following order: $M^{(2)}$, $M^{(4)} = M^{(2)} * M^{(2)}$, $M^{(8)} = M^{(4)} * M^{(4)}$ a.s.o. until the last two coincide.

M	x_0	x_1	x_3	x_2	x_4	x_5	x_6
x_0	0	3	4	9	11	16	**20**
x_1	-∞	0	-∞	6	8	13	17
x_3	-∞	-∞	0	3	5	10	14
x_2	-∞	-∞	-∞	0	7	7	11
x_4	-∞	-∞	-∞	-∞	0	5	9
x_5	-∞	-∞	-∞	-∞	-∞	0	4
x_6	-∞	-∞	-∞	-∞	-∞	-∞	0

From where we read $v[d_{max}(x_0,x_6)]=20=m_{06}$, meaning that the goal can be achieved the soonest in 20 weeks.

2) to determine the tips through which the path with the minimum value passes, for example from x_i to x_j, we transpose the line of x_i from V over the column of x_j from M and sum up the pairs of elements searching for the sum (sums) with the maximum value equal to $m^{(k+1)}_m$. If this sum lies on the line of x_k, the first arch of the searched path is (x_i, x_k). Then we transpose the line of x_k from V over the column of xj from M and similar to that we determine the next tip of the route, a.s.o. until we get to xj.

By retaining the maximum value equal to m_{06}=20, we transpose the line of x_0 from V over the column of x_6 from M and sum up the pairs of elements, transpose the line of x_1 from V over the column of x_6 from M and sum up the pairs of elements, a.s.o. so that:

for x_0 from V and x_6 from M we obtain:

$$\max\{0+20;\ 3+17;\ 4+14;\ 5+11;\ -\infty+9;\ -\infty+4;\ -\infty+0\} = 20,$$

this sum was on the line of x_1, so the first arch of the critical path is (x_0, x_1).

We transpose the x_1 line from V over the x_6 column from M; so that:

for x_1 from V and x_6 from M we have:

$$\max\{-\infty+20;\ 0+17;\ -\infty+14;\ 6+11;\ 7+9;\ -\infty+4;\ -\infty+0\} = 17,$$

and the maximum value will correspond to x_2 line, the next arch is (x_1, x_2).

We transpose x_2 line from V over the x_6 column from M; so that:

$$\max\{-\infty+20;\ -\infty+17;\ -\infty+14;\ 0+11;\ 2+9;\ 7+4;\ 10+0\} = 11,$$

the maximum value will correspond to x_4 line, the next arch will be (x_2, x_4) or (x_2, x_5).

We transpose the x_4 line from V over the x_6 column from M; so that:

$$\max\{-\infty+20;\ -\infty+17;\ -\infty+14;\ -\infty+11;\ 0+9;\ 5+4;\ 3+0\} = 9,$$

and the maximum value will correspond to $x_4$line, the next arch being (x_4, x_5).

We transpose the x_5 line from V over the x_6 column from M; so that:

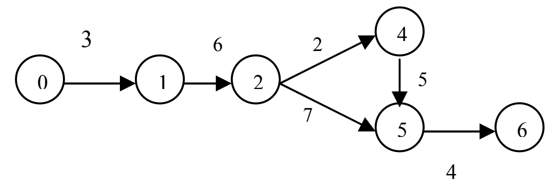
$$\max\{-\infty+20;\ -\infty+17;\ -\infty+14;\ -\infty+11;\ -\infty+9;\ 0+4;\ 4+0\} = 4,$$

and the maximum value will correspond to the x_4 line, the next arch is (x_5, x_6).

Two different critical paths can be distinguished; obviously, both have the same value 20:

$$d_1 = x_0 \xrightarrow{3} x_1 \xrightarrow{6} x_2 \xrightarrow{2} x_4 \xrightarrow{5} x_5 \xrightarrow{4} x_6,$$

$$d_2 = x_0 \xrightarrow{3} x_1 \xrightarrow{6} x_2 \xrightarrow{7} x_5 \xrightarrow{4} x_6$$



The critical events are: x_0, x_1, x_2, x_4, x_5, x_6.

Event x_i	**t^d_i**	**t^t_i**	**Margin $t^t_i - t^d_i$**
x_0	0	0	0
x_1	3	3	0
x_3	4	6	2
x_2	9	9	0
x_4	11	11	0
x_5	16	16	0
x_6	20	20	0

Where t^d_i = v[dmax(x0, xi)], i = 1..6, are read on the x_0 line in M, and

t^t_i = v[dmax(x_0, x_6)] – v[dmax(x_i, x_6)] = 20 – v[dmax(x_i, x_6)],

Where v[dmax(x_i, x_6)], i = 0..6, are real on the column x_6 from M, there fore:

t^t_2 = v[dmax(x0, x6)] – v[dmax(x2, x6)] = 20 – v[dmax(x2, x6)]=20-14= 6

t^t_5 = v[dmax(x0, x6)] – v[dmax(x5, x6)] = 20 – v[dmax(x5, x6)]=20- 4=16

By reading the table, we conclude that the only event than can accept a delay is x_3, by two weeks, maximum.

3) *The critical activities* are represented by the arches that are components of the *critical paths*. For the analysis of potential delays, we order the calculus in the following table:

Activities	**t^d_i**	**vij**	**t^t_j**	**t^t_i**	**t^d_j**	**Margin**
(x_0,x_1)	0	3	3	0	3	0
(x_0,x_2)	0	5	9	4	5	4
(x_0,x_3)	0	4	6	2	4	2
(x_1,x_2)	3	6	9	3	9	0
(x_1,x_4)	3	7	11	4	10	1
(x_3,x_2)	4	3	9	6	7	2
(x_3,x_5)	**4**	**9**	**16**	**7**	**13**	**3**
(x_2,x_4)	9	2	11	9	11	0
(x_2,x_5)	9	7	16	9	16	0
(x_2,x_6)	9	10	20	10	19	1
(x_4,x_5)	11	5	16	11	16	0
(x_4,x_6)	11	3	20	17	14	6
(x_5,x_6)	16	4	20	16	20	0

Let's explain, for example, how the line of activities (x_3, x_5) was filled in.

$\mathbf{t^d_i}$ read from (the line x_0 and the column x_3); therefore;

{(0,0),(0,0),(0,0),(0,1),(0,1),(0,3),**(0,3)**,(0,2),(0,2),(0,2),
0, 0, 0, 3, 3, 4, **4**, 9, 9, 9,

(0,4),(0,4),(0,5)}
11, 11, 16

$$t^d_i = t^d_3 = v[d_{max}(x_0, x_3)] = 4,$$

$\mathbf{v_{ij}}$ read from the graphic or from V like this:

{(0,1),(0,2),(0,3),(1,2),(1,4),(3,2),**(3,5)**,(2,4),(2,5),
3, 5, 4, 6, 7, 3, **9**, 2, 7,

(2,6),(4,5),(4,6),(5,6)
10, 5, 3, 4

$$v_{ij} = v_{35} = 9;$$

$\mathbf{t^t_j}$ the first term read from the x_0 line and the column x_6, and the second term from the line x_5 and the column x_6 from M;

$t^t_j = t^t_5 = v[d_{max}(x_0, x_6)] - v[d_{max}(x_5, x_6)] = 20 - 4 = 16$,

$\mathbf{t^t_i}$ the elements of the column t^t_i are obtained by deducting from the column t^t_j the column v_{ij}

$t^t_i = t^t_j - v_{ij}$, in this case $t^t_3 = t^t_5 - v_{35} = 16 - 9 = 7$;

$\mathbf{t^d_j}$ the elements of the column t^d_j are obtained by adding the elements from the columns t^d_i and v_{ij}

$t^d_j = t^d_i + vij$, becomes $t^d_5 = t^d_3 + v_{35} = 4 + 9 = 13$

The margin is given by $t^t_i - t^d_i$ or by $t^t_j - t^d_j$, in this case $\mathbf{t^t_i - t^d_i}$ or $\mathbf{t^t_j - t^d_j}$;

$$t^t_3 - t^d_3 = t^t_5 - t^d_5 = 7 - 4 = 16 - 13 = 3.$$

The critical activities have a zero margin and the other ones have to be monitored according to its increasing order.

3. Models, Finite automats and grammars

An automatic match of the lines of each P model exists; this automat has to be built (starting from the given model) before it can be used for any search. The next figure illustrates this building for the P model=ababa or P=abba. Thenceforth, we will assume that P is a fixed model.

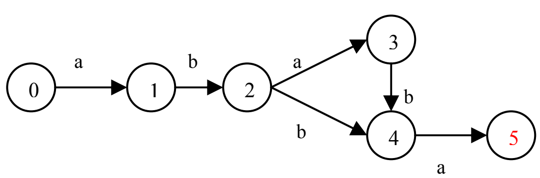


Fig. 1

We are searching for the adequate grammar of the following finite automat.

AF=({a,b},{s_0,s_1,s_2,s_3,s_4,s_5},f), S_f={s_5}, $f(s_0,a)=s_1$, $f(s_1,b)=s_2$, $f(s_2,a)=s_3$, $f(s_2,b)=s_4$, $f(s_3,b)=s_4$, $f(s_5,a)=s_5$,

Input: AF = (I, ST, f), S_f – semiautomat finite determinist. The amount of states is *ST* so that it will not be mistaken for the starting grammar symbol *S*.

Output: G = (V_N, V_T, S, P) – type 3 grammar.

Methode: V_N :=ST, V_T := I, S = s_0 .

$$P := \begin{cases} \{s_0 \to \lambda\}, \text{dacă } s_0 \in S_f \\ \varnothing, \quad \text{dacă } s_0 \notin S_f \end{cases}$$

In order to find the grammar, the next procedure can be used:

PROCEDURE Gramatica_AutomatuluiFinit
I₁:=I // I₁- auxiliary mass
DO WHILE ($I_1 \neq \varnothing$)
For $i \in I_1$; $I_1 := I_1 \setminus \{i\}$
S_1=ST // S_1 - auxiliary mass
DO WHILE ($S_1 \neq \varnothing$)
For $s \in S_1$; $S_1 := S_1 \setminus \{s\}$
IF $f(s,i) \in S_f$
THEN
$P := P \cup \{s \to i\} \cup \{s \to is_1 \mid \forall\ s_1 \in f(s,i)\}$
ELSE
$P := P \cup \{s \to is_1 \mid \forall\ s_1 \in f(s,i)\}$
ENDIF
ENDDO
ENDDO
END PROC

These can be successfully used, by applying automats that match lines so that a P model=ababa can be analized like this:

state	entrance a	entrance b	p
0	**1**	0	a
1	0	**2**	b
2	**3**	**4**	a
3	0	**4**	b
4	**5**	0	a
5	0	0	

(a)

i	–	1	2	3	4	5	6	7	8
T[i]	–	a	b	**a**	**b**	**a**	**b**	**a**	a
$\phi(T_i)$	0	1	2	3	4	5	4	5	6

(b)

Fig. 2

To clarify the way of function of the line matching automat, we will present thenceforth an easy and effective program that simulates its behaviour (represented by its transitory function δ) in search for appearances with a "m" length within an entrance text T[l..n]. For a model of the size "m", the mass of states *Q* is {0, 1, ..., m} and the starting state is 0;the only acceptance state being m:

```
FINITE-MATCHING-AUTOMAT (T,δ,m)
  n ← length[T-text]
  q ← 0
  for i ← 1, n do
  q ← δ(q,T[i])
  if  q = m then
  s ←  i - m
  type "Model appears with the displacement" s
```

A diagram of the transitional states for the line matching automat that accepts all lines ended in ababa. The state 0 is the starting state and the state 5 (coloured in black) is not the only acceptance state. An arch oriented from the state i to the state j, labelled with a, represents $\delta(i, a)$ = j. The arches oriented to the right, the ones that form the "spinal column" of the automat, represented by bold writing, coincide to the correct matches between model and entrance characters. The arches orientated to the left coincide to incorrect matches. Some arches, coinciding incorrect matches, are not represented; through convention, if a state i does not have an outgoing arch labelled with a for $a \in \Sigma$, then $\delta(i, a) = 0$.

The function corresponding to δ and the model line P = ababa. The entrances corresponding to correct matches between model and entrance characters are shaded.

The function of the automat for the text T = abababaaaba. Under each character of the text T[i], we recognise the state $\phi(T_i)$ in which the automat lies, after processing the prefix T_i. An appearance of the model is determined and it ends within the 6^{th} position.

References

[1]. Les Bases de la classification authomatique Lerman, I.C., Paris Gauthier Villars, 1970,

[2]. Tratat de programarea calculatoarelor. Algoritmi fundamentali, Knuth, D. E. Bucuresti Ed. Tehnica 1974 (traducere),

[3]. Algebraic linquistics. Analytical Models Marcus, S. – New York end London Academic Press, 1967;

[4]. Lingvistica matematica Marcus, S. –, Editura Didactica si Pedagogica Bucuresti, 1973;

[5]. Introducere in algoritmi, R. Rivest, C.E. Leiserson, T.H. Cormen, 2000,

[6]. Modelling the dialogue by means of formal language theory, Gh. Paun, C.L.T.A. 25, 1, 1980,