FIRST INTERNATIONAL CONFERENCE
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

On Particular Class of Location-Transition Petri Nets: State Machine

Marin Popa, Mariana Popa, Mihaita Dragan

Abstract

This article discusses important properties of a particular class of Petri nets, class state machine. This class allows finit automate and sequential processes modeling. A deficiency of this class is not allows synchronization of independent processes. Obtain here some theoretical results in a bimarked state machine and shows that any program can be associated in a Petri net's of this class. Some properties of such programs for their accuracy can be studied using results of the class state machine.

Keywords: state machine monomarked, state machine bimarked, viability, surety, S-conflict matrix, S- confluence matrix, T-conflict matrix, T-confluence matrix.

This particular class of Petri Net's the automats modeling and finite sequential processes. Operation of a state machine simulating a multitude of independent processes which unfortunately can not be synchronized in this particular class.

1. PRELIMINARY NOTIONS

In this section we present some concepts and theoretical results of the theory of Petri networks wich are required to demonstrate results that determine the class type state machine of Petri networks.

DEFINITION 1.1 [3] Be $\Sigma = (S, T, pre, post)$ a Petri net and $\overline{\Pr e}$, $\overline{Post}$, two finite matrix size respectively constructed as follows:

$$\overline{\Pr e}(s,t) = \begin{cases} 1, \Pr e(s,t) \neq 0 \\ 0, \Pr e(s,t) = 0 \end{cases}, \quad \overline{Post}(s,t) = \begin{cases} 1, Post(s,t) \neq 0 \\ 0, Post(s,t) = 0 \end{cases}$$

a) We call the matrix T - symmetric conflict matrix $CT = \overline{\Pr e}^t \bullet \overline{\Pr e}$, of size $|T| \times |T|$.

b) We call S-matrix symmetric conflict matrix $CS = \overline{Post}^t \bullet \overline{Post}$ $CS = \overline{Post}^t \cdot \overline{Post}$, of size $|S| \times |S|$.

c) We call T-matrix confluence symmetric matrix $TC = \overline{Post} \bullet \overline{Post}^t$, of size $|T| \times |T|$.

d) We call S-matrix confluence of symmetric matrix $SC = \overline{\Pr e} \bullet \overline{\Pr e}^t$, of size $|S| \times |S|$.

e) We call the matrix T - precedence symmetric matrix $TP = \overline{Post} \bullet \overline{\Pr e}$, of size $|T| \times |T|$.

f) We call the matrix S - precedence symetric matrix $TP = \overline{\Pr e} \bullet \overline{Post}$, of size $|S| \times |S|$.

PROPOSITION 1.2 [3]

Be a Σ finite Petri net, CT, CS, TC, SC, TP, SP conflict matrix, precedence matrix, respectively confluence matrix defined above and $s, \bar{s} \in S$, $t, \bar{t} \in T$. Then :

a) $CT(t,\bar{t}) = \left|{}^{\bullet}t \cap {}^{\bullet}\bar{t}\right|$, and $CT(t,t) = \left|{}^{\bullet}t\right|$

b) $CS(s,\bar{s}) = \left|{}^{\bullet}s \cap {}^{\bullet}\bar{s}\right|$, and $CS(t,s) = \left|{}^{\bullet}s\right|$

c) $TC(t,\bar{t}) = \left|t^{\bullet} \cap \bar{t}^{\bullet}\right|$, and $TC(t,t) = \left|t^{\bullet}\right|$

d) $SC(s,\bar{s}) = \left|s^{\bullet} \cap \bar{s}^{\bullet}\right|$, and $SC(s,s) = \left|s^{\bullet}\right|$

e), $TP(t,\bar{t}) = \left|{}^{\bullet}t \cap {}^{\bullet}\bar{t}\right|$, and $TP(t,t)$ = number of loops formed by *t* and a location to some.

f), $SP(s,\bar{s}) = \left|{}^{\bullet}s \cap {}^{\bullet}\bar{s}\right|$, and $SP(s,s)$ = number of loops formed by s and a transition to some.

DEFINITION 1.3 [5]

Be Σ_M = (S, T, Pre, Post) a marked Petri net, and $A(\Sigma_M, \mu_0)$ multitude of accessible marks of the network. It's called the graph of marks the digraph accessible labeled: $G_A(\Sigma) = (A(\Sigma_M, \mu_0), T, \Gamma)$ where for label $\forall \mu \in A(\Sigma_M, \mu_0)$, $\Gamma_\mu = \{ \mu' \in A(\Sigma_M, \mu_0) \mid \exists t \in T$ so that $\mu[t > \mu' \}$, and T is the multitude of labels for the digraph arcs.

The arch μ [> μ 't is labeled with the t $\Leftrightarrow$ $\mu[t > \mu'$. It notes that $G_A(\Sigma)$ can be finite or infinite as so as $A(\Sigma_M, \mu_0)$ is finite or not. Because from μ_0 we can reach to all the markings of the $A(\Sigma_M, \mu_0) \Rightarrow G_A(\Sigma)$ is a related digraph.

DEFINITION 1.4 [5]

Let be $\Sigma_M = (\Sigma, \mu_0)$ a Petri net marked and $t \in T$ a transition of the network.

a) We say that transition *t* it is cvasiviable $\Leftrightarrow$ $\exists \mu \in A(\Sigma_M)$ so that $\mu[t>$.

b) Petri net Σ_M it is cvasiviable $\Leftrightarrow$ $\forall$ $t \in T$, transition *t* it is cvasiviable.

c) We say that transition *t* it is viable $\Leftrightarrow$ $\forall \mu \in A(\Sigma_M)$, transition *t* it is cvasiviable (Σ, μ)

d) Petri net Σ_M it is viable $\Leftrightarrow$ $\forall$ $t \in T$, transition *t* it is viable.

e) Petri net it is viable $\Leftrightarrow$ $\exists \mu \in \mathbf{N}^{|S|}$ so that the network (Σ, μ) to be viable.

Cvasiable of a transition that gives the opportunity to occur at least once and so that the operation for which is represented in the system modeled is not important in functioning of the system. The viability of a transition expresses that in any moment in the evolution of the transition can occur to a specific mark which is a characteristic of systems with continuous operation and for which a unavailability of a operation corresponds to a feather (error) of the system.

DEFINITION 1.5 [5]

Let be $\sum_M = (S,T,Pre,Post,\mu_0)$ a Petri net marked and $s\in S$ a given location.

a) We say that *s* it is k-bordered ($k\in \mathbf{N}^*$) $\Leftrightarrow$ $\forall\mu\in A(\sum_M)$ we have $\mu(s)<=k$.

For k = 1 we say that *s* it is binary.

We say that *s* is bordered $\Leftrightarrow$ $\exists\ k\in\mathbf{N}^*$ that *s* it is *k*-bordered.

b) $\sum_M$ is bordered $\Leftrightarrow$ $\forall s\in S$, $\exists\ k\in\mathbf{N}^*$ that *s* it is *k*-bordered.

c) A Petri net $\sum$ is bordered $\Leftrightarrow$ $\forall\mu\in N^{|S|}$, the marked network $\sum_M = (\sum,\mu)$ it is bordered.

d) We say that $\sum$ it is sure if throughout its evolution any location of the network it is binary.

PROPOSITION 1.6 [5]

For any $\sigma\in T^*$ $\exists\ \mu\in N^{|S|}$ marking a network so that $\mu[\sigma\rangle$.

PROPOSITION 1.7 [5]

Let be $\sum_M=(\sum, \mu_0)$ a Petri net marked and $t\in T$ a transition of his life. The transiton *t* is viable $\Leftrightarrow$ $\forall\mu\in A(\sum_M)$ $\exists\mu'\in A(\sum, \mu)$ and $\exists\ \sigma\in T^*$ repetitive sequence containing *t* and such that $\mu'[\sigma>$.

2. THEORETICAL CONSIDERATIONS

DEFINITION 2.1 [1] Let Σ = (S, T, Pre, Post) a PT - Petri net {0,1} - value.

a) a) We say that Σ is the state machine $\Leftrightarrow \forall t\in T$ we have $\left|{}^{\bullet}t\right| = \left|t^{\bullet}\right| = 1$ (equivalent to $\forall t\in T$, $\sum_{s\in S}\Pr e(s,t)=1$ and $\sum_{s\in S}Post(s,t)=1$).

b) We say that the machine state Σ is monomarked $\Leftrightarrow$ for $\forall\mu_0$ initial marking $\exists! s\in S$ so that $\mu_0(s)=1$.

If an initial marking μ_0 is allowed a single mark on a particular network location, then the input transitions that have concession at this location μ_0 and may produce one of them. Thus, the mark will move from location to location designating the current state of the machine. If permitted in μ_0 several grades, then by producing the transition grid, these brands moving independently simulating a variety of independent processes on the same program. State machines but do not allow the synchronization of these processes.

OBSERVATION 2.2.

Any scheme logic associated with a program is transformed into a PT-network (0,1)-valuated. Indeed, let P a program (a set of instructions that can be executed in a certain order) and let be SP = (B, Q) a logical schem associated with a logical layout program, when B it is a multitude of blocks for scheme logical , and Q it is a multitude of its arcs.

We associate to location P a PT-net $\sum$ P = (S, T, Pre, Post) as follows:

Each location corresponds to an arch from Q or a lot of arcs with same node terminal , each transition corresponds to a block b from B if it is not test block (block predicative) or transitions k from T correspond to a test block b from B, where k it is aritate for b- and Pre, Post:$S \times T \rightarrow \mathbf{N}$ are given by:

$\Pr e(s,t) = \begin{cases} 1, \exists t' \in T, s = (t',t) & \text{(t it is final node of s)} \\ 0, otherwise \end{cases}$ $Post(s,t) = \begin{cases} 1, & \text{t it is intial node of arc s} \\ 0, otherwise \end{cases}$

Obtain such a PT net in wich transitions correspond to the scheme blocks , and the locations correspond arcs of a logical scheme. If $\sum_P$ is a state machine monomarked, then by producing the transitions, mark is moving from location to location and indicating a instruction wich to be executed next .

PROPOSITION 2.3 [1]

Let be Σ a state machine monomarked and hard-related, μ_0 an initial marking, the $G_\Sigma=(S \cup T, \Gamma)$ associated the digraph and his Σ and GA $G_A(\Sigma)=(A(\Sigma, \mu), T, \overline{\Gamma})$ graph marking accessible Σ 's definition given in 2.2. Then $G_A(\Sigma)$ and T-labeled graph :

G=(S, T, Γ^2) are isomorphic.

PROOF

Because Σ it is monomark by producing the transitions, the unique mark of the network transitions moving from location to location. How $\left|t^{\bullet}\right| = 1$ for $\forall t \in T$ result that $\forall \mu \in A(\sum, \mu_0), (\sum, \mu)$ it is monomark and for $\left|t^{\bullet}\right| = 1$, $\forall t \in T$ and Σ it is hard-related result that $\forall t \in T$ can producing at a specific marker (to which the only marked location, is it his entry) and that the number of different Σ marking is equal to the number of locations in S. Thus $\left|A(\sum, \mu_0)\right| = |S|$ and so there is a correlation between S and $A(\sum, \mu_0)$ which associating biunivoc a marker μ_s of a location s and is it given by $\mu_s(s') = \begin{cases} 1, s' = s \\ 0, s' \neq s \end{cases}$.

Since $\overline{\Gamma}\mu = \{\overline{\mu} \in A(\sum, \mu_0) \mid \exists t \in T, \mu[t > \overline{\mu}\}$ we have that $\forall \mu, \overline{\mu} \in A(\sum, \mu_0)$ we have $\overline{\mu} \in \overline{\Gamma}\mu \Leftrightarrow \exists s, \overline{s} \in S, \exists t \in T$ that $\mu = \mu_s$ $\overline{\mu} = \mu_{\overline{s}}$ şi $\mu_s[t > \overline{\mu}_{\overline{s}} \Leftrightarrow \exists s, \overline{s}$ şi $\exists t \in T$ that $\Gamma(s) = t$ and $\Gamma t = \overline{s} \Leftrightarrow \exists s, \overline{s} \in S$ şi $\exists t \in T$ that $\overline{s} = \Gamma t = \Gamma(\Gamma s) = \Gamma^2 s \Leftrightarrow \exists s, \overline{s} \in S$ that $\overline{s} \in \Gamma^2 s$. It follows that $\forall(\mu, \overline{\mu})$ arch in $G_A(\Sigma)$, $\exists(s, \overline{s})$ arch in G corresponding from bijective above him $(\mu, \overline{\mu})$.
Conversely, it is obvious because $(s, \overline{s})$ the arch in G, take the $(\mu_s, \mu_{\overline{s}})$ arch in $G_A(\Sigma)$.

COROLLARY 2.4

Let be Σ a state machine monomarked with n = | S | location and marking μ some of it. Then the number of the accessible markings is finished, and in addition $\left|A(\sum, \mu_0)\right| \leq n$.

PROOF.

Obviously by moving a marking unique from location to location we obtained new markings of the network, in which one location is marked. So each marker has a single element 1 and 0 the rest, and this element can occupy at most n-positions, can be obtained at most n-different markings. The maximum number is reached when the state machine it is hard-connex, because of the isomorphism established in the previous theorem.

PROPOSITION 2.5.

Let be Σ a machine state monomarked and hard-connex. Then Σ is it a PT-net viable and sure.

PROOF.

Let be $\forall\mu_0$ a initial marker of a $\sum$. We will show that $(\sum, \mu_0)$ it is viable and sure. Since Σ it is hard-connex $\Rightarrow$ the graph G=(S, T, Γ^2) it is hard-connex and according to previous 2.5, the graph of the marks accessible $G_A(\Sigma)$ it is hard-connex, which means $\forall\mu, \overline{\mu} \in A(\sum, \mu_0), \exists\sigma \in T^*$ that $\mu[\sigma > \overline{\mu}$.

It follows that after a certain sequence of procedures is it possible re-obtaining any marking $\mu \in A(\sum, \mu_0)$. How in a state machine $\forall t \in T$ is permitted at a marker μ (ie one that assigns 1 only its entry) and how μ to obtain the sequence σ what containing t, so a repetitive sequence, resulting in 1.7 sentence that it is viable. Σ therefore follows that is it viable.

Fact that Σ it is sure resulting from definition of a Σ namely from $\sum_{s \in S} Post(s,t) = 1$ we obtain that $\exists! s \in S$ so that Post(s, t)=1. In other words, by producing t marking unique of the network is transformed from entry of t of exit its s ie and $\forall\mu \in A(\sum, \mu_0)$ şi $\forall s \in S$ we have $\mu(s) \le 1$.

THE EXAMPLE 2.6.

Σ Petri net of figure 1 is a state machine with initial marking $\mu_0 = (100)'$.

It is noted that: $\mu_0[a > \mu_1 = (010)'$
$\mu_1[b > \mu_2 = (001)'$
$\mu_2[c > \mu_1, \mu_2[d > \mu_0$

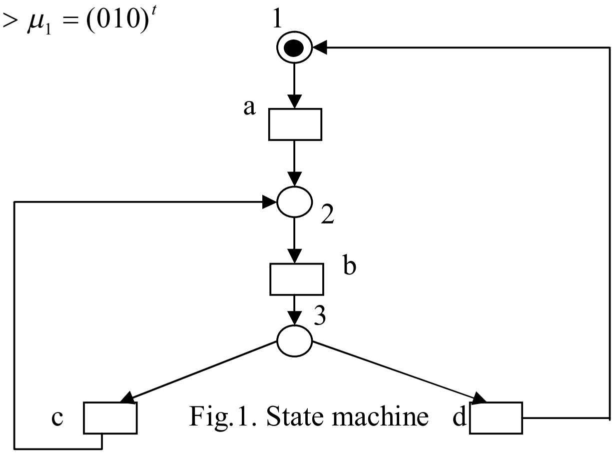
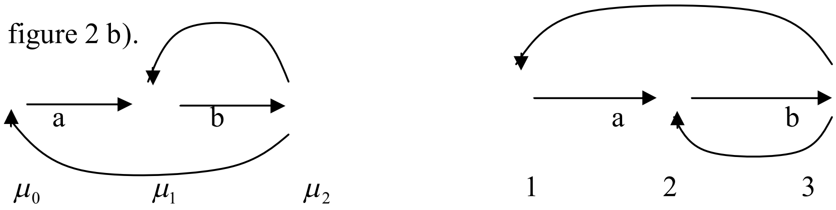


Fig.1. State machine

So : $A(\Sigma, \mu_0) = (\mu_0, \mu_1, \mu_2)$

The graph's accessible markings it is in the figure 2 a) and graph G=(A, T, Γ^2) it is in figure 2 b).



a) the graph's accessible markings b) graph G=(A, T, Γ^2)

Fig. 2. The isomorphic graphs

Obviously the two graphs are isomorphic and Σ is viable and sure.

Follows a characterization theorem of the machine state by a matrix T-conflict and T-confluence defined in 1.1.

THEOREM 2.7 Let be Σ= (S, T, Pre, Post) a PT-Petri network (0, 1)-value and CS, SC, CT, TC matricx S-conflict, S-confuence, T-conflict, T-confluence defined in 1.1. Then:

a) Σ is it a machine state $\Leftrightarrow$ CT(t, t)=1,TC(t, t)=1 for $\forall t \in T$ for (have all elements of the diagonally 1)

b) If Σ it is the state machine then CS and SC are matrix diagonal.

c) If CS and SC are diagonal matrix then Σ it is state machine $\exists t \in T$ that $t^\bullet = \Phi$ or ${}^\bullet t = \Phi$.

PROOF.

a) Using proposition 1.2 a) and c) we have CT(t, t)=$\left|{}^\bullet t\right|$ and TC(t, t)=$\left|t^\bullet\right|$ for $\forall t \in T$. From definition we have Σ state machine $\Leftrightarrow$ $\left|{}^\bullet t\right| = \left|t^\bullet\right|$=1 $\Leftrightarrow$ TC(t, t)=CT(t, t)=1 for $\forall t \in T$.

b) If Σ it is a state machine $\Rightarrow \forall t \in T$ has one entrance and one exit and so t can not be many entries in two different locations would be any different because $t \in T$ that will be $\left|t^\bullet\right|$=2>1. It follows therefore that $\forall s, \bar{s} \in S, s \neq \bar{s}$ we have ${}^\bullet s \cap {}^\bullet \bar{s} = \Phi$ which means that CS(s, $\bar{s}$)=|${}^\bullet s \cap {}^\bullet \bar{s}$|=0.

Analog $\forall t \in T$ output can not be common to two different locations that otherwise would have

that $\left|t^\bullet\right|$=2>1 and hence $\forall s, \bar{s} \in S, s \neq \bar{s}$ we have $s^\bullet \cap \bar{s}^\bullet = \Phi$. Thus we deduce from 1.2. item d) SC(s, $\bar{s}$)=$\left|s^\bullet \cap \bar{s}^\bullet\right| = 0$, $\forall s, \bar{s} \in S, s \neq \bar{s}$.

c) If CS and SC are diagonal we have $\Rightarrow \forall s, \bar{s} \in S, s \neq \bar{s}$ avem CS(s, $\bar{s}$)=SC(s, $\bar{s}$) = 0 and hence the 1.2. gain ${}^\bullet s \cap {}^\bullet \bar{s} = \Phi$ and ${}^\bullet s \cap \bar{s}^\bullet = \Phi$. It follows that $\forall t \in T$, ${}^\bullet t$ și $t^\bullet$, and can contain no more than one location, ie $\left|{}^\bullet t\right| \leq 1$ and $\left|t^\bullet\right| \leq 1$. This means that time Σ it is a state machine or that $\exists t \in T$ that t has no input and / or any output.

COROLLARY 2.8.

Let be Σ a PT- net {0, 1}-value and hard-connex. Then Σ it is the state machine $\Leftrightarrow$ SC and CS are diagonals matrix.

PROOF.

Because Σ it is hard-connex, resutls that $\forall t \in T$ we have $t^{\bullet} \neq \Phi$ and ${}^{\bullet}t \neq \Phi$ and then point c) of the theorem says that SC and CS are diagonals $\Rightarrow \Sigma$ it is machine state. The reciprocal is even point b) of the theorem

Follows an interesting result by fact that it specifies an upper edge for the number of the accessible markings in a machine state bimarked.

DEFINITION 2.9

Let be Σ= (S, T, Pre, Post) a state machine. We say that Σ it is bimarked $\Leftrightarrow$ $\forall \mu_0$ initial marking of the network, μ_0 puts on network locations only two marks .

PROPOSITION 2.10

Let be Σ= (S, T, Pre, Post) a bimarked state machine, with n locations and μ_0 is it marking initial.

Then the lot of markings accessible is finished and in addition $\left|A(\sum, \mu_0)\right| \leq \dfrac{n(n+1)}{2}$.

PROOF.

The two markings which they puts on the network locations could both be attributed to a location one or one to two different locations. In the first case, the transitions from output of that location have concession at μ_0 and producing one of them, one of a marking is moved to a new location. In this situation by producing one or another of the transitions located at the exit of the two locations, are obtained new markings that does not change the number of locations marked and this because $\forall t \in T$ we have $\left|{}^{\bullet}t\right| = 1$ which means that the number of locations is not diminishing and we have $\left|t^{\bullet}\right| = 1$ which says that the number of locations marked increase not. It follows therefore that any marking μ of the network will mark the only two locations and that μ may not have than the elements 0, 1 and 2. Since the number of possible combinations of these values on the n-locations of the network, so their sum be 2 is finite, results that the number of markings accessible is finite .

Obviously the maximum value is obtained when the machine state is hard-connex because otherwise if there is at least a location with no one exit when a marking arrive into it will she not be able to move never from s, and so certain marking can not be obtained (it is as if a marking can be in a fixed location and the other would cover a part or all of the other locations).

In this case will be n-markings that assigns each locations with two markings, and the others leave them unmarked .

Assume that the locations are: S={s_1, s_2,……, s_n}.

If a mark is puts in s_1 then other mark can occupy any of the other n-1 free locations , and so, in this case we obtain n-1 distinct markings by the firsts. Similarly, if a mark is placed in the s_2 then the other can handle any of the other n-2 locations $s_3,\dots s_n$ free (do not use and not to s_1 with a marker already obtained in the previous step).

By recursive procedure is apparent that if one marking the deals locationsn s_{n-1} then the other can only deal on sn obtain the marking $(00\ldots011)^t$. So the total number of markings obtained in this case is :

$$n+(n-1)+\ldots+1=\frac{n(n+1)}{2}.$$

PROPOSITION 2.11

Let be Σ a bimarked state machine and hard-connex. Then Σ it is a PT-net viable and 2-bordered.

PROOF.

We will show that for any marking initial μ_0 of the network, $(\sum,\mu_0)$ is viable and 2-bordered. Since Σ it is hard-connex, results and graph of the marking accessible $G_A(\Sigma)$ is hard-connex, reduce to the absurdity that if there is a marking from which no longer leaves any arc to another marking would mean that at least one of two locations indicated that the marking has no output arc to a transition of any network, meaning that $\exists s \in S$ that $\forall t \in T$, Pre(s, t)=0. But this means that s and t can not be linked by road and therefore would not be Σ loud related. $G_A(\Sigma)$ is hard-connex that $\Rightarrow \forall \mu, \overline{\mu} \in A(\sum,\mu_0)$, $\exists \sigma \in T^*$ that $\mu[\sigma > \overline{\mu}$. Because in a state machine $\forall t \in T$ is allowed to μ a bookmarking and how μ to re-obtain after a repetitive sequence σ containing t resulting from 2.1. that it is viable and how it is certain, $(\sum,\mu_0)$ it is viable.

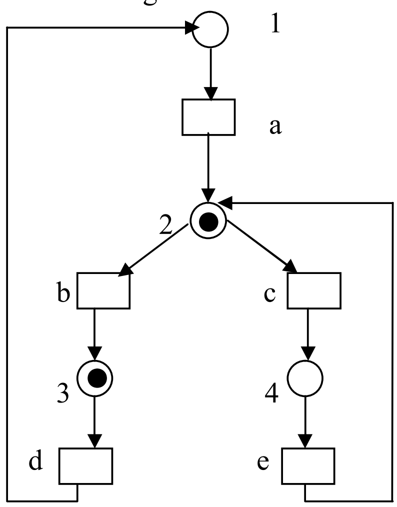
That Σ that is 2-bordered resulted from the proof previous, where was shown earlier that $\forall \mu \in A(\sum,\mu_0)$ şi $\forall s \in S$ we have $\mu(s) \le 2$.

THE EXAMPLE 2.12

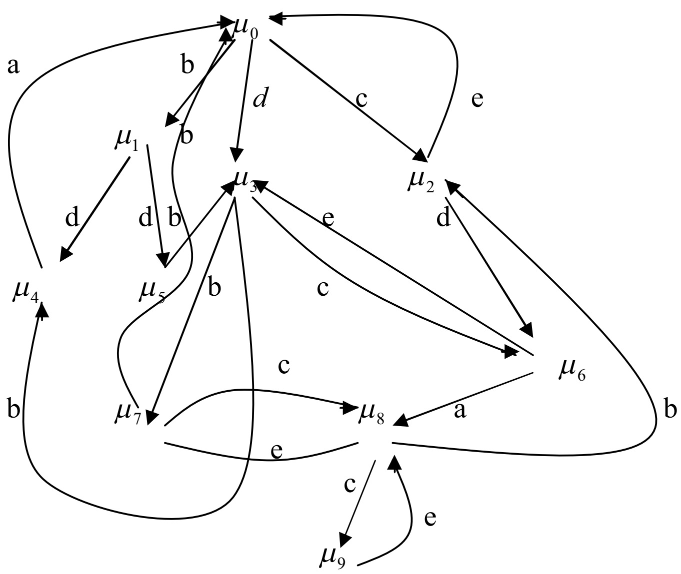
Let the hard-connex state machine from figure 3.6.a) in wich $\mu_0 = (0110)^t$.

It is noted that, $A(\sum, \mu_0) = \{\mu_0, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \mu_7, \mu_8, \mu_9\}$, where $\mu_1(0020)^t$, $\mu_2 = (0011)^t$, $\mu_3 = (1100)^t$, $\mu_4 = (1010)^t$, $\mu_5 = (2000)^t$, $\mu_6 = (1001)^t$, $\mu_7 = (0200)^t$, $\mu_8 = (0101)^t$, and $\mu_9 = (0002)^t$. Graph marking accessible ∑'s is given in figure 3.6.b). Since ∑ it is hard-connex observed that $G_A(\sum)$ is hard-connex.

Applying the proposition 2.10 we get for n = 4.10 marked ∑ 's accessible, which is easily seen on the figure. Machine state is viable and 2-bordered as seen on the figure.



a) Bimarked state machine



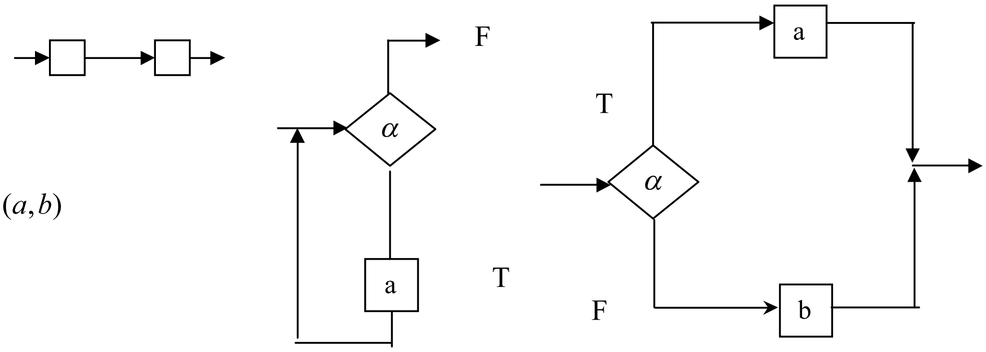
b) Graph marking accessible
Fig. 3. State machine

In what follows we prove that any program P can be associated state machine. For this we use the results of structured programming [] including the structure theorem of Bohn and Jacopini.

DEFINITION 2.13 [4]

We call the basic structures of structured programming structures: SEQVENCE noted $\pi(a,b)$ (sequential structure), WHILE-DO noted $\Omega(\alpha,a)$ (repetitive structure subject before) and IF-THEN-ELSE noted $\Delta(\alpha;a,b)$ (alternative structure) where a and b are functional blocks (operations on certain basic variables), and α the block is predicative which selects the next operation to be run.

We will show in figure 4 the basic structures of the above.



a) $\pi(a,b)$

b) $\Omega(\alpha,a)$ c) $\Delta(\alpha;a,b)$

Fig 4 The basic structures of structured programming

DEFINITION 2.14. [2]

Let SP = $(F \cup P, Q)$ schedule an associated program logic P where F is many blocks fonction of the scheme, many blocks predicative P, $F \cap P = \Phi$ and Q many arcs. We introduce three functional blocks F, T, K$\notin F$ and a block predicative $\omega \notin P$ in the following manner. Block F transforms any object x in the pair (0, x), T transforms any object x in the pair (1, x) where 0 and 1 values are indeed true and false respectively associated K block any pair (v, x) with $v \in \{0,1\}$ the of the second component to x. So F (x) = (0, x), T (x) = (1, x),

K(v, x) = x.

Predicative block ω is defined by $\omega(v,x)=1 \Leftrightarrow v=1$ (ω checked $\Leftrightarrow$ first component of the pair (v, x) is 1).

THEOREM 2.15. [2]

Any logical schema SP = $(F \cup P, Q)$ related to a program P can be transformed into a logical schema structured SP $'=(F'\cup P', Q')$ using basic structures defined in 2.13. and where $F' = F \cup \{F, T, K\}, P' = P \cup \{\omega\}, Q = Q' = Q \cup A$, A is a lot of arcs required legării blocks F, T, K, ω with blocks of SP.

PROOF. We found in [2].

THEOREM 2.16.

Let P be a program and SP = (B, Q) a logical scheme associated with P..

Whether $\sum$ P = (S, T, Pre, Post) PT-Petri net associated with SP included in note 2.2.

Then $\sum$ P is a state machine.

PROOF. Using theorem 2.15. will be sufficient to prove that the basic structures π, Ω, Δ are associated Petri subrețele are sections of a state machine. Using the construction in remark 2.2. basic structures are associated with sections of Petri networks fig 5 [a), for π, b) for Ω, c) for Δ].

It is noted that a block with two predicative values of truth and is associated if the two transitions T is predicative

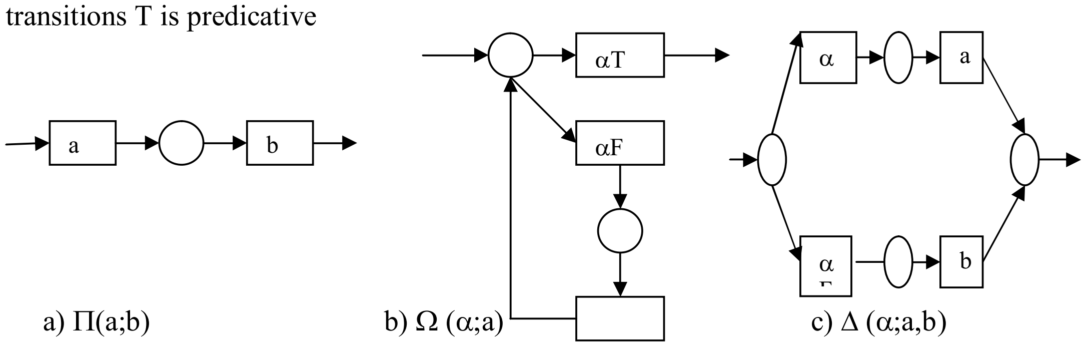


Fig. 5. Sections Petri networks associated with the basic structures of structured programming true and αF if false. Also note that all three sections of Petri networks are state machines because each transition has one input and one output.

OBSERVATION 2.16.

It is not necessary for structured logic diagram SP' by theorem 2.15. to use all auxiliary blocks placed in the definition 2.14. In fact if the initial schedule does not contain cycles (return to blocks already completed the transition from START to STOP) will not be any need for auxiliary block structure but will be used procedure codes halving [39] which consists of repeating certain blocks on every branch of $\Delta(\alpha;a,b)$ when this is necessary for the structuring and Π and Δ. If the schema contains original ciclări who can not speak and only Π and Δ can use technique of Boolean variable described in [123] leading to the use of auxiliary blocks T, F, K and for structuring ω.

THE EXAMPLE 2.17**.**

Let P be the following program:

Let be determine max (x, y, z) for x, y, z read from the keyboard, and print the result.

A logical schema SP associated with P is given in Fig. 6

Since isn't need for recovery SP it is structured only by Π and Δ.

The network Petri, associated with SP, as we can see in construction from remark 2.2 is given in Fig. 7 and notes on drawing that present a state machine.

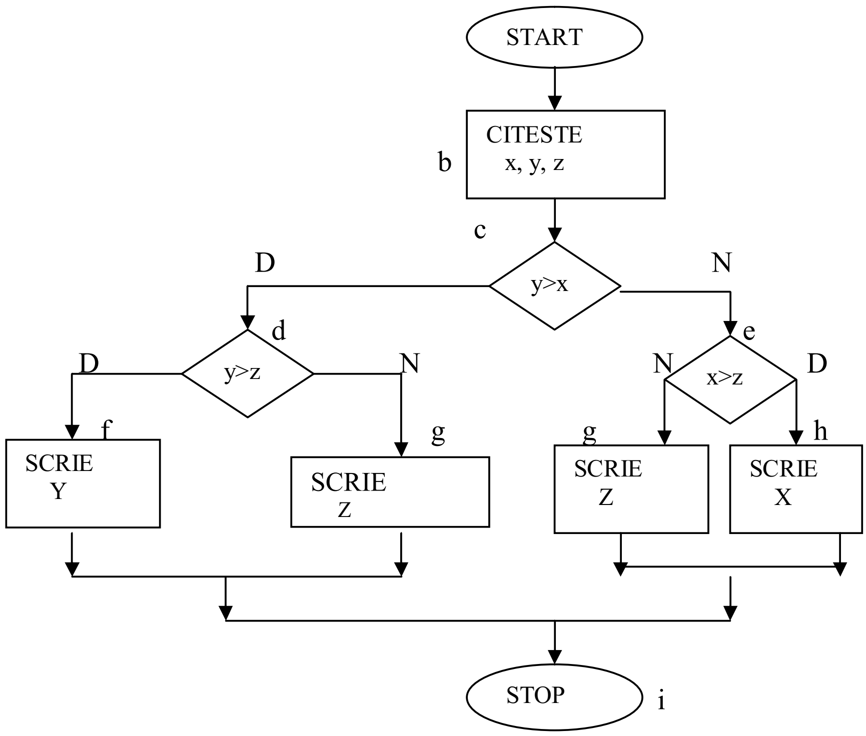


Fig. 6. Logical structural scheme

It is noted that, for the structuring is necessary of g block dedublarea . in the state machine associated with this block is associated not two transition, only one transition.

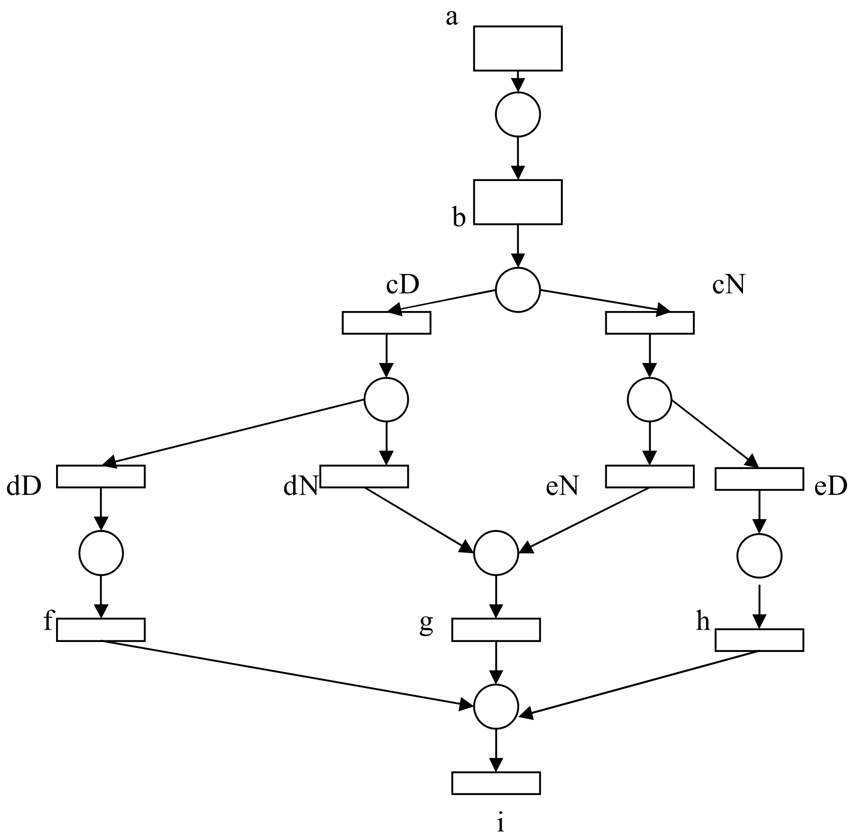


Fig.7. State machine associated to a program

CONCLUSIONS

This article highlights the importance of Petri networks in modeling discrete systems. Even with this class of the Petri net called the state machine class, which is one of the simplest classes, it is noted of this article, that can be obtained results interesting in complex research fields such as automata theory, sequential processes and accuracy study programs.

REFERENCES

[1]. E. Best, A. Merceron-Brecht - *"Some properties of nonsequential processes",* IFS Report 82. 07 (august 1982), Gesellschaft fur mathematik und Datenverarbeitung MBH, Bonn.
[2]. M. Popa I. Vaduva - *"Culegere de probleme de sisteme informatice",* Tipografia Universitatii Bucuresti, 1988.
[3]. J. L. Johnson, T. Murata - *"Structure matrix for Petri nets and their applications*", Journal of Franklin Institute, Vol 319 Nr. 3 march, 1985.
[4]. L. Dumitrascu , A. Ioachim - *"Tehnici de construire a programelor cu structuri alternative*", Editura Academiei, 1981..
[5]. G. Brams - *"Reseaux de Petri: Theory et Practique*", Mason, Paris, 1983

THE AUTHORS

MARIN POPA

UNIVERSITATEA BUCURESTI
DEP. TEHNOLOGII
FAC. : TEHN. INFORMATIEI
BLVD. M.KOGALNICEANU
NR. 36-46, SECTOR 5
COD POSTAL 70709
BUCURESTI, ROMANIA
marpopa2002@yahoo.com

MARIANA POPA

UNIVERSITATEA BUCURESTI
DEP. TEHNOLOGII
FAC. : TEHN. INFORMATIEI
BLVD. M. KOGALNICEANU
NR. 36-46, SECTOR 5
COD POSTAL 70709
BUCURESTI, ROMANIA
marpopa2002@yahoo.com

MIHAITA DRAGAN

UNIVERSITATEA BUCURESTI
DEP. TEHNOLOGII
FAC. : TEHN. INFORMATIEI
BLVD. M. KOGALNICEANU
NR. 36-46, SECTOR 5
COD POSTAL 70709
BUCURESTI, ROMANIA
dragan_mihaita@yahoo.com