БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ . BULGARIAN ACADEMY OF SCIENCES

ПРОБЛЕМИ НА ТЕХНИЧЕСКАТА КИБЕРНЕТИКА И РОБОТИКАТА, 51
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, 51

София . 2001 . Sofia

Computer Aided Synthesis of Interacting Processes Models*

Hristo Hristov, Tasho Tashev

Institute of Information Technologies, 1113 Sofia

1. Introduction

The design of modern information systems requires the investigation of interacting information processes occurring in information computing systems and sets, which are characterized by their multi-level structure and the presence of asynchronous interactions and indeterminacy. One of the formal tools used to study such processes, is the apparatus of Petri nets [1].

Petri nets (PN) are an illustrative and well formalized model, designed for the purposes of parallel systems study, which in a compact form reflect the interactions between the system elements and their dynamics. The level of model abstraction is high and it is placed between finite automata and Turing's machine.

Besides a model, a mathematical tool is also necessary for the investigation, which enables not only the analysis of a given variant of interactions, but also provides the possibility to synthesize new interacting processes of a specific class. The application of tensor analysis is suggested for the solution of similar problems [2]. Tensor methodology enables the investigation of complex systems in parts, the properties of one system being obtained from the simplest one – a "primitive system" of one class on the basis of a "sample" system of another class. The tensor approach proposed by G. Kron includes methods for analysis, as well as synthesis procedures [3]. Kulagin has developed and applied it for Petri nets [4]. Questions concerning the computer support of this approach for PN are the subject of the present paper.

2. Tensor methodology for PN-models

The tensor approach of Kron assumes the use of a group of transformations that correspond to a geometry of a certain type. In order to define the geometric transformations on interacting structures, represented by Petri nets (PN), some

* The investigations are partially supported by the National Scientific Fund, contract No И-703/97.

denotations are introduced.

Elementary net – defined in [5] and representing a transition with one input and one output position.

Primitive PN – formed by a set of elementary nets not connected one to another.

Space of PN-structures – space *Rm* with the respective dimension, a set of operations *G* and a system of coordinates (defined by Kulagin [4]), in which the output model is represented.

In [4] the type of the main tensor equations for PN is given, and also a method for PN-models study on the basis of a primitive system (PS). The disadvantage of the approach described is that for each PN, interpreting any interaction, the construction of separate tensors of the transformation is necessary. Kulagin proposes their avoiding, constructing a tensor of the transformation (TT) in the primitive system of one generalized PN, defined by him as a reduced net. It consists of a set of linear fragments (LFS) and the stage of its obtaining coincides with the stage of analysis of the interpreted interaction [6]. Realizing unions on the primitive system elements, the initial PN model can be constructed, as well as a set of other PN-models from the given class of interactions (synthesis stage). The union operations belong to the set *G* and the NP models synthesized will belong to the space *Rm* of NP-structures located in a coordinate system connected with the primitive system. The image of the new structures in the output system of coordinates is done with the help of a transformation tensor *C*, as shown in Fig. 1 (according to [6]).

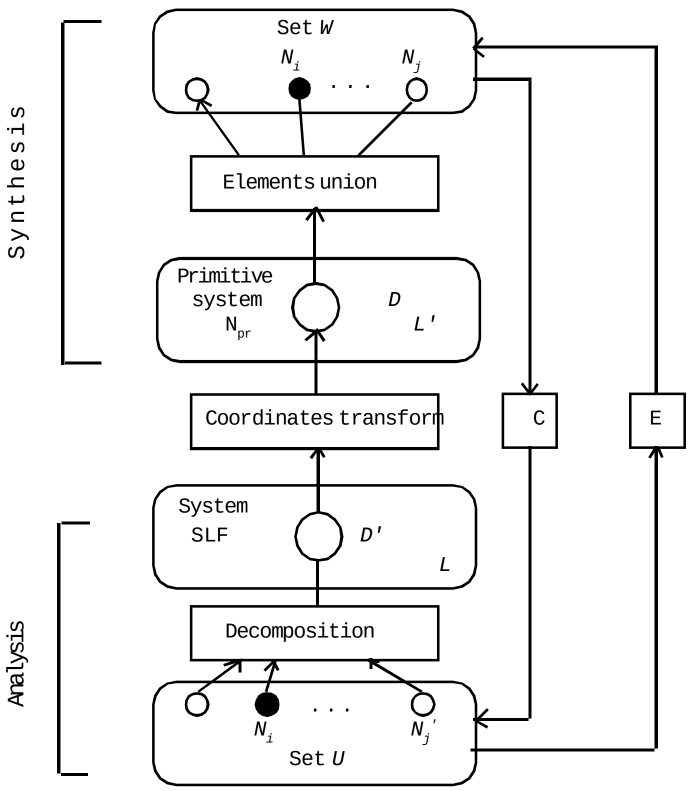

Fig. 1. NP structures transformation

The set U in Fig. 1 indicates the set of NP-structures in the output system of coordinates (in tensor analysis denotations), and W – the set of NP-structures in a coordinate system of the primitive system R_m; D and D' denote the matrices of incidence of the corresponding systems.

3. Functional scheme of the software realization

Following the logic of the approach formulated by Kulagin [6] a functional software structure could be suggested consisting of five basic blocks implementing separate problems in the aspect of [4] – Fig. 2. Each block will comprise procedures which will call some basic functions for their execution.

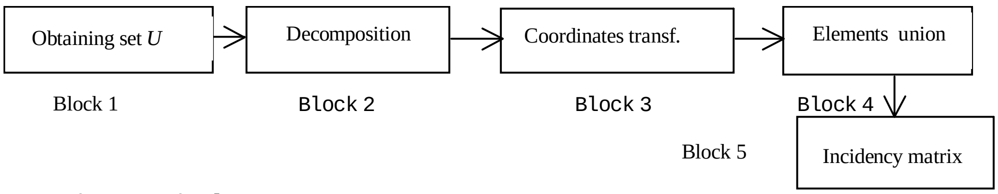

Fig. 2. Functional structure

The Block 1 will transform the PN-model from an ordinary Petri net into a basic PN (BPN) [7] representing the set U. It must contain the procedures:

– removing loops;

– removing the forbidden situations for start-stop positions;

– check of the elementary transitions.

The Block 2 realizes the decomposition of the BPN to a LFS system. The necessary procedures are:

– definition of the nodes that have to be divided.

With necessary functions:

– realization of a help operation Sum;

– realization of a help operation Num;

– division of the nodes.

With necessary functions:

– realization of the operations

Tear_p1, Tear_p2, Tear_p3v and Tear_p3h;

Tear_t1, Tear_t2, Tear_t3v and Tear_t3h;

– arranging according to LFS,

– loops breaking.

As a result of these transformations the number h of LFS is obtained, the number k_i of the transitions into each fragment, the dimension of the configured space R.

The Block 3 realizes the stage of coordinates transformation with the help of the following procedures:

– constructing the incidence matrix of the primitive system;

– computing the coefficients of the transforming tensor.

Necessary functions:

– computing the coefficients C_{Fg} for LFS with a number of transitions g;

– composition of coefficients C_q.

The synthesized matrix is computed in Block 4, joining nodes with the help of the following procedures:

– check of the restricting conditions.

With necessary functions:

– constructing the line A_w;

– determining the number of LFS, to which the node belongs;
– joining nodes.

With necessary functions:

– execution of the operations

Con_p (q, r);
Con_t(q, r):

As a result matrix $D_{new,T}$ is obtained.

The last Block 5 is designed to give a description of the PN synthesized. The necessary procedures are:

– transition towards the new coordinate system.

With necessary functions:

– standard mathematical operations (multiplying the coefficients of TT by the elements of $D_{new,T}$).

– cancellation of equivalent nodes.

With necessary functions:

– discovering coinciding rows;
– discovering coinciding columns.

As a result a coincidence matrix of the synthesized model will be obtained.

4. Modelling capacity. Using the matrix of arcs weights

The model thus obtained will be of the class of ordinary and homogeneous PN. This is a sequence of the requirement for completeness of the operations of analysis and

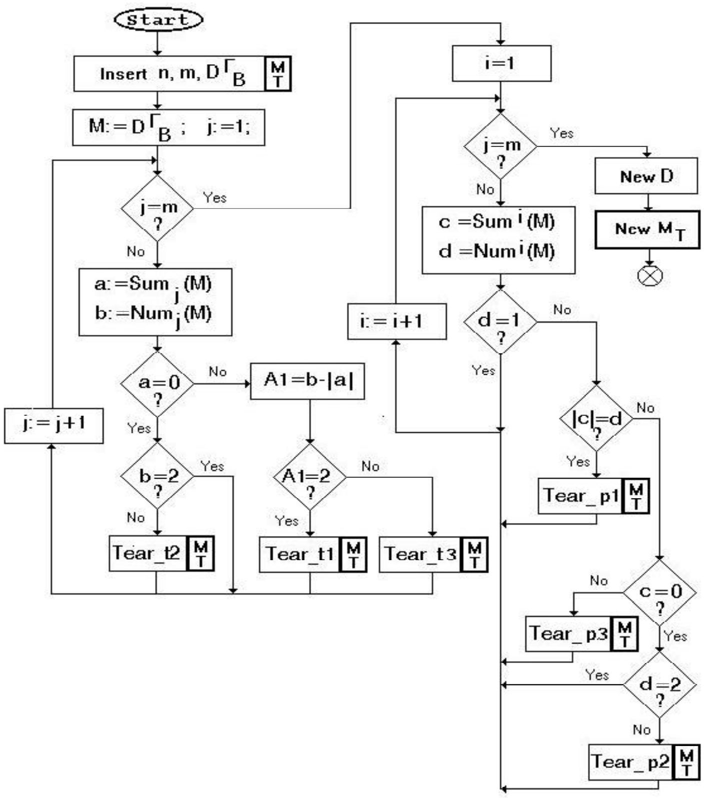

Fig. 3. Functional scheme of Block 2 (with a MW)

synthesis [8], which implies the exclusion of the multi-arcs in BPN and the model of the generalized PM is reduced to a simple PN.

The problem for analysis and synthesis of models of generalized PN is studied in [9]. For this purpose the notion matrix of arcs weights (MW) has been introduced and its properties and the corresponding equations have been investigated.

Some additions to the functional scheme have to be introduced in order to obtain models of generalized PN. The use of the matrix of arcs weights does not alter the method, it only requires each operation executed on the matrix for BFS, to be executed for MW as well. This is demonstrated in Fig. 3 with the example of Block 2. Naturally, the first procedure for Block 1 has to be the obtaining of MW, and the last procedure in Block 5 – the accounting of arcs weights (according to the equations in [9]).

The connection between the blocks and the researcher will be realized by user's interface. It is convenient to have an additional block with the task to visualize graphically the PN.

We would like to note that though the algorithm description is oriented towards matrix operations, the matrices storing and their processing as list structures may prove more suitable for large dimension PN .

5. Conclusion

The time complexity of the operations division and uniting is not greater for a polynom of second order than the number of nodes of the initial PN, and hence the software realization will have sufficient speed on personal computers also.

References

1. Peterson, J. L. Petri nets. - Computing Surveys, Vol. 9, September, No 3, 1977, 225-252.
2. Peterson, J. Theory of PN and Systems Modelling. M., Mir, 1984, p. 264.
3. Petrov, A. E. Tensor Methodology in Systems Theory. M., Radio I sviaz, 1985, p. 152.
4. Kron, G. Tensor analysis of networks. N.Y., J. Wiley & Sons, 1965, p. 720.
5. Kotov, B., E. Petri Nets. M., Nauka, 1984. p.160 (in Russian).
6. Vashkevich, N. P., V. P. Kulagin. Tensor analysis of computing networks models. Theses of reports of Federal Conference. Riga, 1987, 239-243 (in Russian).
7. Kulagin, V. P. Algebra of network models describing parallel computing structures. - Automation and Modern Technology, 1993, No 2, 25-30 (in Russian).
8. Tashev, T. D., M. B. Marinov. Basic structures and operations in tensor transformations of PN. – In: Proc. of Bulgarian-Russian Seminar "Methods and Algorithms for Distributed Information Systems Design, Theory and Applications. May, 28-20, 1996, Sofia, Bulgaria,152-162.
9. Kulagin, V. P. Following tensor technology in the design of a concurrent computing system. - In: CAD-94 - New Information Technology for Science. Education, Medicine and Business. Rousija, Yalta (Gurzuf), May 4-13, 1994. M., 87-89.
10. Tashev, T. D., H. R. Hristov. Introduction of the matrix of arcs weights in tensor transformations of PN. - In: Proc. of Bulgarian-Russian seminar "Methods and algorithms for distributed information systems design. Theory and Applications. Sept. 29- Oct. 5, 1997. Sofia, Bulgaria, 121-132.

Компьютерно-дополненый синтез моделей взаимодействующих процессов

Христо Христов, Ташо Ташев

Институт информационных технологий, 1113 София

(Р е з ю м е)

Рассматривается общая функциональная структура программной системы для синтеза моделей взаимодействующих процессов с заданными свойствами. Модели описаны при помощи аппарата сетей Петри. Определяются основные блоки системы и процедуры, необходимы для их конструирования. Синтез основается на тензорное преобразование сетевых структур.