ON SIMILARITY OF PETRI NETS LANGUAGES

© **Elena Lukyanova**

Taurida National V. I. Vernadsky University

e-mail: *lukyanovaea@mail.ru*

Abstract. *The concept of languages similarity of Petri nets is introduced. It is determined, that mapping of languages similarity of Petri nets is a surjective homomorphism. The similarity of languages of component Petri net and original detailed Petri model of the investigated parallel distributed system is considered. The work reveals that the language of the original detailed Petri net model can always be restored using the language of its component model.*

Introduction

Currently, the development of research in the field of theoretical computer science is caused by the necessity of development of formal methods of modeling and analysis of parallel distributed systems having complex structural organization and operating in real time. The establishment of adequate systems of this type is not a trivial task. The solution to this problem depends on the nature of the problems under consideration, class of simulated systems, the level of their structure detailing and behavior, and requires complex fundamental research of various formal methods and tools. Petri nets are one of the most popular and convenient modern formalisms for modeling and analysis of parallel distributed systems. This formalism has several important advantages, such as visibility, availability of simple structures to describe concurrency structures (sequential composition, choice, parallel merging) and the solubility of many behavioral properties [1, 2]. Petri nets allow, with sufficient detailing level, to model the computational processes, management processes in parallel systems and communication protocols. The main advantage of Petri nets is the ability to display the interaction of multiple parallel sequential processes as a single structure. This formalism has several drawbacks. High ability of Petri nets modeling and complexity of the simulated systems can lead to larger nets [3], [4] and as a result — to the "state explosion" problem [4]. Petri nets do not describe explicitly the dynamics of states change (behavior), and in analyzing the behavior of Petri nets we have to simultaneously monitor the situation and several points to remember these situations. In the case of errors localization, the route (path) to error site is not indicated. These circumstances are essential for the analysis of Petri nets, errors identifying and eliminating in the real system. In this connection there is a need to find trails that lead to the suspicious or erroneous state in the net operation. Such an analysis is logical to perform by constructing the relevant languages. As for significant reduction of verification efficiency in this formalism due to the "state explosion" problem, preliminary reduction of Petri net is required, which models system. The way to solve the problems formulated is in studying languages of component Petri nets (CN-net) [5, 6] and the establishment of links between languages of the original detailed Petri net model and the reduced CN model — component Petri net. The purpose of this work is to continue studies [7, 8, 9] of language connections of detailed and component Petri models of parallel distributed system and to establish the possibility of language recovery of detailed Petri net model of investigated parallel distributed system by its reduced model language — component Petri net.

1. Preliminary information

Component Petri net, introduced in works [5, 6], is a directed graph, described by the ordering quinary:

$$CN = (P, T, F, W, M_0),$$

where $P = P_1 \cup P_2$ is a finite set of places (P_1 is a finite set of component-places, P_2 — a finite set of places that are left after the separation of component-places); $T = T_1 \cup T_2$ — final set of transitions (T_1 is a finite set of components-transitions and T_2 — a finite set of transitions that are left after the separation of the component transitions); $F \subseteq P \times T \cup T \times P$ — the incidence relation between places and transitions; $W : F \to N \setminus \{0\}$ — the multiplicity function of arcs; M_0 — the initial marking of net.

It is stated in work [10] that allocation procedure in the initial detailed Petri net model of the system under consideration with concurrency of composite components (component places C_p and component transitions C_t) is a structural transformation that can significantly reduce the number of nodes of the net N while preserving its behavioral properties. This means that CN-net, built as a result of such transformations, is adequate, and hence preserving the description expressiveness of the original system under consideration. The proof of the correctness of such transformations is justified by defining component χ_1 ratio at the set of nodes of reachable markings of detailed Petri net model [11], establishing homomorphism of graphs of reachable markings of investigated Petri N and CN models, and proof of bisimular equivalence of N and CNnets.

In works [7, 8, 9] the following languages of component Petri net were introduced: language $L_t(CN)$ of component Petri net containing only components-transitions C_t, language $L_p(CN)$ of component Petri net containing only components-places C_p, language $L_{p,t}(CN)$ of component Petri net containing components-places C_p and components-transitions C_t. In this case when determining the languages $L_p(CN)$ and $L_{p,t}(CN)$, operation of nets N and CN is described in terms of the set of net reachable markings, and in determining the language $L_t(CN)$ — in terms of sequences of transitions firing. It is connected with the structures of respective composite components [12] and with the fact that composite component information, accumulated in the nodes of the component net, should be reflected in the words of the language of the corresponding component net.

2. Similarity of Petri nets languages

An important concept of the theory of formal systems is the notion of equivalence of behaviors. Equivalence of this type provides an opportunity to compare the parallel and distributed systems, taking into account certain aspects of their functioning. One type of behavioral equivalences for parallel systems and programs is language equivalence [13], i. e. the equivalence of languages, generated by systems. Language equivalence allows us to compare the behavior of both serial and parallel systems. Analytical representation is convenient for Petri net models of these systems, using a formula in algebra nets [14, 15, 16]. Net formula is constructed from symbols that define some basic net from net operations. With the help of these operations, the net described is built from elementary nets. In this way it is possible to verify the equality or inclusion of generated languages [1]. And what if languages are 'similar'? What does it mean — "similar", by how much?

For languages of Petri nets we introduce the concept of similarity of languages.

Definition 1. Similarity of Petri nets languages is understood as such transformation of Petri nets languages, defined over the same alphabet, which allows recovering one Petri net language by means of language of the other.

Statement 1. *Languages $L_t(N)$ and $L_t(CN)$ are similar.*

Argument. Consider languages $L_t(N)$ and $L_t(CN)$ [7] of some Petri net N and its component CN-net in which only components-transitions C_t are allocated, respectively, over a finite alphabets A and B (let's recall that the functioning of nets N and CN, when allocating only component-transitions, is described in terms of sequences of firing transitions). Then A^* is a set of all words in the alphabet A, $B^* = (A \cup \{T_1^*, T_2^*, ...T_n^*\})^*$ — a set of all words in the alphabet $B = A \cup \{T_1^*, T_2^*, ...T_n^*\}$, where T_k^* $(k = 1, 2, ...n)$ are the names of the various components-transitions C_{t_k} $(k = 1, 2, ...n)$ in the CN-net.

Let some word $\tau \in A^*$ have a form $\tau = abt_1t_2cdt_3t_4h$, where the symbols a, b, c, d, h denote the names of transitions of detailed model N, outside of any components-transitions C_t, and symbols t_1, t_2, t_3, t_4 are the names of transitions, which are elements of the components-transitions C_t. Making notations in the word τ:

$$ab = \tau_1, t_1t_2 = \overline{\tau}_1, cd = \tau_2, t_3t_4 = \overline{\tau}_2, h = \tau_3,$$

we have a record of the original word τ as a concatenation of the words $\tau_1, \overline{\tau}_1, \tau_2, \overline{\tau}_2, \tau_3$, so that $\tau = \tau_1\overline{\tau}_1\tau_2\overline{\tau}_2\tau_3$. In the transition from words of the language $L_t(N)$ to the words of the language $L_t(CN)$, the word τ is converted into word $\tau^{'}$ from B*: $\tau^{'}$=$\tau_1 T_1^* \tau_2 T_2^* \tau_3$. At the same time, for the names T_1^* and T_2^* of the component net CN, their record is known as subwords of words of language $L_t(N)$. Taking into account [7], that the language, as described in terms of sequences of transition firing, of identical and single-type components is congruent, it is enough to remember the word(s) of one representative from identical and single-type components to substitute its record instead of the appropriate symbol for the component-transition in words of component CN net language and get the words of original detailed Petri N model. Languages $L_t(N)$ and $L_t(CN)$ are similar.

Let's consider free languages L and $L^{'}$ of two Petri nets N and $N^{'}$, over the same alphabet W. Let this alphabet represents the grouping of the alphabets A and B, respectively, of the languages L and $L^{'}$ under consideration. And also let there be mapping "onto" of one language onto another, for example, L onto $L^{'}$. Let's mark this mapping by S. Then for each word $\tau^{'} \in L^{'}$ there should be a word $\tau \in L$ so that the equation $S(\tau) = \tau^{'}$ takes place. And because words of the language L ($L^{'}$) are written as a sequence of characters of the corresponding alphabet A (B), the mapping S generates mapping ϕ that translates the characters of each word of the language L into the characters of words of the language $L^{'}$. Given that the original mapping is "onto", mapping, then mapping ϕ is also "onto" mapping. Then when mapping ϕ the image of each character in the alphabet A (letter of the alphabet B) has at least one prototype in the alphabet A. This means that some of the letters of the alphabet B may be images of several letters of the alphabet A. Then, having the words of the language $L^{'} \subset B^*$ (B^* is a set of all words in the alphabet B) and knowing the prototypes of the letters from B being the letters of words of the language L, you can always restore the word from A^* (A^* is a set of all words in the alphabet A). And this means that the considered language L of Petri net N. Theorem 1 holds:

Theorem 1. *Similarity of Petri nets languages is surjective mapping.*

Thus, the mapping S of words in the language L of Petri net N on the set of words in language $L^{'}$ of Petri net $N^{'}$ is completely determined by the values on the letters of the alphabet W so, that each character $a \in A$ is an image of at least one character $b \in B$, that is, at mapping S for any $b \in B$ there is $a \in A$ so that $b = \phi(a)$. Then we can draw the following conclusions regarding mapping S:

1. $S(\tau\mu) = S(\tau)S(\mu)$ holds for all words τ and μ in concatenation of word $\tau\mu$ over $A \subset W$;
2. $S(e) = e$, where e is an empty word;

3. $S(\tau) = \phi(a_1)\phi(a_2)...\phi(a_k)$ for words $\tau \in A^*$ of any length. Then for $L(N)$ of net N language and $L^{'}(N^{'})$ of net $N^{'}$ language such equation is true: $L^{'}(N^{'}) = S(L(N)) = \{\tau^{'}/\tau^{'} = S(\tau),\ \ where\ \ \tau \in L(N)\}$.

Theorem 2 follows.

Theorem 2. *Similarity of Petri nets languages is a homomorphism.*

3. Similarity of languages of component Petri net, containing components-places among the allocated composite components

Consider the possibility of language recovery of detailed Petri net model by its component Petri net language, which contains components-places C_p among its constituent components. This can be a component Petri net, containing only the components-places, or component Petri net, containing both types of components: components-transitions C_t and components-places C_p. In this case, when among the constituent components of the net components-places are allocated, operation of the net has to be described in terms of the set of net reachable markings [8, 9].

Statement 2. *Languages $L_{p,t}(N)$ and $L_{p,t}(CN)$ are similar. Languages $L_p(N)$ and $L_p(CN)$ are similar.*

Argument. Consider only one variant of net. Let it be a net with both types of constituent components. Words of the language $L_{p,t}(N)$ of detailed Petri net model, with allocated constituent components C_p and C_t, and the language $L_{p,t}(N)$ of its component Petri model, represent sequences obtained by writing out symbols of nodes along the paths in the graph of reachable markings of respectively nets N and CN, starting at the initial marking and leading to each reachable net markup. Let A be a finite alphabet for the detailed model N language. It consists of a set of names, for example, s-dimensional vectors. Let B be a finite alphabet of component net CN with two types of composite components consisting of a set of names, for example, r-dimensional vectors. Then $r = s-k+l$, where k is total number of places occurred in the allocated components, l is a number of components-locations. Then A^* is a set of all words in the alphabet A, $B^* = (\psi(A) \cup \{a_1^{'},\ a_2^{'}, \ldots, a_n^{'}\})^*$ is a set of all words in the alphabet $B = \psi(A) \cup \{a_1^{'}, a_2^{'}, \ldots, a_n^{'}\}$, where $a_k^{'}(k = 1, 2, ..., n)$ are the names of the nodes of the graph of reachable markings of component Petri net CN, in which the nodes have moved or different parts of the graph of reachable markings of detailed Petri net N encapsulated. Hereat ψ — mapping that converts s-dimensional vectors of the graph of reachable markings of detailed Petri net in the r-dimensional vectors of the graph of reachable markings of component Petri net is surjective mapping [9].

Consider a word $\tau \in L_{p,t}(N)$. Let the word be of the form $\tau = a_1b_1b_2a_1a_3b_1^{'}b_2^{'}a_4$. Symbols a_1, a_1, a_3, a_4 mark the names of nodes of the graph of reachable markings of detailed net N, which are not nodes of any sections of the net, reflecting the operation of composite components. Symbols $b_1, b_2, b_1^{'}, b_2^{'}$ are the names of nodes of the graph of reachable markings of detailed net N, which are nodes of such sections. At the transition from words of the language $L_{p,t}(N)$ to the words of the language $L_{p,t}(CN)$, the word τ is converted in the word $\tau^{'}$ from B^*: $\tau^{'} = \psi(a_1)a_1^{'}\psi(a_2)\psi(a_3)a_2^{'}\psi(a_4)$. In the word $\tau^{'}$, symbol $\psi(a_i)$ $(i = 1, 2, 3, 4)$ denotes the image of the corresponding node $a_i$of the graph of reachable markings of net N, which is not a node of any of the sections of the net, reflecting the operation of the composite component. This image is determined in a one-to-one manner. Symbol $a_j^{'}$ $(j = 1, 2, 3)$ in the word $\tau^{'}$ denotes the name of the node-encapsulent. Such a node is the image of all nodes from the sections of the graph of reachable markings of detailed Petri net N, which reflects the dynamics of the functioning of the composite components. For the names $a_j^{'}$ of the language $L_{p,t}(CN)$ of the component net CN, their record as subwords of words of the language $L_{p,t}(N)$ is known by the construction of net component. Then, knowing all image prototypes of characters of any word from $L_{p,t}(N)$, the language $L_{p,t}(N)$ is easy to recover according to the words of language $L_{p,t}(CN)$ and get the language of original detailed Petri net model N, with allocated constituent components (components-places and components-transitions). Languages $L_{p,t}(N)$ and $L_{p,t}(CN)$ are similar. To establish the similarity of languages $L_p(CN)$ and $L_p(CN)$, the argument is similar.

Conclusion

When modeling thoroughly functioning of parallel distributed systems, we have to deal with so-called problem of "state explosion", when the full system model becomes immensely large. This is the problem of building detailed models of real systems. Application of the component Petri nets for modeling of parallel distributed systems allows us to build smaller — reduced models. Study of languages of such networks allows us to investigate their behavioral properties. Proceeding with the problem of how "similar" languages of detailed model of the system under consideration and its component model are, we show that the language of the reduced model (component Petri nets) can restore the language of detailed Petri net model of the system in question. Languages of detailed and component models of parallel distributed systems are similar. The concept of language similarity of Petri nets, introduced in this work, allows to determine surjective homomorphism of the languages of such networks, and on this basis to carry out the qualitative analysis of the considered Petri nets languages.

References

1. Kotov, V. 1984. *Petri nets*. Moscow: Nauka.
2. Peterson, J. L. 1977. Petri nets. *ACM Computing Surveys (CSUR)*, 9 (3), pp. 223–252.
3. Kryvy, S. and Matveeva, L. 2003. Formal Methods of Analysis of the Systems Properties. *Cybernetics and Systems Analysis*, 2, pp. 15–36.
4. Esparza, J. and Heljanko, K. 2008. *Unfoldings: A Partial-Order Approach to Model Checking. EATCS Monographs in Theoretical Computer Science*, ISBN: 978-3-540-77425-9, Springer-Verlag.
5. Lukyanova, E. 2012. On component modeling of systems with concurrency. *Naukovi Zapysky of NaUKMA. Computer Science*, t. 138, pp. 47–52.
6. Lukyanova, E. and Dereza, A. 2012. The study of the structural elements of the single-type *CN*-network during the component modeling and analysis of complex systems with concurrency. *Cybernetics and Systems Analysis*, 6, pp. 20–29.
7. Lukyanova, E. 2012. On the relationship between the language of CN-Model with component junctions, and the language of detailed Petri Model of parallel distributed system. *Visnyk of Kyiv Univ. im. Tarasa Shevchenka*, 4, pp. 145–150.
8. Lukyanova, E. 2013. Component modeling: on connections of detailed Petri model and component model of parallel distributed system. *ITHEA*, 2, pp. 15–22.
9. Lukyanova, E. 2013. On language of component Petri net with the components-places and components-transitions. *Bionics of intelligence*, 4.
10. Lukyanova, E. 2013. On Bisimulational Equivalence of Detailed Petri Net and Its CN-model of the Investigated Parallel Distributed System. *Visnyk of Kyiv Univ. im. Tarasa Shevchenka, special issue.*
11. Lukyanova, E. 2014. On homomorphism of component Petri net. *Cybernetics and Systems Analysis*, 1.
12. Lukyanova, E. 2012. The structural elements of a Petri net component. *Problemy programuvannya*, 2–3, pp. 25–32.
13. Jancar, P., Kucera, P. and Moller, F. 2000. USimulation and Bisimulation over One-Counter Processes. *Proc. of STACS'2000, 1770 of LNCS*, Springer, pp. 334–345.
14. Kotov, V. 1980. Algebra of Regular Networks. *Cybernetics*, 5, pp. 10–18.
15. Kotov, V. 1978. An algebra for parallelism based on Petri nets. *Computer Science*, 64, pp. 39–55.
16. Kapitonova, Yu. and Letichevsky, A. 1988. *Mathematical Theory of Computer Systems Designing.* Moscow: Nauka.