Computers and Artificial Intelligence, Vol. 10, 1991, No. 6, 541—560

AN ALGEBRA OF REGULAR MACRONETS FOR FORMAL SPECIFICATION OF COMMUNICATION PROTOCOLS

Nikolay A. Anisimov

Institute of Automation and Control Processes
Far East Division of the USSR Academy of Sciences, 5 Radio Str.,
Vladivostok, 690032, USSR

Abstract. The paper addresses the formal model for formal description techniques of communication protocols based on Petri nets. As a starting point we take an algebra of regular Petri nets (RPN) proposed by V. Kotov. This algebra is generalized in order to enhance an adequacy of protocol representation. A generalization of RPN called regular macronets (RMN) is described. We aso define a set of operations on macronets which are generalizations of those in RPN. In particular, in order to describe such constructions as reset, restart, disconnection, etc., we introduce a new net operation called disruption. The notion of bisimulation equivalence is reformulated for RMNs. It is shown that this equivalence is a congruence with respect to all net operations. Finally, we give an informal comparison of our approach with related ones. We illustrate an application of the introduced formalism with the aid of the ISO transport protocol.

Алгебра регулярных макросетей для формального описания протоколов сетей ЭВМ
Николай А. Анисимов

Резюме. Работа посвящена разработке формальной модели для средств формального описания протоколов на основе сетей Петри. В качестве отправной точки берется алгебра регулярных сетей Петри, введенная В. Котовым. Она обобщается с целью повышения адекватности при описании протоколов. Вводится обобщение регулярных сетей, названное макросетями. Определяются операции над макросетями, являющиеся обобщением соответствующих операций в алгебре регулярных сетей. На основе определения бисимуляции вводится понятие эквивалентности макросетей, которое является конгруэнтностью по отношению к введенным операциям. Дается неформальное сравнение с аналогичными подходами. Применение введенного аппарата иллюстрируется на примере описания транспортного протокола ISO.

Keywords: communication protocols, formal description techniques, Petri nets, compositionality.

CR classification descriptors: E4 [Coding and Information Theory] Formal models of communication.

1 INTRODUCTION

A design of computer network software is a very complex process which involves a sequence of stages. Most of them deal with communication protocols, namely with their design, formal specification, verification, performance prediction, implementation, etc. In order to provide the correctness of designed protocols each stage should be supported by appropriate formal techniques. One of them called Formal Description Technique (FDT) plays an important role in this process. Roughly speaking, each FDT consists of a formal model and specification language. Moreover, FDTs usually give rise to various computer aided tools which are intended for supporting a protocol design process.

At present the best-known FDTs are Estelle [17] and LOTOS [18] developed within the ISO. As formal models they use an extended state transition model and Milner's one, respectively. Despite the fact that Petri nets are widely used for the specification and verification of communication protocols [11], they are not known as a formal model of any FDT. At the same time, Petri nets enable to describe concurrency and asynchrony in a very natural and explicit way. Moreover, the Petri net theory possesses a rich scope of models and techniques [8, 27] that can be used successfuly in a protocol design process. Therefore it seems to be useful to cast Petri nets into FDT.

Our long-term goal is to develop such a FDT with corresponding computer-aided tool which would show such main features as wide use of Petri net theory results; compositional approach to protocol design, the use of standard specification languages such as Estelle and LOTOS, visualization of all design procedures by exploiting graphical facilities of Petri nets. The aim of this paper is to develop a mathematical basis of this approach. In particular, we offer an algebra of regular macronets being an extension of the algebra of regular Petri nets introduced by V. E. Kotov [20]. This algebra is intended for the description of protocol structures. The use of an algebraic approach itself enables us to design and specify protocols in a compositional manner. On the other hand, the net approach provides us with an explicit and natural notion of concurrency, graphical facilities, various methods of analysis. In this paper the main emphasis is on a formal basis of the FDT mentioned, its adequacy to protocols. Other aspects of the approach discussed concerning verification methods, proper specification languages, some specific problems of a protocol theory can be found in [3—5].

2 COMMUNICATION PROTOCOLS AND AN ALGEBRAIC APPROACH

There are two main sources of difficulties in a protocol specification by means of Petri nets. The first one is a difficulty in a formal representation of operations having parameters and variables. This problem is usually overcome by using models of High-level Petri nets (e.g. Predicate/Transition nets [12] or Coloured nets [19]). The second difficulty is the complexity of control structures of real-world protocols. They are usually defined as a set of more simple components such as phases, procedures, subprotocols which are united into the entire protocol in a special way. For the formal supporting of this approach we need some protocol composition rules which must correspond to intuitive methods of protocol design, produce clear and strict specifications, guarantee correctness. A very natural way to achieve this is to use an algebraic approach to the design and specification of communication protocols.

At present there are a number of works concerning the combining of algebraic and net approaches [26]. As a starting point we take an algebra of regular Petri nets [20]* proposed by Kotov. This algebra contains a definition of atomic nets, each atomic net consisting of a single transition together with head and tail places. Some net operations have been introduced, including a join, exclusion, parallel composition and iteration. In general, all these operations are well suited for formalizing practical methods of protocol design and specification. Indeed, they enable to build sequential, mutually exclusive, parallel and cyclic constructions.

On the other hand, the algebra of regular Petri nets has some considerable disadvantages. First, this algebra does not enable us to use multiple transitions, i.e. several transitions with the same label. In addition, it is impossible to use so called "silent" transitions to represent internal actions. Second, the combining of iteration and exclusion operations can result in constructions which do not agree with our intuition. For example, the exclusion of two nets in Fig. 1 results in an unexpected net. That is, the firing of the transition labelled *a* does not exclude the firing of *b* and vice versa. Third, the algebra of regular Petri nets has not a convenient notion of equivalence. For this reason the algebra does not possess a sufficient analytical power. And, finally, the algebra is incapable of representing such widely used protocol constructions as restart, disconnection, reset, etc., i.e. those where one procedure can disrupt the execution of another one.

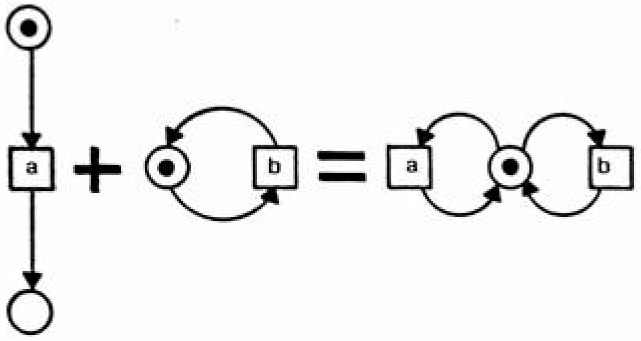

Fig. 1

The paper [2] is a first attempt to extend Kotov's algebra but it concerns only the last disadvantage. The present paper contains a more substantial extension that removes all the above restrictions.

3 BASIC DEFINITIONS AND NOTATIONS

In this section we give some definitions of nets we need and some auxiliary net operations. It is recognized (e.g. [25]) that the main drawback of net operations is that their formal definitions are somewhat cumbersome while their intuitive meaning is quite clear and obvious. One way to simplify the definitions is to use special definitions of nets, see, for example, that in [15, 34] which simplifies a definition of parallel composition. In this paper we use not one definition of nets but a set of equivalent ones, each of them being used at a suitable moment.

Let A be a set. Then A^* will denote the set of all possible words composed from elements of A. A^+ is equal to A^* without an empty word. $\boldsymbol{P}(A)$ will denote the power set of A. A concatenation of words v, $w \in A^*$ will be simply written as vw.

Let B be a set that will be called basic alphabet.

Definition 3.1. A net in P-form is a tuple $NET^P = \langle P, T \rangle$ where

(1) $P \subseteq B^+$ a finite set of places;
(2) $T \subseteq \boldsymbol{P}(P) \times B^+ \times \boldsymbol{P}(P)$ a finite set of transitions.

For $t = \langle S_1, v, S_2 \rangle \in T$ we write ${}^\bullet t = S_1$, $t^\bullet = S_2$, $\bar{t} = v$. Moreover, for each $p \in P$ we define ${}^\bullet p = \{t | p \in t^\bullet\}$, $p^\bullet = \{t | p \in {}^\bullet t\}$, $\bar{p} = p$.

Thus places and transitions are defined with the aid of the same alphabet B that is necessary only for technical reasons, which will become clear later. In other respects this definition is similar to those used in [15, 34].

Definition 3.2. A net in T-form is a tuple $NET^T = \langle P, T \rangle$ where

(1) $T \subseteq B^+$ a finite set of transitions;
(2) $P \subseteq \boldsymbol{P}(T) \times B^+ \times \boldsymbol{P}(T)$ a finite set of places.

Similarly, for each $p = \langle S_1, v, S_2 \rangle \in P$ we write ${}^\bullet p = S_1$, $p^\bullet = S_2$, $\bar{p} = v$. Moreover, ${}^\bullet t = \{p | t \in p^\bullet\}$, $t^\bullet = \{p | t \in {}^\bullet p\}$, $\bar{t} = t$ for each $t \in T$.

The two definitions are equivalent. Indeed, we can translate $NET^P = \langle P_1, T_1 \rangle$ into $NET^T = \langle P_1, T_2 \rangle$ as follows: $T_2 := T_1$, $P_2 := \{\langle {}^\bullet p, p, p^\bullet \rangle | p \in P_1 \rangle$. The inverse translation is absolutely symmetrical. Thus we are free in using one of these definitions. Moreover, we will not indicate the form of nets at all if this does not result in a confusion.

Definition 3.3. Let Σ be an alphabet and $\tau \notin \Sigma$ a distinguishable symbol. We write $\Sigma\tau = \Sigma \cup \{\tau\}$. A labelled net is a tuple $LNET = \langle NET, \sigma \rangle$ where NET is a net and $\sigma: T \to \Sigma\tau$ is a labelling function.

Now we introduce a more compact notation of nets called structured nets.

Definition 3.4. A structured net (s-net, for short) is a tuple $NET^S = \langle NET, SP, \pi \rangle$ where

(1) $SP \subseteq P$ is a set of structured places (s-places, for short) such that $\forall p \in SP$: ${}^\bullet p = \emptyset$;
(2) $\pi: SP \to \boldsymbol{P}(P - SP)$ a function that associates each s-place with a set of simple ones.

Notice s-places have no input transitions. Now we are ready to give a basic definition of net which we will use in this paper.

Definition 3.5. A structured Petri net is a tuple $N = \langle LNET^S, H, SL \rangle$ where

(1) $LNET^S$ a structured labelled net;
(2) $H \subseteq P - SP$ a set of head places;

(3) $SL \subseteq SP$ a set of tail s-places.

The class of structured Petri nets will be denoted as $\mathscr{S}$. Further if this does not result in a confusion, structured Petri nets will be referred to as nets. Moreover, we will denote these nets as a tuple $N = \langle P, T, \sigma, SP, \pi, H, SL \rangle$. When we want to stress a form of this net (P or T) we will use appropriate indices, for instance, N_1^P, N_2^T, etc. For a net N we also define a set of all tail places $L = \cup_{p \in SL} \pi(p)$.

Graphically, transitions and places will be represented as boxes and circles, each s-place $p \in SP$ containing a set of inner places $\pi(p)$. Places and transitions are connected by directed arcs. Head and tail places will be identified by extra incoming and outcoming arcs, respectively.

Thus a structured Petri net is nothing but a net plus some additional information about its initial and terminal states. We assume that an initial state of net N corresponds to the marking of all head places. We consider a net is in a terminal state if each tail s-place has at least one marked internal place.

Define a marking of net as a function $M: (P - SP) \to \{0, 1, 2, \ldots, \}$, i.e. an allocation of tokens among simple places. Here we do not define an internal marking M_0 explicitly and assume it is equal to $M_0 = \{(p, 1) | p \in H\} \cup \{(p, 0) | p \notin H\}$. We assume a net has conventional firing rules [28] which are omitted here. As for the manipulation with s-places they will be explained a little later. We write $M[t\rangle M'$ when a transition t occurs at the marking M to yield a new marking M'. If $v = t_1 t_2 \ldots t_n \in T^*$, then $M[v\rangle M'$ means that there exist $M_1, M_2, \ldots, M_n$ such that $M[t_1\rangle M_1 [t_2\rangle \ldots M_{n-1} [t_n\rangle M'$. M' is said to be reachable from M. $M[\rangle M'$ means that there exists $v \in T^*$ such that $M[v\rangle M'$. The set of all reachable markings from M is defined below as $[M\rangle = \{M' | M[\rangle M'\}$.

It is intuitively clear that a choice of a basic alphabet and encodings of places and transitions (P and T) do not affect the behaviour aspects of nets. Therefore we will define net up to its isomorphic class.

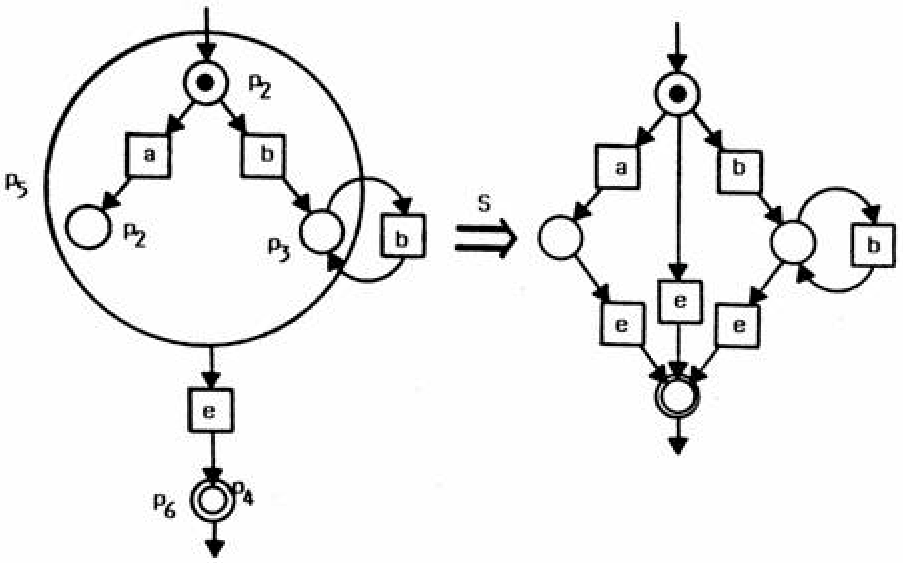

Fig. 2. S-normalization

Definition 3.6. Let N_1, N_2 be nets in P-form which are defined with the aid of basic alphabets B_1 and B_2, respectively. Then N_1 is isomorphic to N_2 ($N_1 \equiv N_2$) iff there exists a mapping α: $B_1^+ \to B_2^+$ such that $p \in P_1 \Leftrightarrow \alpha(p) \in P_2$, $t \in T_1 \Leftrightarrow \alpha(t) = \langle \alpha({}^\bullet t),\ \alpha(t),\ \alpha(t^\bullet) \rangle \in T_2$, $\sigma_1(t) = \sigma_2(\alpha(t))$, $p \in SP_1 \Leftrightarrow \alpha(p) \in SP_2$, $\pi_1(p) = \pi_2(\alpha(p))$, $p \in H_1 \Leftrightarrow \alpha(p) \in$ $\in H_2$, $p \in SL_1 \Leftrightarrow \alpha(p) \in SL_2$. Here $\alpha(X) = \{\alpha(x) | x \in X\}$.

Now we introduce three normal representation forms of a net and corresponding normalization procedures which will be required for us.

We will consider a net N to be in a structured normal form (s-normal, for short) if all its s-places have no output transitions, i.e., $\forall p \in SP: p^\bullet = \emptyset$. The s-normalization procedure is as follows. First we remove all s-places with output transitions $p_s \in SP: p_s^\bullet \neq \emptyset$. Every transition $t \in p_s^\bullet$ is replaced by the set $\{t\bar{p} | p \in \pi(p_s)\}$ where each transition is labelled by the same label like t. Each internal place $p \in \pi(p_s)$ is also replaced by the place $\langle {}^\bullet p, \bar{p}, p^\bullet \cup \{t\bar{p} | t \in p_s^\bullet\} \rangle$. Fig. 2 contains an example of s-normalization. Here $B = \{p_1, p_2, p_3, p_4, p_5, p_6, t_1, t_2, t_3, t_4\}$ is a basic alphabet. There are two s-places p_5 and p_6 with $\pi(p_5) = \{p_1, p_2, p_3\}$, $\pi(p_6) = \{p_4\}$, place p_5 having an output transition. The s-normalization that is to "open" place p_5 results in some new transitions encoded by t_4p_1, t_4p_2, t_4p_3. Thus the s-form is nothing but an abbreviated notation of nets and, in fact, s-normalization defines firing rules for structured nets by reducing them into well-known Petri nets.

We will consider a net N to be in h-form if each its head place has no input transitions, i.e. $\forall p \in H$: ${}^\bullet p = \emptyset$. There is a procedure of h-normalization, called root-unwinding [13, 14]. For each cyclic head place, i.e. a place having input transitions $p \in H: {}^\bullet p \neq \emptyset$, we define one copy $\tilde{p}$. For each its output transition $t \in p^\bullet$ we also define one copy $\tilde{t}$ which has the same sets of input and output places except for the fact that each input cyclic place p is replaced by its copy $\tilde{p}$. The labelling of $\tilde{t}$ coincides with that of t, i.e., $\sigma(\tilde{t}) = \sigma(t)$. In a set of head places all cyclic places are replaced by their copies. Moreover, if a cyclic head place belongs to a tail set then its copy is also added to this set. Fig. 3 shows an example of h-normalization.

A net N will be considered as 1-normal if for every tail s-place $ps \in SL$ its internal set

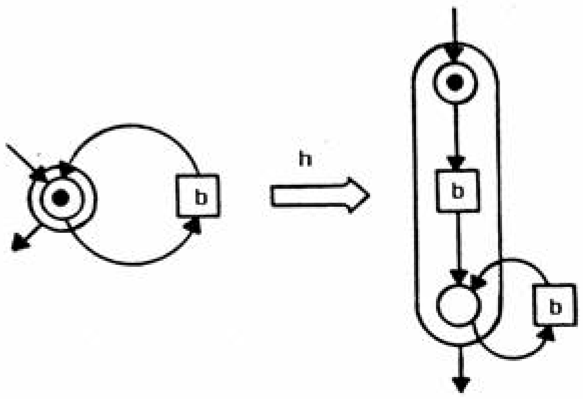

Fig. 3. H-normalization

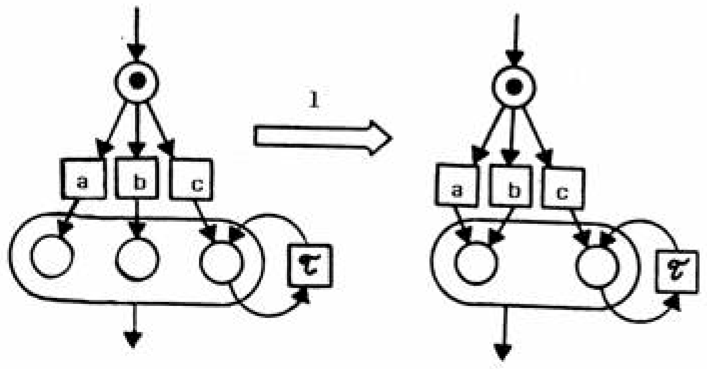

Fig. 4. L-normalization

has at most one place without output arcs, i.e., $|\{p | p \in \pi(p_s), p^\bullet \neq \emptyset\}| \leqslant 1$. If this is not the case we can 1-normalize a net by merging all such places into one, as shown in Fig. 4.

Procedures of *s*-, *h*- and 1-normalization of a net N will be denoted as $s(N)$, $h(N)$, $1(N)$. It is easy to show that $s(s(N)) \equiv s(N)$, $h(h(N)) \equiv h(N)$, $1(1(N)) \equiv 1(N)$.

4 OPERATIONS ON NETS

In this section we introduce an algebra of regular nets by extending Kotov's algebra onto our definition of nets. First of all we define a class of atomic nets.

Let $a \in \Sigma_\tau$ be a symbol. Then an atomic net which corresponds to this symbol is a net (in P-form): $N_a = \langle P, T, \sigma, SP, \pi, H, SL \rangle$ where $P = \{p_1, p_2, p_3\}$, $T = \{\langle\{p_1\}, t_1, \{p_2\}\rangle\}$, $\sigma(\langle\{p_1\}, t_1, \{p_2\}\rangle) = a$, $SP = \{p_3\}$, $\pi = \{(p_3, \{p_2\})\}$, $H = \{p_1\}$, $SL = \{p_3\}$. The graphical representation of an atomic net is shown in Fig. 5. Sometimes, we will write simply a instead of N_a.

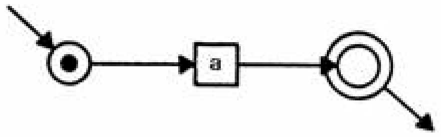

Fig. 5. Atomic net

Now we define four operations on nets which are generalizations of corresponding Kotov's operations. These are join, exclusion, iteration and parallel composition. Nets which are built with the aid of these operations and a class of atomic operations form a class of regular nets $\mathscr{R}$. If an operation of parallel composition is not used we have a class of primitive nets $\mathscr{P}$.

Now we give some preliminary notations. Let $p_1, p_2 \in P$ be places of a net N in T-form. Then a merging of these places is a new place $p_1 \otimes p_2 = \langle {}^\bullet p_1 \cup p_2^\bullet, \bar{p}_1 \bar{p}_2, p_1^\bullet \cup p_2^\bullet \rangle$. If

$P_1, P_2 \subseteq P$ then $P_1 \otimes P_2 = \{p_1 \otimes p_2 | p_1 \in P_1, p_2 \in P_2\}$. For P-form of net N we have $t_1 \otimes t_2 = \langle {}^\bullet t_1 \cup t_2^\bullet, \bar{t}_1 \bar{t}_2, t_1^\bullet \cup t_2^\bullet \rangle$ and $T_1 \otimes T_2 = \{t_1 \otimes t_2 | t_1 \in T_1, t_2 \in T_2\}$, $T_1, T_2 \subseteq T$.

Let $N = \langle P, T, \sigma, SP, \pi, H, SL \rangle$, $N_1 = \langle P_1, T_1, \sigma_1, SP_1, \pi_1, H_1, SL_1 \rangle$, $N_2 = \langle P_2, T_2, \sigma_2, SP_2, \pi_2, H_2, SL_2 \rangle$ be nets.

1. A join operation. A join of nets $N_1, N_2 \in \mathscr{R}$ is a net $N = (N_1 ; N_2)$, where N is built from N_1^T and N_2^T as follows:

$T = T_1 \cup T_2$, $\sigma = \sigma_1 \cup \sigma_2$,
$P = P_1 \cup P_2 - (SL_1 \cup L_1 \cup H_2) \cup SL_1 \otimes H_2 \cup L_1 \otimes H_2$
$SP = SP_1 \cup SP_2 - SL_1 \cup SL_1 \otimes H_2$
$\pi = \pi_1 \cup \pi_2 - \{(p, \pi_1(p)) | p \in SL_1\} \cup \{(p_1 \otimes p_2, \pi_1(p_1) \otimes p_2) | p_1 \otimes p_2 \in SL_1 \otimes H_2\}$
$H = H_1$, $SL = SL_2$.

In other words, nets are united in such a way that all tail s-places of the first net are merged with head places of the first net. It should be noted that the resulting net N is not s-normal and so we can s-normalize it. Fig. 6 shows an example of this operation. Here the first net has two tail s-places and the second one has two head places. A join of these nets results in Cartesian product of the places, four new places being generated.

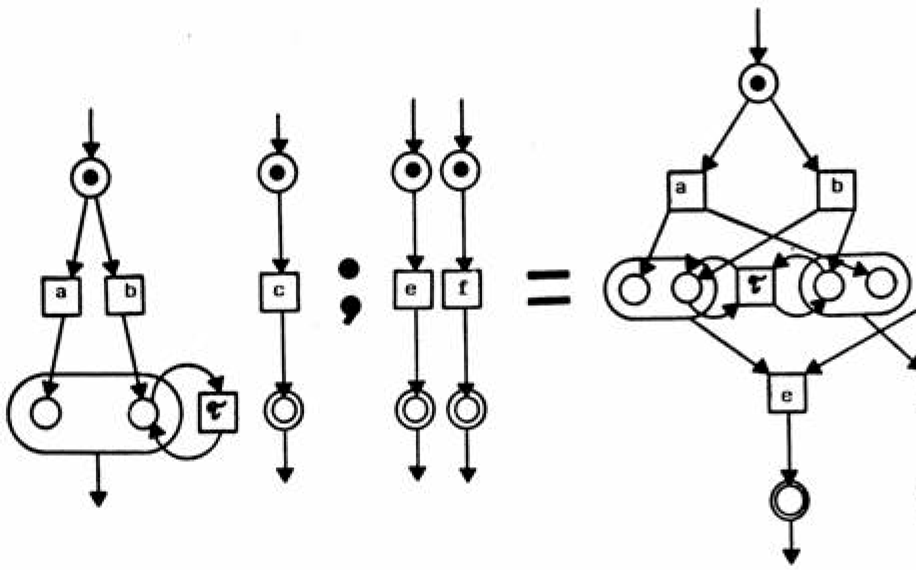

Fig. 6. Join operation

2. An exclusion operation. An exclusion of nets $N_1, N_2 \in \mathscr{R}$ is a net $(N_1 \square N_2) = I(N)$ where N is built from $h(N_1^T)$ and $h(N_2^T)$ as follows:

$T = T_1 \cup T_2$, $\sigma = \sigma_1 \cup \sigma_2$,
$P = P_1 \cup P_2 - (H_1 \cup H_2 \cup SL_1 \cup SL_2 \cup L_1 \cup L_2) \cup$
$\cup H_1 \otimes H_2 \cup SL_1 \otimes SL_2 \cup SL_1 \otimes L_2 \cup L_1 \otimes SL_2$
$SP = SP_1 \cup SP_2 - (SL_1 \cup SL_2) \cup SL_1 \otimes SL_2$
$\pi = \pi_1 \cup \pi_2 - \{(p, X) | p \in SL_1 \otimes SL_2\} \cup \{(p_1 \otimes p_2, \pi_1(p_1) \otimes \{p_2\} \cup$

$$\cup \{p_1\} \otimes \pi_2(p_2)) | p_1 \otimes p_2 \in SL_1 \otimes SL_2\}$$
$$H = H_1 \otimes H_2,\ SL = SL_1 \otimes SL_2.$$

Less formally, this operation unites two nets in such a way that their sets of head places are merged. The same is done with sets of tail s-places. Note that previously nets N_1 and N_2 are h-normalized. The obtained net N, in its turn, may not be in 1-normalized form and so we must 1-normalize it. Fig. 7 shows an example of this operation.

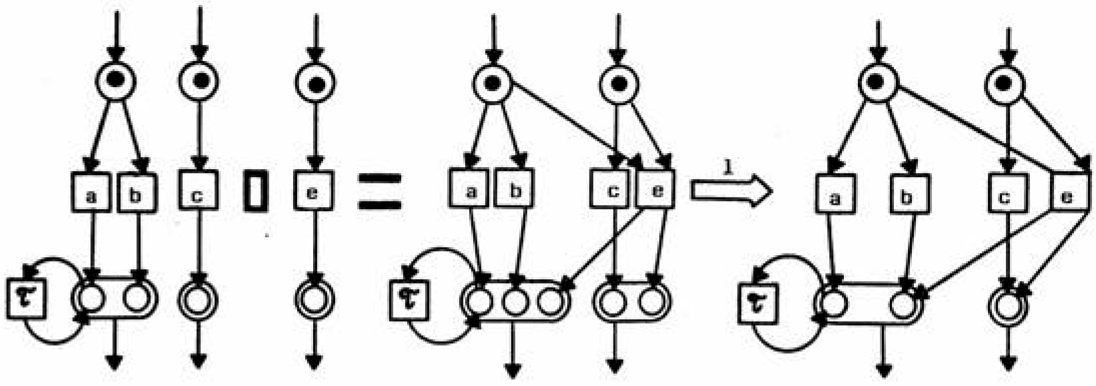

Fig. 7. Exclusion operation

3. A parallel composition. A parallel composition of nets N_1, $N_2 \in \mathscr{R}$ is a net $(N_1 \| N_2) = N$ where N is built from N_1^p and N_2^p as follows:

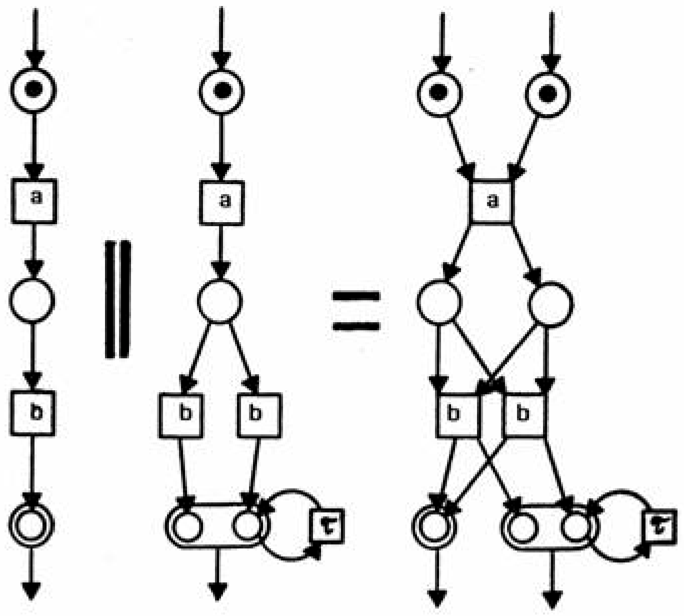

Fig. 8. Parallel composition

$T = T_1 \cup T_2 \cup T_{syn} - T_{aut}, \; T_{syn} = \{t_1 \otimes t_1 | t_1 \in T_1, \; t_2 \in T_2, \; \sigma_1(t_1) = \sigma_2(t_2) \neq \tau\}$
$T_{aut} = \{t_1, \; t_2 | t_1 \in T_1, \; t_2 \in T_2, \; \sigma_1(t_2) = \sigma_2(t_2) \neq \tau\}$
$\sigma = \sigma_1 \cup \sigma_2 - \{(t, a) | t \in T_{aut}\} \cup \{(t_1 \otimes t_2, \; \sigma_1(t_1)) | t_1 \otimes t_2 \in T_{syn}\}$
$SP = SP_1 \cup SP_2, \; \pi = \pi_1 \cup \pi_2, \; H = H_1 \cup H_2, \; SL = SL_1 \cup SL_2.$

In other words, parallel composition unites two nets by merging those transitions which are labelled by the same names. This operation differs from a similar Kotov's operation of superposition, by that it does not merge places at all. Fig. 8 contains an example of this operations.

4. *An iteration operation*. In order to avoid cumbersome definitions we give a restricted form of an iteration, namely, only for primitive nets. Nevertheless, it is sufficient for many applications. Let $N \in \mathscr{P}$ be a primitive net. It is clear that it has single head and tail places, i.e., $|H| = |SL| = 1$. Let $H = \{p_h\}$ and $SL = \{p_1\}$ be such sets. If $|\pi(p_l)| = 1$ then the net $N' = *(N)$ is obtained by merging head and tail places: $T' = T$, $\sigma' = \sigma$, $P' = P - (H \cup SL \cup L) \cup L \otimes H$, $SP' = SP - SL \cup SL \otimes H$, $\pi' = \pi - \{(p_l, \; \pi(p_l)\} \cup \cup \{(p_l \otimes p_h, \pi(p_l)\}$, $H = \pi(p_l)$, $SL = \{(p_l \otimes p_h)\}$. If $|\pi(p_l)| > 1$ then we first split the head place p_h together with its output transitions, i.e. we add a new place $\tilde{p}_h$ and a set of new transitions $\{\tilde{t} | t \in p_h^{\bullet}\}$ with $\sigma(\tilde{t}) = \sigma(t)$. After that we merge place $\tilde{p}_h$ with the tail s-place p_l in much the same way as above. Moreover, we add the place p_h to the tail set. This enables to take into account a possibility to execute the net zero times which also corresponds to our intuition of an iteration operator. Fig. 9 shows examples for both cases.

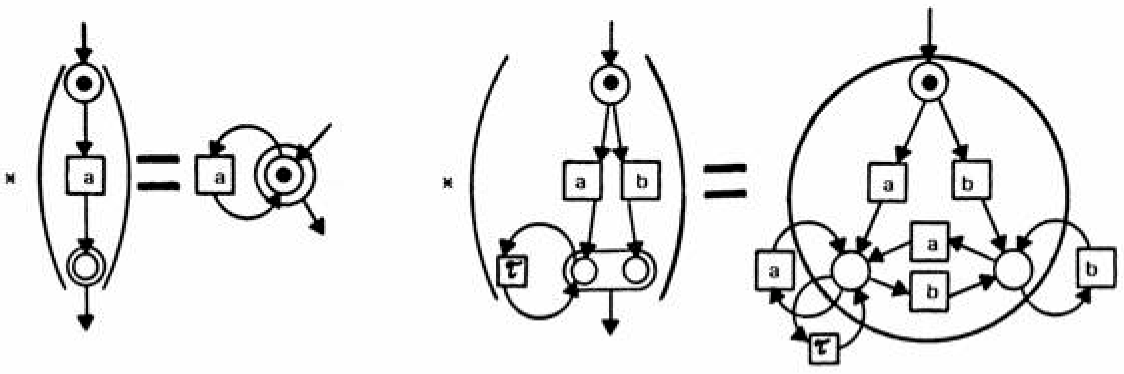

Fig. 9. Iteration operation

Thus these net operations are much more satisfactory for generally accepted intuition. Indeed, the exclusion of nets shown in Fig. 1 results in a net with wanted behaviour, see Fig. 10. Here we build a net $N = (a \square * (b))$. Once the transition a has fired, transition b cannot be enabled at all and vice versa. Moreover, an initial marking is at the terminal one. This formalizes an execution of b zero times. Fig. 2 shows a net $(N; e)$.

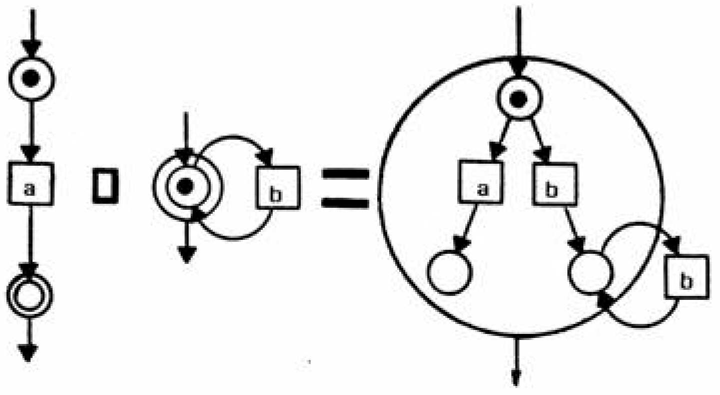

Fig. 10

5 AN INTRODUCTION TO MACRONETS

In this section we extend our algebra by introducing more general nets called macronets in order to account for disruption.

Informally, a regular macronet (RMN) is a regular net whose set of places consists of two disjoint sets of simple places and macroplaces. Each macroplace contains an own inner macronet. RMNs have following firing rules. A transition having an input macroplace is enabled if the inner macronet of this place has at least one token. The firing of a transition with an input macroplace involves removing one token from inner macronet regardless of the place where it is located. If the inner net has parallel fragments, i.e. this net was built with the aid of parallel composition, tokens are removed from each of them. If the transition has an output macroplace, tokens are put into all head places of the inner net.

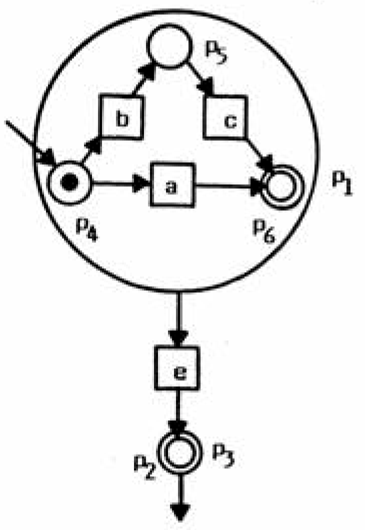

Fig. 11. Example of a macronet

An example of RMN is presented in Fig. 11. This macronet contains a single macroplace p_1 which has an inner regular net. At the beginning, transitions a and b of the internal net are enabled and can fire. At any time during execution of the internal net the transition e is also enabled. If it fires, a token, whenever it may be (p_4, p_5, p_6) is removed from the internal net and placed into p_2.

Fig. 12 contains an RMN which describes a control structure of the well-known class 2 of the ISO transport protocol [16]. This protocol is defined using the set of command names $\Sigma = \Sigma_1 \cup \Sigma_2 \cup \Sigma_3 \cup \Sigma_4 \cup \Sigma_5$ where $\Sigma_1 = \{\uparrow CR, \downarrow CR, \uparrow CC, \downarrow CC\}$, $\Sigma_2 = \{\uparrow DT, \downarrow DT, \uparrow AK, \downarrow AK\}$, $\Sigma_3 = \{\uparrow ED, \downarrow ED, \uparrow EA, \downarrow EA\}$, $\Sigma_4 = \{\uparrow DR, \downarrow DR, \uparrow DC, \downarrow DC\}$, $\Sigma_5 = \{\uparrow N\_DIS, \uparrow N\_RES\}$ are sets of command names for connection establishment, normal data transfer, expedited data transfer, disconnection and error release procedures, respectively. Unlike the original specification, this structure is slightly regularized. In particular, the procedure of connection refusal is provided by the disconnection procedure. Thus after transmitting a $\downarrow CR$ the remote entity may refuse connection using a disconnection procedure. The entity can also use a disconnection procedure before a connection establishment (head place p_1 is marked). When an entity is at data transfer phase, three

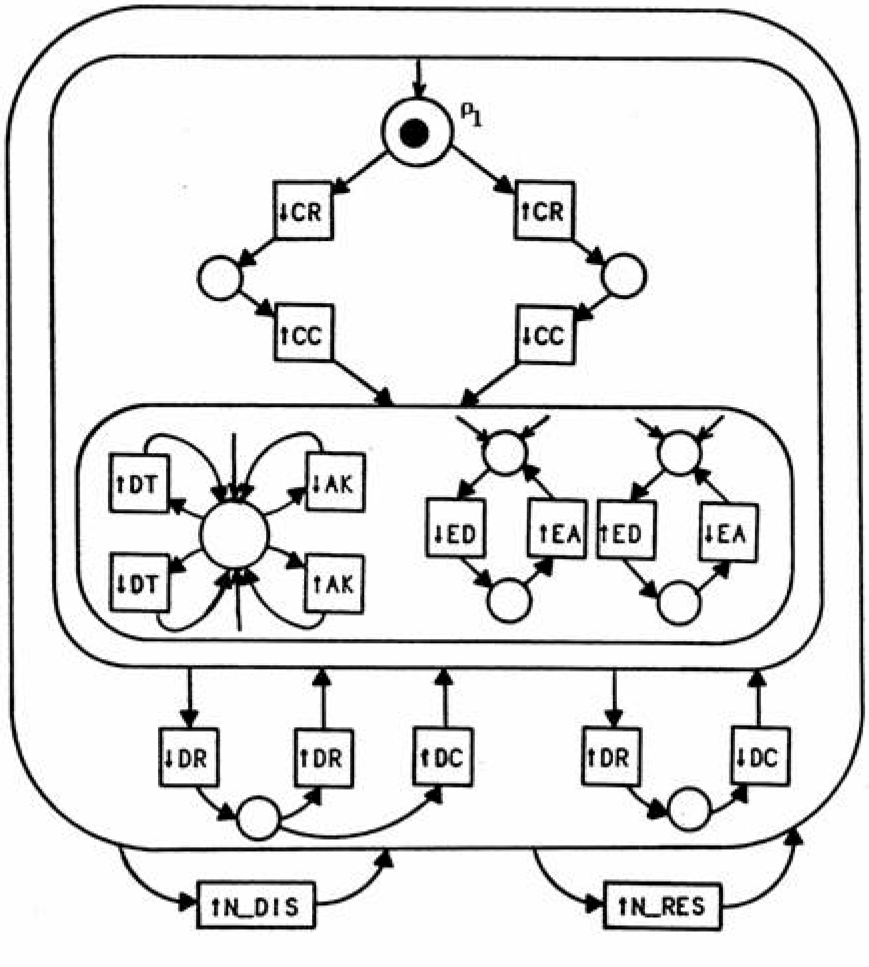

Fig. 12. Transport protocol

parallel fragments can be performed, namely normal data transfer, sending and receiving of expedited data. At any time errors from the network level can occur. In this case the execution of all procedures is disrupted by removing tokens from the internal net and putting a token into head place p_1.

6 FORMAL DEFINITION OF MACRONETS AND A DISRUPTION OPERATION

Now we give a formal definition of a regular macronet and its firing rules.

Definition 6.1. A regular macronet is a tuple $MN = \langle Q, W, N_0, \beta\rangle$ where

(1) $Q = \{N_1, N_2, \ldots, N_n\}$ a finite set of regular nets, a net having no common elements:
$$T_i \cap T_j = P_i \cap P_j = \emptyset,\ i \neq j;$$
(2) $W = \{p_1, p_2, \ldots, p_n\}$ finite set of macroplaces such that $\forall i \exists j : p_i \in P_j$;
(3) $N_0 \notin Q$ an external regular net.
(4) $\beta : W \to Q$ a function of macroplace definition.

In addition, we restrict the function β by forbidding a recursion in the definition of β. Notice a regular net is a special case of a RPN provided that $A = W = \beta = \emptyset$, $N_0 \neq \emptyset$.

Now we generalize the above operations onto RMNs. Let $op \in \{;, \Box, \|, *\}$ be a net-operation and $MN_1 = \langle Q_1, W_1, N_{01}, \beta_1\rangle$, $MN_2 = \langle Q_2, W_2, N_{02}, \beta_2\rangle$ be macronets. Then define

$$MN = (MN_1\ op\ MN_2) = \langle Q_2 \cup Q_1,\ W_1 \cup W_2,\ (N_{01}\ op\ N_{02}),\ \beta_1 \cup \beta_2\rangle,$$

i.e. the operations are defined with the aid of corresponding operations on external nets. Similarly, $*(MN) = \langle Q_1, W_1, *(N_{01}), \beta_1\rangle$.

Now we define a new operation on macronets that, unlike previous ones, results in inclusion. Let the external net N_{02} be primitive and its head place not a macroplace, i.e. $N_{02} \in \mathscr{P}$, $N_{02} \cap W = \emptyset$. Then a disruption operation of MN_1 and MN_2 builds a new macronet:

$$MN = (MN_1 \bigcirc MN_2) = \langle Q_1 \cup Q_2 \cup \{N_{01}\},\ W_1 \cup W_2 \cup H_2,$$
$$N_{02},\ \beta_1 \cup \beta_2 \cup \{(p, N_{01}) | p \in H_2\}\rangle.$$

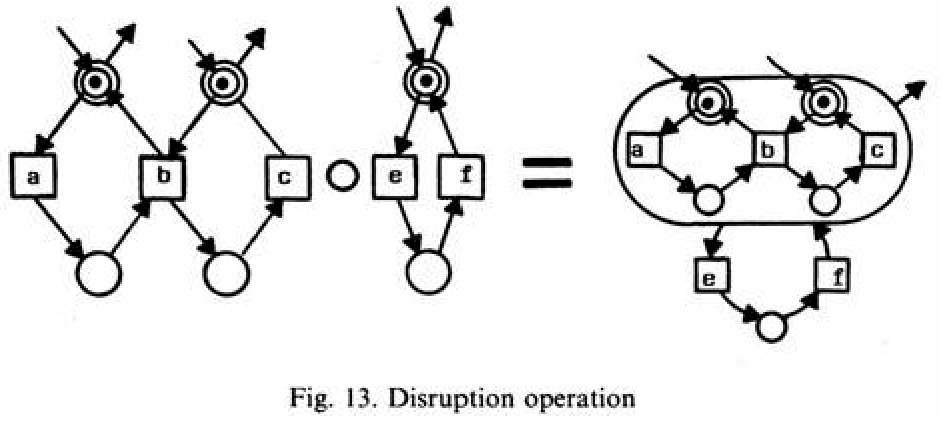

Fig. 13. Disruption operation

Fig. 13 shows an example of this operation.

Thus, macroplaces in RMN result from the use of the disruption operation. We can define the disruption operation within a class of simple nets and hence define a translation procedure of RMNs into simple nets. This operation is denoted by "●". A class of nets built using this operation is denoted by $\mathscr{D}$.

Let [illegible] $N_2 \in \mathscr{P}$ be primitive nets. Then the net $N = (N_1 \bullet N_2)$ is built as follows:

$$P = P_1 \cup P_2,\ SP = SP_1 \cup SP_2,\ \pi = \pi_1 \cup \pi_2,\ H = H_1,\ SL = SL_2\,,$$

$$T = T_1 \cup T_2 - \{t | t \in H_2{\bullet}\} \cup U_{t \in H_2\bullet} D_t(N_1), \quad \text{where } D_t(N_1) = \{\langle\{p\}, \bar{t}p, t^{\bullet}\rangle | p \in P_1\,,$$

$$\sigma = \sigma_1 \cup \sigma_2 - \{(t, \sigma_2(t)) | t \in H_2{\bullet}\} \cup \{(t', \sigma_2)) | t' \in D_t(N_1)\}\,.$$

Fig. 14 shows a net which is built from the net of Fig. 11.

Let $N_1 \in \mathscr{D}$, $N_2 \in \mathscr{P}$ be nets. Then $N = (N_1 \bullet N_2)$ is defined in the same way as above, but for $D_t(N_1)$ it is determined recursively. Let $N_1 = (N_1' \| N_1'')$. Then

$$D_t(N_1' \| N_1'') = \{\langle {}^{\bullet}t_1 \cup {}^{\bullet}t_2,\ \bar{t}_1\bar{t}_2,\ t^{\bullet}\rangle | t_1 \in D_t(N_1'),\ t_2 \in D_t(N_1'')\}\,.$$

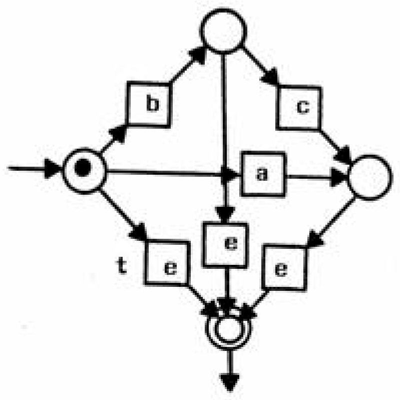

Fig. 14

For other operations, $D_t(N_1'\ op\ N_1'')$ is determined simply as a union of $D_t(N_1')$ and $D_t(N_1'')$ where all input places occurring in merging are modified correspondingly. For example, $D_t(N_1'{}^{\bullet}N_1'') = \,= D_t(N_1') \cup D_t(N_1'') - \{t' | {}^{\bullet}(t') = H_2\}$. For simplicity we omit these formal definitions. Fig. 15 shows a net which is built from the macronet of Fig. 13.

Thus for each RMN we can build a corresponding net. RMNs are an abbreviated notation of nets which belong to class $\mathscr{D}$. Undoubtedly, this abbreviation is very useful. Indeed, if we "unfold" the RMN of Fig. 12, we obtain a very large net which is hard to comprehend and hence to use.

Proposition 6.2. $\mathscr{P} \subset \mathscr{R} \subset \mathscr{D} \subset \mathscr{S}$

Proof. All strict inclusions are proved by means of simple counterexamples. The first inclusion is trivial as the net $a\|b$ does not belong to primitive nets. The second one is proved by the net shown in Fig. 14 which does not belong to $\mathscr{R}$. This can be stated

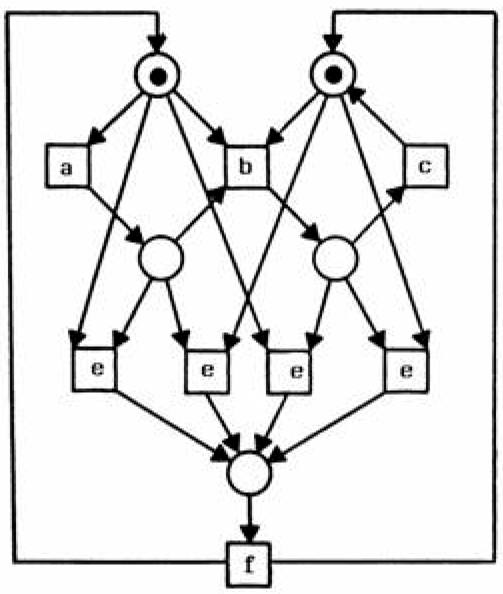

Fig. 15

directly by looking over its top-level operations. If we remove a transition t we obtain the net which does not belong to $\mathscr{D}$. □

In other words, the disruption operation does extend the algebra of regular nets and cannot be expressed in other operations. Let us state another fact which will be useful for us.

Proposition 6.3. Each net $N \in \mathscr{D}$ is safe, i.e., $\forall p \, \forall M \in [M_0\rangle$: $|M(p)| \leqslant 1$.

The proof is performed by induction over the net structure. Obviously, each atomic net N_a, $a \in \Sigma\tau$ is safe. This is a basis of the induction. Let $N_1, N_2 \in \mathscr{D}$ be safe nets. It is easy to show that all nets $(N_1 ; N_2)$, $(N_1 \square N_2)$, $(N_1 \| N_2)$, $*(N_1)$, $(N_1 \bullet N_2)$ are also safe, which directly follows from the definitions of these operations.

7 A NOTION OF EQUIVALENCE

At present a wide variety of equivalence notions for parallel processes are known, most of them being represented in terms of Petri nets [29]. At the same time, the algebra of regular Petri nets makes use of only a classic notion defined as equality of languages produced by nets. This notion seems to be weak for many applications because it does not preserve some important behaviour properties such as deadlock-freedom etc. Therefore we give a stronger notion which is based on a well-known notion of bisimulation equivalence proposed in [27] and developed in many papers [7, 14, 15, 23, 34].

First we extend the labelling function σ (see Definition 3.3) to a homomorphism σ: $T^* \to \sigma\tau^*$. If $M[t\rangle M'$ and $\sigma(t) = a$ we will also write $M[a\rangle M'$. Define σ: $T^* \to \Sigma^*$ obtained from σ by removing all τ-symbols from strings $v \in \Sigma^*$. If $w \in \Sigma^*$, then $M[w\rangle\rangle M'$ means that $\exists y \in T^*$: $M[y\rangle M'$ and $\sigma(y) = w$.

Definition 7.1. Two nets $N_1, N_2 \in \mathscr{D}$ are (weak) bisimilar equivalent ($N_1 \approx N_2$) if and only if there exists a relation $R \subseteq [M_{01}\rangle \times [M_{02}\rangle$ such that

(1) $(M_{01}, M_{02}) \in R$
(2) $(M_1, M_2) \in R \Rightarrow (M_1 [a\rangle\rangle M_1' \Rightarrow \exists M_2' : M_2 [a\rangle\rangle M_2' : (M_1', M_2') \in R)$
(3) $(M_1, M_2) \in R \Rightarrow (M_2 [a\rangle\rangle M_2' \Rightarrow \exists M_1' : M_1 [a\rangle\rangle M_1' : (M_1', M_2') \in R)$

It is easy to show that this equivalence is a congruence relation with respect to join, iteration and parallel composition operations. On the other hand, this equivalence is not a congruence with respect to operations of exclusion and disruption. For example, although $a \approx a$ and $b \approx (\gamma; b)$ but $a \Box b \not\approx a \Box (\gamma; b)$ and $a \Box b \not\approx a \bullet (\gamma, b)$. For this reason we enhance the discriminating power of bi-equivalence similarly to [7].

Definition 7.2. Two nets $N_1, N_2 \in \mathscr{D}$ are r-equivalent ($N_1 \approx_r N_2$) iff

(1) $N_1 \approx N_2$ with the relation $R \subseteq [M_{01}\rangle \times [M_{02}\rangle$;
(2) $(M_1, M_2) \in R \Rightarrow (M_1 = M_{01} \Leftrightarrow M_2 = M_{02})$.

Theorem 7.3. An r-equivalence is a congruence relation with respect to operations ;, $\Box$, $\|$, $*$, $\bullet$, i.e. if $N_1' \approx_r N_2''$ and $N_2' \approx_r N_2''$ then

(1) $(N_1'; N_2') \approx_r (N_1''; N_2'')$
(2) $(N_1' \Box N_2') \approx_r (N_1'' \Box N_2'')$
(3) $(N_1' \| N_2') \approx_r (N_1'' \| N_2'')$
(4) $*(N_1') \approx_r *(N_1'')$
(5) $(N_1' \bullet N_2') \approx_r (N_1'' \bullet N_2'')$

Proof. Let R_1 and R_2 be relations of equivalences $N_1' \approx_r N_1''$ and $N_2' \approx_r N_2''$, respectively. In order to prove each relation (1)—(5) we build a new relation R from R_1 and R_2 and show they satisfy all conditions of definitions 7.1 and 7.2. Build, for example, R for the clause (4). Due to the proposition 6.3 we can express a marking as a subset of places: $M \subseteq (P - SP)$. Let $R = R_1 \cup (R_2 - (M_{02}', M_{02}''))$. The relation is a bisimulation. Denote $N' = (N_1' \bullet N_2')$ and $N'' = (N_1'' \bullet N_2'')$. Obviously, $(M_0', M_0'') \in R$ because $M_0' = M_{01}'$ and $M_0'' = M_{01}''$. Assume $(M_1', M_1'') \in R$ and $M_1 [a\rangle\rangle M_2'$ in the net N_1'. If $M_1', M_2' \subseteq P'_1$ then due to R_1 there exists M_2'' such that $M_1''[a\rangle\rangle M_2''$. If $M_1', M_2' \subseteq P_2'$ then due to $R_2 \exists M_2''$: $M_1''[a\rangle\rangle M_2''$. Consider the third case $M_1' \subseteq P_1'$, $M_2' \subseteq P_2'\rangle$. Assume $M_1' [t'\rangle M_2'$, $\sigma'(t') = a$. Then due to the definition of disruption we have $t' \in (H_2')^\bullet$ and $\exists t'' \in (H_2)^\bullet$ such that $M_1''[t''\rangle M_2''$. Due to $N_2' \approx_r N_2''$ we have also $\exists M_2''$: $M_1''[t''\rangle M_2''$, $\sigma''(t'') = a$. If $M_1'[w\rangle M_2'$, $\sigma'(w) = a$ we can divide w into three parts $w = w_1 t w_2$ where $w_2 \in (T_1')^*$, $t \in (H_2')^\bullet$, $w_2 \in (T_2')^*$ and reduce to previous cases. Consequently, the condition (2) of the definition 7.1 is valid. The proof of (3) is absolutely symmetrical. Proofs of other clauses are similar and omitted here. $\Box$

Using isomorphic and bisimulation relations we can formulate various useful properties for regular macronets. For example it is not hard to show that:

$N_1 \Box N_2 \equiv N_2 \Box N_1$
$(N_1 \Box N_2) \Box N_3 \equiv N_1 \Box (N_2 \Box N_3)$
$N_1 \Box N_1 \approx N_1$

$(N_1 \square N_2);\, N_3 \approx (N_1;\, N_3) \square (N_2;\, N_3)$
$(N_1;\, N_2);\, N_3 \equiv N_1;\, (N_2;\, N_3)\,.$

Other well-known equalities can be rewritten from CCS [22], ACP [7], etc. But we will not do this here. We only formulate some equations without proofs which concern the disruption operation:

$a \bullet b \equiv a \square (a;\, b)$
$a \bullet N \approx (a;\, N) \square N$
$(a;\, N_1) \bullet N_2 \approx (a;\, (N_1 \bullet N_2)) \square N_2$
$(N_1 \square N_2) \bullet N_3 \approx (N_1 \bullet N_3) \square (N_2 \bullet N_3)\,.$

8 RELATION WITH OTHER WORKS

There are a number of approaches [1, 9, 20, 24, 30—33, 35] which enable us to represent complex nets by a composition of simpler ones in order to describe a system hierarchy and to simplify an analysis. We can distinguish all these models by the following features:

- What elements of nets can be macroelements (places or transitions)?
- What nets can be internal for these macroelements?
- How does the execution of macroelements relate with the execution of internal nets?

Macrographs [1, 31] and complex nets [33] use macroplaces while hierarchical nets [9, 20], D-nets [35], constructable nets [24] and recursive nets [30] use macrotransitions. In order to guarantee correctness of the resulting net some subclasses of Petri nets are used as internal nets like WF- and D-blocks [35], regular Petri nets [9], conservative nets [31], A-and P-blocks [33]. In all these models there is a strong relation between the beginning (end) of the execution of the macroelements and the beginning (end) of the execution of the internal net. Therefore each internal net has a some sort of input and output elements. In our approach, however, the removing of a token from a macroplace does not depend on the state of the internal net and can occur at any moment. In other words, the execution of an internal net completely depends on the state of the outside net. Thus, we introduced a new specific kind of net hierarchy, which can be used in combination with the above ones.

Let us compare our algebra with other algebraic approaches (most of them have been considered in [26]). Our algebra can be attributed to denotational approach to a semantic definition of abstract languages such as CCS, TCSP, ACP, etc. We started with one of the earlier works of this stream [20, 21] and enriched it by accounting for practical needs in the area of protocol engineering and fundamental ideas of the above calculus. As a result we have obtained an algebra which has many common features with other ones [10, 13, 15, 25, 34]. One of the main differences is a new treatment of a terminal state of a net defined with the aid of tail structured places. This enables to define an operation of sequential composition (join) in a very natural way. It should be noted that other approaches have either only a prefix form of this operator [15, 25] or express it in terms of parallel composition [23, 18, 34]. In [13] a terminal state of a net defined as absence of tokens in all places and so sequential composition looks like extremely cumbersome.

Another main difference of our algebra consists in using the new disruption operation. Note, however, operation of disruption itself is not new. The similar operation called disabling is used in LOTOS [18]. Also, the operation called mode transfer is defined in ACP [6]. But both these operations are defined in the framework of an interleaving approach and, like a parallel composition, can be expressed in terms of basic operations. We defined disruption in terms of Petri nets that enables us to describe disruption of processes with an explicit concurrency. Moreover, we give the suitable graphical representation of this construction. You can say that it seems to be more reasonable to define a net interpretation of the disruption in the framework of an operational approach to a net-semantic. We believe it is quite right but due to the statement of our problem we are forced to follow and thus to advocate the denotational approach.

9 CONCLUDING REMARKS

The main result presented here is an algebra of regular macronets that serves as a formal basis for the development of special and practical methods of protocol theory: design and specification, validation, hierarchical composition, etc. For example algebra of RMN may be used in composition with one of the extension of Petri nets for complete protocol specification including operations with parameters and variables. RMNs provide graphical structured representation of protocols and allow to design, describe and possibly implement protocols visually. Moreover, the algebraic approach simplifies the reading and understanding of protocols. In fact, this process reproduces protocol design and description

Casting of more realistic equivalence into Kotov's algebra enhances its analytical power. In particular, it enables to solve a protocol verification problem as a demonstration of equivalence between protocols and services [5]. The fact that the equivalence is a congruence relation provides a compositional style in protocol design and specification.

Acknowledgement. It is a pleasure to thank Prof. V. Kotov and L. Cherkasova for their valuable support and permanent interest in this work. The author is also thankful to the anonymous referees for their constructive and useful comments which enabled him to improve this paper substantially.

REFERENCES

[1] ANDRE, C.—BOERI, F.—MARIN, J.: Synthèse et Réalisation de Systèmes Logiques à Évolutions Simultanées. R.A.I.R.O. Automatique, 10, 1976, No. 4, pp. 67—86 (in French).

[2] ANISIMOV, N. A.: An Algebraic Approach to Specification and Verification of Communication Protocols based on the Petri Net Theory. In: Proc. of the Conf. COMPACK '85, Riga 1985, pp. 158—162 (in Russian).

[3] ANISIMOV, N. A.: Petri Nets as Formal Model of Estelle and LOTOS. In: Proc. of the Conf. LOCNET '89, Riga, 1988, Part 1, pp. 76—80 (in Russian).

[4] ANISIMOV, N. A.: A Formal Model for Design and Specification of Communication Protocols Based on the Petri Net Theory. Avtomat. i Vychisl. Tekn., 1988, No. 6, pp. 3—10 (in Russian).

[5] ANISIMOV, N. A.: A Notion of Petri Net Entity for Communication Protocol Design. Institute for Automation and Control Processes, Vladivostok 1989.

[6] BERGSTRA, J. A.: A Mode Transfer Operator in Process Algebra. Report P8808, University of Amsterdam, Programming Research Group, 1988.

[7] BERGSTRA, J. A.—KLOP, J. M.: Algebra of Communication Processes with Abstraction. Theor. Comput. Sci. 37, 1985, No. 1 pp. 77—121.

[8] BRAUER, W.—REISIG, W.—ROZENBERG, G. (Eds): Petri Nets. Part I and II. Proc. of an Advanced Course. Bad Honnef. LNCS 254, 255, 1987.

[9] CHERKASOVA, L. A.—KOTOV, V. E.: Structured Nets. LNCS 118, 1981, pp. 242—251.

[10] DEGANO, P.—DE NICOLA, R.—MONTANARI, U.: A Distribution Semantics for CCS based on Condition/Event Systems. Acta Informatica, 26, 1988, No. 1—2, pp. 59—91.

[11] DIAZ, M.: Petri Net Based Models in the Specification and Verification of Protocols. LNCS 255, 1987, pp. 135—170.

[12] GENRICH, H. G.: Predicate/Transition Nets. LNCS 254, 1987, pp. 207—247.

[13] VAN GLABBEEK, R. J.—VAANGRAGER, F. W.: Petri Net Models for Algebraic Theories of Concurrency (extended abstract). LNCS 259, 1987, pp. 224—242.

[14] GOLTZ, U.: Building Structured Petri Nets. Arbeitspapiere der GMD, 223, Sankt Augustin 1986.

[15] GOLTZ, U.: A Class of CCS Programs Representable by Finite Petri Nets. LNCS 324, 1988, pp. 339—350.

[16] ISO 8072, Information Processing Systems — Open Systems Interconnection — Connection-Oriented Transport Protocol Specification, 1984.

[17] ISO 9074, Information Processing Systems — Open Systems Interconnection — "ESTELLE — A Formal Description Technique Based on an Extended State Transition Model", 1988.

[18] ISO 9074, Information Processing Systems — Open Systems Interconnection — "LOTOS — A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour", 1988.

[19] JENSEN, K.: Coloured Petri Nets. LNCS 254, pp. 248—299.

[20] KOTOV, V. E.: An Algebra for Parallelism Based on Petri Nets. LNCS 64, 1978, pp. 39—59.

[21] LAUER, P. E.—CAMPBEL, R. H.: Formal Semantics of a Class of High-Level Primitives for Coordinating Concurrent Processes. Acta Informatica, 5, 1975, pp. 297—332.

[22] MILNER, R.: A Calculus for Communication Systems. LNCS 92, 1980, 70 pp.

[23] MILNER, R.: Lectures on a Calculus of Cummunication Systems. LNCS 197, 1985, pp. 197—220.

[24] MÜLLER, K.: Constructable Petri Nets. Electron. Int. Verarb. Kybern. EIK, 21, 1985, No. 4/5, pp. 171—199.

[25] OLDEROG, E.-R.: Operational Petri Nets Semantics for CCSP. LNCS 266, 1987, pp. 196—233.

[26] OLDEROG, E.-R.—GOLTZ, U.—VAN GLABBEEK, V. (Eds.): Combining Compositionality and Concurrency — Summary of a GMD Workshop, Konigswinter, March 1988. Arbeitspapiere der GMD, 320, Sankt Augustin 1988.

[27] PARK, D.: Concurrency and Automata on Infinite Sequences. LNCS 104, 1981.

[28] PETERSON, J. L.: Petri Net Theory and the Modelling of Systems. Prentice-Hall, Inc., Englewood Cliffs 1981.

[29] POMELLO, L.: Some Equivalence Notions for Concurrent Systems — An Overwiew. Arbeitspapiere der GMD, Nr. 103, Sankt Augustin 1984.

[30] REISIG, W.: Recursive Nets. Informatik-Fachberichte, 52, 1982, pp. 125—130.

[31] SILVA, M.: Sur le consept de macroplace et son utilisation pour l'analyse des réseaux de Petri. R.A.I.R.O. Automatique, 15, 1981, No. 4, pp. 335—345 (in French).

[32] SUZUKI, I.—MURATA, T.: A Method for Stepwise Refinement and Abstraction of Petri Nets. J. Comput. Syst. Sci., 27, 1983, No 1, pp. 51—76.

[33] TAL, A. A.—YUDITSKY, S. A.: Hierarchy and Parallelism in Petri Nets. Avtomatika i Telemek., 1982, No. 7, pp. 113—122, (in Russian).

[34] TAUBNER, D.: Finite Representation of CCS and TCSP Programs by Automata and Petri Nets. LNCS 369, 1989, 168 pp.

[35] VALETTE, R.: Analysis of Petri Nets by Stepwise Refinements. J. Comput. Syst. Sci., 18, 1979, No. 1, pp. 35—46.

[36] WINSKEL, G.: Petri Nets, Algebras, Morphisms and Compositionality. Inform. and Comput., 72, 1987, No. 3, pp. 197—238.

Received June 26, 1988 — revised version March 15, 1990

Nikolay A. ANISIMOV was born in 1957. He graduated from the Moscow Technical Institute of Physics in 1980 and received the degree of Candidate of Sciences in Engeenering in 1986. Since 1980 he has been working at the Institute of Automation and Control Processes. Far East Division of the USSR Academy of Science. His current research interests include communication protocols, specification and verification techniques. Petri nets, process algebras.