Issues in IFSs and GNs, Vol. 14, 2018/19, 77–91

Generalized Nets with Limited Number of Token Splitting Allowed

Dafina Zoteva[1,2], Anthony Shannon[3] and Maciej Krawczak[4]

[1] Faculty of Mathematics and Informatics
Sofia University "St. Kliment Ohridski"
5 "James Bourchier" Blvd, Sofia 1164, Bulgaria
e-mail: dafy.zoteva@gmail.com

[2] Department of Bioinformatics and Mathematical Modelling
Institute of Biophysics and Biomedical Engineering
Bulgarian Academy of Sciences
105 Acad. Georgi Bonchev Str., Sofia 1113, Bulgaria

[3] Warrane College, the University of New South Wales,
Kensington NSW 2033, Australia

[4] Systems Research Institute, Polish Academy of Sciences
Newelska 6, 01-447 Warsaw, Poland
e-mail: krawczak@ibspan.waw.pl

Abstract: A new extension of the standard generalized nets (GNs), namely Generalized Net with Limited Number of Token Splitting Allowed (GNLNSA), is introduced in this paper. The general algorithm for transition functioning in GNLNSA is presented. It is proved that GNLNSAs are conservative extensions of the class of the standard GNs, i.e., their functioning and the results of their work can be described by standard GNs.

Keywords and phrases: Generalized nets, Generalized nets extensions, Algorithm for transition functioning.
2000 Mathematics Subject Classification: 68Q85.

1 Introduction

The Petri nets, and their extensions and modifications were extended with the concept of Generalized Nets (GNs) in 1982 [7, 9, 10, 12]. The standard GNs are not simply an automatic aggregation of the components of the other types of nets. Some new components have been added to their description so that all other extensions and modifications of the Petri nets can be described by standard GNs.

The first major difference with the other types of Petri nets is the *"history"* that every token accumulates as a result of the characteristics gained during each transfer from an input to an output place of a transition. The *"place-transition"* relation is the second major difference. Transitions, as objects with a complicated nature, contain m input and n output places, $m, n \geq 1$. The transitions' conditions for transfer between these input and output places, as well as the capacities of the transitions' arcs are described by *Index Matrices* (IMs) [8, 11]. This is the third major difference. The fourth major difference is related to the time during which the GN functions. A *global time scale* is associated with the GN. This time scale depends on the particular process described by the GN. All events in the GN evolve over this time scale. In the definition of the standard GN time is discrete and it increases with discrete steps.

Since 1982, the concept of the GNs itself has been extended numerous times. Each extension is defined with the primary purpose providing a means for describing a certain class of processes, or to facilitate an already existing one. The studies on the early defined extensions of the standard GNs, like Intuitionistic Fuzzy GNs of type 1, 2, 3 and 4, Colour GNs, GNs with interval activation time, GNs with complex structure, GNs with global memory, GNs with stop conditions, GNs with tokens' duration of "life", and many others, are summarized in [9, 10, 12]. The more recently defined extensions are GNs with volumetric tokens [14], GNs with characteristics of the places [4, 2], Intuitionistic fuzzy generalized nets with characteristics of the places of types 1 and 3 [1], GNs with time dependent priorities [5], GNs with dynamic priorities [6], GNs with characteristics of the arcs [3], Interval valued intuitionistic fuzzy GNs [13] and GNs with additional intuitionistic fuzzy conditions for tokens transfer [15]. All of these extensions are conservative extensions of the standard GNs.

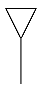
The new extension presented here differs from the other extensions in the splitting of the tokens during their transfer. The number for splitting of a token is referred to here not as the number of identical copies of the token generated during a single transfer, but as the number of transfers during which the token could be split.

In the case of standard GNs and their other extensions, if a token is allowed to split, it splits into as many identical tokens as the number of the corresponding predicates evaluated as *true*. This happens during each transfer of the token. The number of splitting of a token is then in some sense *"unlimited"*. A token is allowed to split an unlimited number of times provided that the GN is still functioning and there are more than one predicates evaluated as *true* during the transfer.

In the case of the new extension presented here, the number for splitting of a certain token is *limited* by an upper bound. The number of times each token is allowed to split is determined by a specific function. This new function is added here to the standard definition of GNs.

The paper is organized as follows. The formal definitions concerning the new extension and the general algorithm for the transition's functioning are presented in Section 2. Section 3 contains the proof that the new extension is a conservative extension of the class of standard GNs. The concluding remarks are in Section 4.

2 Generalized Nets with Limited Number of Token Splitting Allowed

The standard GNs and all their modifications, similarly to the other types of Petri nets, contain transitions, places and tokens. In general, a GN-place is represented as the symbol $\bigcirc$, while the graphical structure of a transition is represented as the symbol , which indicates the transition's conditions.

There is one arc entering and one arc exiting each place. The *GN's input places* are the places without entering arcs. The places with no exiting arcs are the *GN's output places*. *Transition's input places* are those places which are to the left of a transition, while those to the right of a transition are the *transition's output places*.

Each transition has at least one input and one output place. When tokens enter the input places of a transition, it becomes *potentially fired* (activated). The tokens are transferred from input to output places of a transition under certain conditions. A transition is *fired* right at the moment of this transfer.

Each token enters the net with *an initial characteristic*. Consequently, the token receives new characteristics with each transfer. This is an essential difference from the other types of nets, the memory of the GNs.

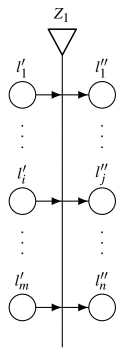


Figure 1. A GNLNSA-transition.

The new extension presented here, namely *Generalized Nets with Limited Number of Token Splitting Allowed* (GNLNSA), is no exception to these general considerations.

Definition 1. Every GNLNSA-transition is given by a seven-tuple (see Figure 1):

$$Z = \langle L', L'', t_1, t_2, r, M, \square \rangle,$$

where

(a) L' and L'' are finite, non-empty set of the transition's input and output places, respectively.

$$L' = \{l'_1, ..., l'_i, ..., l'_m\},$$

$$L'' = \{l''_1, ..., l''_j, ..., l''_n\};$$

(b) t_1 is the current time-moment of the transition's firing;

(c) t_2 is the current value of the duration of its active state;

(d) r is an IM with the transition's conditions for transfer of certain tokens from the transition's input places to corresponding outputs. The IM r has the form:

$$r = \begin{array}{c|c} & l''_1 \ \dots \ l''_j \ \dots \ l''_n \\ \hline l'_1 & \\ \vdots & r_{i,j} \\ l'_i & (r_{i,j} \ - \ \text{predicates}) \\ \vdots & (1 \le i \le m, 1 \le j \le n) \\ l'_m & \end{array}$$

where the element $r_{i,j}$ is a predicate which evaluation determines the possibility of transferring a token from the i-th input to the j-th output place. This transfer is only possible when the $r_{i,j}$ truth value is $true$;

(e) M is an IM of the capacities of the transition's arcs:

$$M = \begin{array}{c|c} & l''_1 \ \ldots \ l''_j \ \ldots \ l''_n \\ \hline l'_1 & \\ \vdots & m_{i,j} \\ l'_i & (m_{i,j} \in \mathscr{N}, \\ \vdots & 1 \le i \le m, 1 \le j \le n) \\ l'_m & \end{array},$$

where $\mathscr{N} = \{0, 1, 2, \ldots\} \cup \{\infty\}$;

(f) $\Box$ is the transition's type. It has the form of a Boolean expression with the identifiers of the transition's input places as variables, and the Boolean operations $\wedge$ and $\vee$. This formula has the following semantics:

$\wedge(l_{i_1}, l_{i_2}, \ldots, l_{i_u})$ — there has to be at least one token in each of the places $l_{i_1}, l_{i_2}, \ldots, l_{i_u}$,
$\vee(l_{i_1}, l_{i_2}, \ldots, l_{i_u})$ — there has to be at least one token in one of all places $l_{i_1}, l_{i_2}, \ldots, l_{i_u}$,
where $\{l_{i_1}, l_{i_2}, \ldots, l_{i_u}\} \subset L'$;

The transition can become active only when the value of the transition's type, evaluated as a Boolean expression, is $true$.

Definition 2. The ordered four-tuple

$$E = \langle \langle A, \pi_A, \pi_L, c, f, \theta_1, \theta_2 \rangle, \langle K, \pi_K, \theta_K, \sigma_{K,T} \rangle, \langle T, t^o, t^* \rangle, \langle X, \Phi, b \rangle \rangle$$

is called a *Generalized Net with Limited Number of Token Splitting Allowed* (GNLNSA), if:

(a) A is a set of transitions, defined according to Definition 1;

(b) π_A is a function which gives the priorities of the transitions, i. e., $\pi_A : A \to \mathscr{N}$;

(c) π_L is a function which gives the priorities of the places, i.e., $\pi_L : L \to \mathscr{N}$, where

$$L = pr_1 A \cup pr_2 A\,,$$

i.e., L is the set of all GN-places, and $pr_i\{x_1, x_2, ..., x_n\} = x_i$, where $n \in \mathscr{N}$, $n \geq 1$ and $1 \leq i \leq n$;

(d) c is a function which gives the capacities of the places, i.e., $c : L \to \mathscr{N}$;

(e) f is a function which evaluates the truth values of the predicates of the transition's conditions;

(f) θ_1 is a function which gives the next moment of time when a transition can be activated, i.e., $\theta_1(t) = t'$, where $t = pr_3 Z$, and $t, t' \in [T, T + t^*]$ and $t \leq t'$. The value of this function is calculated when the transition terminates its active state;

(g) θ_2 is a function which gives the duration of the active state of a transition, i.e., $\theta_2(t) = t'$, where $t = pr_4 Z$, $t \in [T, T + t^*]$ and $t' \geq 0$. The value of this function is calculated at the moment of the transition's activation;

(h) K is the set of the GN's tokens;

(i) π_K is a function which gives the priorities of the tokens, i.e., $\pi_K : K \to \mathscr{N}$;

(j) θ_K is a function which gives the moment of time when a certain token can enter the net, i.e., $\theta_K(\alpha) = t$, where $\alpha \in K$ and $t \in [T, T + t^*]$;

(k) $\sigma_{K,T}$ is the modification which marks the difference between the standard GNs, the rest of their extensions, and the one proposed here. $\sigma_{K,T}$ is a function which gives the number of times a certain token is allowed to split in a particular moment of time during the functioning of the GN, $\sigma_{K,T} : K \times [T, T + t^*] \to \mathscr{N}$.

(l) T is the moment of time when the GN starts functioning. This moment is determined with respect to a fixed (global) time-scale;

(m) t^o is an elementary time-step related to the fixed (global) time-scale;

(n) t^* is the duration of the GN functioning;

(o) X is the function which assigns initial characteristics to each of the tokens when they enter the GN;

(p) Φ is a characteristic function which gives a new characteristic to each token upon its transfer from an input to an output place of a certain transition;

(q) b is a function which gives the maximum number of characteristics a certain token can receive, i.e., $b : K \to \mathcal{N}$.

The general algorithm for transition's functioning, described in [9, 10, 12], is modified here to take into consideration the number of times a given token and its derivatives are allowed to split during the GN functioning. Initially this value is assigned by the function $\sigma_{K,T}$ to each token upon entering the GN.

Apart from the splitting of tokens, the algorithm for tokens transfer after the moment of time $t_1 = TIME$ (the current GN moment of time), denoted by *algorithm A*, takes into consideration also the possibility of merging tokens.

The list of tokens which can be merged with a given token α have to be specified in the initial characteristics of this α token. For example, the expression

$$x_0^{\alpha} = \text{“}\langle\{\beta_1, ..., \beta_k\}, x_0^{\alpha,*}\rangle\text{”},$$

denotes that the α token can be merged with the tokens in the set $\{\beta_1, ..., \beta_k\}$. The rest of the information of the token's initial characteristics is stored in $x_0^{\alpha,*}$.

The *algorithm A* can be described in 12 steps, as follows:

A01 Sort the input and output places of the transitions by their priorities.

A02 Form two lists of tokens in each input place l. The first list contains those of the tokens that might be transferred to a certain output place during the current time moment. Sort these tokens by their priorities. The second list is empty at first. The two lists shall be denoted with P_1 and P_2, respectively.

A03 Generate an empty index matrix R that corresponds to the index matrix of the predicates r. The values in the matrix R can only be 0 or 1, which correspond to predicate evaluations $false$ and $true$. Assign 0 to all elements $R_{i,j}$ of R which:

- are in a row that corresponds to an empty input place, i.e., there are no tokens in the input place that can be transferred to an output place of the current transition;
- are placed in the position (i, j) for which the predicate $r_{i,j}$ is set as *false* or $m_{i,j} = 0$, i.e. the current capacity of the arc between the i^{th} input place and the j^{th} output place is 0.

Proceed with step **A04**.

A04 Iterate through the input places in the order set by their priorities, starting with the place with highest priority for which no token has been transferred during the current time-step and which has at least one token in it. The token α which might be transferred on the current time-step is the one with the highest priority in the P_1 list of the current input place. Perform the following steps in order to determine if and where to transfer the current token.

A04a Find the next $R_{i,j}$ value on the relevant row of the IM R which has not been checked on the current time-step. If such a value exists, go to step **A04b**. If all the values on the relevant row of the IM R have been checked, go to step **A04g**.

A04b Check **if** the relevant **output place**, which the token α might be transferred to, **is full**. If so, go to step **A04c**, otherwise go to step **A04d**.

A04c Check if the relevant output place **has a token (or tokens) that can be merged** with the one being transferred. If so, go to step **A04d**. Otherwise, set $R_{i,j}$ value to 0 and go to step **A04a** to check the next element of R on the relevant row.

A04d Evaluate the truth value of the corresponding predicate $r_{i,j}$ of the index matrix r. If $r_{i,j}$ is $true$, set the $R_{i,j}$ value of R to 1 and go to step **A04e**. Otherwise set the $R_{i,j}$ value to 0 and go to step **A04a**.

A04e Check **if the current token can still split**. This is possible, if the value of the function $\sigma_{K,T}$ for the token α in the moment of the transition activation $TIME$ is not 0.

If $\sigma_{K,T}(\alpha, TIME) \neq 0$, go to step **A04a** to check the rest of the elements of R on the relevant row. If the token cannot split, go to **A04f**.

A04f The token α is transferred to the corresponding output place and is merged with specified tokens in the new host, if there are such. The number of times the tokens, merged with the token α, are allowed to split does not change, since $\sigma_{K,T}(\alpha, TIME) = 0$.

Evaluate the characteristic function of this output place and assign this value as a new characteristic of the transferred token upon entering the output places.

Proceed with step **A05**.

The evaluation of the predicates stops with the first one evaluated as *true*. The token is moved then to the highest priority output place amongst those, the token can be transferred to.

A04g If all the values on the relevant row are 0, go to step **A05**.

If there is **more than one** $R_{i,j}$ value set to 1, the token α splits in as many new tokens as the number of the $R_{i,j}$ values set to 1. These newly generated tokens are identical to the original token α in any other way except for one. The number of times each of the newly generated tokens is allowed to split is set to $\sigma_{K,T}(\alpha, TIME - t^o) - 1$. These tokens are transferred to the corresponding output places and are merged with specified tokens in the new hosts, if there are such.

In case the token α_i, generated by the splitting of the original token α, is merged with the tokens $\{\beta_1, ..., \beta_k\}$ in the new host, the number of times each of the resulting tokens is allowed to split is set to $\max(\sigma_{K,T}(\alpha_i, TIME), \sigma_{K,T}(\beta_j, TIME))$, where $\beta_j \in \{\beta_1, ..., \beta_k\}$.

If there is **exactly one** $R_{i,j}$ value set to 1, the token that is transferred to the corresponding output place is the original token α. No splitting is performed in this case. The number of times the token α is allowed to split does not change.

In case the token α is merged with the tokens $\{\beta_1, ..., \beta_k\}$ in the new host, the number of times each of the resulting tokens is allowed to split is set to $\max(\sigma_{K,T}(\alpha, TIME), \sigma_{K,T}(\beta_j, TIME))$, where $\beta_j \in \{\beta_1, ..., \beta_k\}$.

Evaluate the characteristic function of these output places. Assign these values as new characteristics to the corresponding transferred tokens upon entering the output places.

Proceed with step **A05**.

A05 If the highest priority token cannot be transferred during the current time step, move the token to the P_2 list of the input place.

A06 Increase by 1 the current number of tokens in each output place to which a token has been transferred if the token has not been merged with any of the other tokens in the host. Do not change the current number of tokens in the output place otherwise.

A07 Decrease by 1 the current number of tokens in each input place from which a token has been transferred. If the current number of tokens in such an input place becomes 0, set to 0 all the elements in the corresponding row of the index matrix R.

A08 Decrease by 1 the capacities of all the arc through which a token has been transferred. If the current capacity of an arc becomes 0, assign 0 to this element of the index matrix R that corresponds to the arc.

A09 If there are more input places with lower priority from which no token has been transferred to an output place, go to step **A04**. Otherwise, go to step **A10**.

A10 Add t^0 to the current model time.

A11 If the value of the current time is less than or equals $t_1 + t_2$ (the time components of the considered transition), go to **A04**. Otherwise, go to step **A12**.

A12 End of the transition's functioning.

The general algorithm for the GN's functioning is the same as the one of a standard GN.

3 GNLNSAs are conservative extension of the standard GNs

Let Σ be the class of the standard GNs and Σ_{LNSA} be the class of the GNLNSAs. Every standard GN is a GNLNSA, i.e. the relation

$$\Sigma \vdash \Sigma_{LNSA}$$

holds.

Every standard GN can be seen as a GNLNSA in which

$$\sigma_{K,T}(\alpha,t) = \infty, \forall\alpha \in K, \forall t \in [T, T+t^*],$$

if the tokens are allowed to split. If the tokens are not allowed to split,

$$\sigma_{K,T}(\alpha,t) = 0, \forall\alpha \in K, \forall t \in [T, T+t^*].$$

Therefore, GNLNSAs are extensions of the standard GNs.

In addition, every GNLNSA can be represented as a standard GN.

Theorem 1: The class Σ_{LNSA} is a conservative extension of the class Σ, i.e.

$$\Sigma_{LNSA} \equiv \Sigma.$$

Proof.

It is sufficient to show that the functioning and the results of the work of every GNLNSA can be described by some ordinary GN.

Let the GNLNSA

$$E = \langle\langle A, \pi_A, \pi_L, c, f, \theta_1, \theta_2\rangle, \langle K, \pi_K, \theta_K, \sigma_{K,T}\rangle, \langle T, t^o, t^*\rangle, \langle X, \Phi, b\rangle\rangle$$

be given.

Let G be a standard GN with the following definition:

$$G = \langle\langle A, \pi_A, \pi_L, c, f, \theta_1, \theta_2\rangle, \langle K, \pi_K, \theta_K\rangle, \langle T, t^o, t^*\rangle, \langle X^G, \Phi^G, b\rangle\rangle,$$

where

$$X^G = X \cup \{x_0^\alpha \mid \alpha \in K, x_0^\alpha = \sigma_{K,T}(\alpha, \theta_K(\alpha)), x_0^\alpha \in \mathcal{N}\},$$

$$\Phi^G = \Phi \cup \{n \mid n = \sigma_{K,T}(\alpha,t), \alpha \in K, t \in [T, T+t^*], n \in \mathcal{N}\}.$$

The standard GN G has the same static components and graphical structure as the GNLNSA E. The time components, the capacities of the arcs, places and transitions, the priorities, even the tokens are identical in both of the nets. The only two components that differ them are the function X, which gives the initial characteristics of the tokens, and the function Φ, which assigns new characteristics to each token upon its transfer.

In the case of the standard GN G, the function X^G assigns to each token the same initial characteristics as the function X in the net E, but in addition the functions X^G sets the initial number of times each of the tokens can split. These values are calculated at the moment the tokens enter the GN G. Apart from the new characteristics assigned to each token by the function Φ in E, the characteristic function Φ^G adds as a new characteristic the number of times each token is allowed to split at the current moment of time $t, t \in [T, T+t^*]$.

Now, do the GNs G and E function in the same way? Are the results of their work equal?

The functioning and the results of the work of a transition Z^G in G will be compared to the functioning and respectively the results of the work of its corresponding transition Z^E in E.

Let $Z^E = \langle L', L'', t_1, t_2, r, M, \square \rangle$ is a transition in the GNLNSA E.

The corresponding transition in the standard GN G is defined in the following way:

$$Z^G = \langle L', L'', t_1, t_2, r^G, M, \square \rangle.$$

These two transitions become active at the same moment of time and have equal duration of the functioning. The sets of input and output places are also equal. The priorities and the capacities of the corresponding places are equal, as well as the capacities of the arcs.

The transition's conditions in r^G, however, have to be constructed in a way which guarantees that a token α^G will have an identical behaviour with the corresponding token α^E in Z^E. For this purpose, the predicate $r^G_{i,j}$ has the following form:

$$r^G_{i,j} = r_{i,j} \wedge (\sigma_{K,T}(\alpha^G, TIME) \neq 0 \vee \neg r^G_{i,k}, \forall k < j).$$

The initial number of times each token in G is allowed to split is set as an initial characteristic by the function X^G.

If $r_{i,j} = false$, then $r^G_{i,j}$ will be evaluated as $false$. If $r_{i,j} = true$, then $r^G_{i,j}$ will have the same truth value as the second argument of the conjunction.

If the number of times the token α^G is allowed to split at the moment of the transition activation is not 0, i.e., $\sigma_{K,T}(\alpha^G, TIME) \neq 0$, the $r^G_{i,j}$ is evaluated as $true$. The token α^G will then split in as many identical tokens as the number of the predicates $r^G_{i,j}$ evaluated as $true$, which is exactly the number of the predicates $r_{i,j}$ evaluated as $true$.

In the case of $\sigma_{K,T}(\alpha^G, TIME) = 0$, the token α^G is not allowed to split any more. The token then has to be transferred to the first output place l_j for which the corresponding predicate $r^G_{i,j}$ is evaluated as $true$. In the standard GNs, the tokens are split each time there is more than one predicate evaluated as $true$ in the transition's conditions. Therefore, there should be no more than one predicate, evaluated as $true$, in the corresponding row of the IM r^G. Since the output places are processed sequentially, this means that there should be no predicates evaluated as $true$ before the currently evaluated one, or $r^G_{i,k}$ should be $false, \forall k < j$.

The so defined predicates in G guarantee that the tokens will be transferred to the output places which correspond to the output places determined by the predicates in E. When a token α^G is transferred to an output place, it receives a new characteristic, assigned by the function Φ^G. Apart from the characteristics which the corresponding token in E receives when it is transferred, the token α^G in G also receives the number of times it is allowed to split. The maximum number of characteristics each of the tokens can have is equal in both of the nets.

Therefore, the corresponding tokens behave in the same way. Since they are chosen randomly, it can be concluded that the same assertion can be made for any other pair of corresponding tokens of the nets. This leads to the point that the two corresponding transitions Z^G and Z have identical behaviour. Hence, the GN G can describe the functioning and the work of the GNLNSA E.

4 Conclusion

A new conservative extension of the concept of standard GNs, namely Generalized Net with Limited Number of Token Splitting Allowed, has been presented here together with its specific algorithm for the functioning of the transitions.

The possibilities for the application of GNLNSAs are topics for further research. An operator to be applied to a token during the functioning of the net, in order to change the number of times this token is allowed to split, will be defined as the next step in the development of this new class of GNs.

It will be interesting to study combinations of the new GN-extension with other GN-extensions, e.g., Generalized Nets with Volumetric Tokens (GNVT) [14], Generalized Nets with Characteristics of the Places (GNCP) [4], Generalized Nets with Characteristics of the Arcs (GNCA) [3], Intuitionistic Fuzzy Generalized Nets with Characteristics of the Places of Types 1 (IFGNCP1) and

3 (IFGNCP3) [1], etc.

Acknowledgements

This work is partially supported by the National Science Fund of Bulgaria under Grant DN02/10 "New Instruments for Knowledge Discovery from Data, and their Modelling".

References

[1] Andonov, V., Intuitionistic fuzzy generalized nets with characteristics of the places of Types 1 and 3. *Notes on Intuitionistic Fuzzy Sets*, Vol. 19, 2013, No 3, 99–110.

[2] Andonov, V., Reduced generalized nets with characteristics of the places. *International Journal "Information Models and Analyses"*, Vol. 3, No 2, 2014, 113–125.

[3] Andonov V., Generalized nets with characteristics of the arcs. *Compt. rend. Acad. bulg. Sci.*, Vol. 70, 2017, No 10, 1341–1346.

[4] Andonov, V., K. Atanassov, Generalized nets with characteristics of the places. *Compt. rend. Acad. bulg. Sci.*, Vol. 66, 2013, No 12, 1673–1680.

[5] Angelova, N., Generalized nets with time dependent priorities. *Proc. of 15th Int. Workshop on Generalized Nets*, Burgas, 16 October 2014, 17–22.

[6] Angelova, N., Generalized Nets with Dynamic Priorities. *Issues in Intuitionistic Fuzzy Sets and Generalized Nets*, Vol. 11, 2014, 15–22.

[7] Atanassov, K., The Generalized nets and the other graphical means for modelling. *AMSE Review*, Vol. 2, 1985, No. 1, 59–64.

[8] Atanassov, K., Generalized index matrices. *Compt. rend. Acad. bulg. Sci.*, Vol. 40, 1987, No. 11, 15–18.

[9] Atanassov, K., *Generalized Nets*. Singapore, New Jersey, London, World Scientific, 1991.

[10] Atanassov, K., *On Generalized Nets Theory*. "Prof. M. Drinov" Academic Publishing House, Sofia, 2007.

[11] Atanassov, K., *Index Matrices: Towards an Augmented Matrix Calculus*. Springer, Cham, 2014.

[12] Atanassov, K., E. Sotirova, *Generalized Nets Theory*. "Prof. M. Drinov" Academic Publishing House, Sofia, 2017 (in Bulgarian).

[13] Atanassov, K., D. Zoteva, N. Angelova, Interval valued intuitionistic fuzzy generalized nets. *Notes on Intuitionistic Fuzzy Sets*, Vol. 24, 2018, No. 3, 111–123.

[14] Atanassova, V., Generalized Nets with Volumetric Tokens. *Compt. rend. Acad. bulg. Sci.*, Vol. 65, No 11, 1489–1498.

[15] Zoteva, D., E. Szmidt, J. Kacprzyk, Generalized nets with additional intuitionistic fuzzy conditions for tokens transfer. *Notes on Intuitionistic Fuzzy Sets*, Vol. 25, 2019, No. 2, 104–114.