Notes on Intuitionistic Fuzzy Sets
Print ISSN 1310–4926, Online ISSN 2367–8283
2026, Volume 32, Number 3, 325-332
DOI: 10.7546/nifs.2026.32.3.325-332

Extending OnlineGN with evaluation of intuitionistic fuzzy predicates

Angel Ivanov Dimitriev [1,2]

[1] Department of Bioinformatics and Mathematical Modelling,
Institute of Biophysics and Biomedical Engineering,
Bulgarian Academy of Sciences, Sofia, Bulgaria

[2] Faculty of Mathematics and Informatics,
Sofia University "St. Kliment Ohridski", Sofia, Bulgaria
e-mail: adimitriev@fmi.uni-sofia.bg

Received: 28 August 2026 **Revised:** 17 October 2026
Accepted: 22 October 2026 **Online First:** 6 November 2026

Abstract: OnlineGN is a web-based environment for constructing and simulating Generalized Net (GN) models. This paper presents its support for intuitionistic fuzzy transition predicates. Rather than returning only a Boolean value, a predicate can be evaluated by separate degrees of validity and non-validity. OnlineGN interprets this pair using one of seven token-transfer conditions from the theory of intuitionistic fuzzy GNs. The evaluation condition is selected in the interface, so the same model can be simulated under different transfer rules without rewriting its predicates. The paper explains how intuitionistic fuzzy predicate evaluation is incorporated into the existing OnlineGN simulation process. A simple example with the evaluation $\langle 0.4, 0.4 \rangle$ shows how strict and non-strict comparison conditions can lead to different token-transfer decisions. The accompanying figures illustrate the predicate configuration and simulation interface.

Keywords: Generalized nets, OnlineGN, Intuitionistic fuzzy predicates, Token transfer, Simulation.
2020 Mathematics Subject Classification: 03E72, 68Q85.

Copyright © 2026 by the Author. This is an Open Access paper distributed under the terms and conditions of the Creative Commons Attribution 4.0 International License (CC BY 4.0). https://creativecommons.org/licenses/by/4.0/

1 Introduction

Atanassov's Generalized Nets (GNs) extend Petri nets by introducing predicates that govern token transfers and characteristics that can change as tokens move through the net [4]. A transition therefore depends not only on the positions of its tokens but also on the conditions associated with possible movements. This makes GNs suitable for processes that combine parallel activity with conditional decisions.

The connection between Generalized Nets and fuzzy set theory was examined by Atanassov in his 1985 paper on generalized nets and their fuzzings [1]. This earlier work provides historical context for subsequent developments in intuitionistic fuzzy Generalized Nets.

As a model grows, keeping track of its predicates, token characteristics, and possible transfers becomes increasingly cumbersome. OnlineGN provides a browser-based editor and simulator for this purpose [2]. Users can construct a net, attach functions to its elements, and inspect token movement step by step. An earlier application used OnlineGN to simulate a parallel formulation of the Towers of Hanoi problem [8].

The conventional predicate evaluation in a GN is Boolean. That is enough for an unambiguous condition, but it discards the distinction between support for a statement, support against it, and the part that remains undetermined. Intuitionistic fuzzy sets retain this information through separate membership and non-membership degrees [3, 5]. Applying the same representation to a transition predicate gives the simulator more information than a single true/false result.

The theory of intuitionistic fuzzy Generalized Nets (IFGNs) includes several ways to interpret such evaluations. Atanassov, Dimitrov, and Atanassova specify seven conditions for deciding whether a token may pass a predicate [6]. These conditions need not agree: an evaluation accepted by one condition may be rejected by another.

This work concerns the predicate decision in OnlineGN, rather than the definition of a new IFGN class. The implementation retains OnlineGN's existing handling of capacities, priorities, and token movement, while adding a selectable interpretation of intuitionistic fuzzy predicate evaluations. We first recall the seven transfer conditions, then explain their use in the simulator. The final example compares conditions C_4 and C_5 using the OnlineGN interface. Figures 1, 2, and 4 are screenshots of the application; the token-position diagram in Figure 3 is a schematic representation.

2 Theoretical background

2.1 Intuitionistic fuzzy predicate evaluations

A classical predicate takes a Boolean value. For the intuitionistic fuzzy evaluation considered here, a predicate p is assigned a pair

$$V(p) = \langle \mu(p), \nu(p) \rangle,$$

where $\mu(p)$ is its degree of validity and $\nu(p)$ its degree of non-validity [3, 5]. The values satisfy the conditions:

$$0 \leq \mu(p), \nu(p) \leq 1, \qquad \mu(p) + \nu(p) \leq 1.$$

The remaining part,

$$\pi(p) = 1 - \mu(p) - \nu(p),$$

is the degree of indeterminacy. For example, $\langle 0.6, 0.2\rangle$ leaves 0.2 undetermined. The pair describes the evaluation of the predicate, but it does not by itself decide whether the corresponding token transfer is allowed.

2.2 Predicates in Generalized Nets

A GN transition is described by the seven-tuple [4]

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

Here L' and L'' denote the input and output places, t_1 and t_2 specify timing, r is the predicate matrix, M gives the arc capacities, and $\Box$ specifies the activation type. Each matrix entry $r_{i,j}$ governs a potential transfer from input place l'_i to output place l''_j.

In the classical setting, $f(r_{i,j})$ returns a Boolean result. With an intuitionistic fuzzy predicate, the evaluation has the form

$$f(r_{i,j}) = \langle \mu_{i,j}, \nu_{i,j}\rangle, \qquad 0 \le \mu_{i,j}, \nu_{i,j} \le 1, \quad \mu_{i,j} + \nu_{i,j} \le 1.$$

The simulator must reduce this pair to a transfer decision before applying the remaining movement rules. IFGN theory also considers other forms of intuitionistic fuzziness, including evaluations of token quantities and characteristics [6, 7]. Those variants are outside the scope of this work: only the transition predicates are evaluated as intuitionistic fuzzy pairs.

2.3 Seven conditions for token transfer

For a predicate evaluation $\langle \mu_{i,j}, \nu_{i,j}\rangle$, the seven transfer conditions are given by Atanassov, Dimitrov, and Atanassova [6]:

$$\begin{aligned}
C_1 &: \quad \mu_{i,j} = 1, \quad \nu_{i,j} = 0,\\
C_2 &: \quad \mu_{i,j} > \tfrac{1}{2},\\
C_3 &: \quad \mu_{i,j} \ge \tfrac{1}{2},\\
C_4 &: \quad \mu_{i,j} > \nu_{i,j},\\
C_5 &: \quad \mu_{i,j} \ge \nu_{i,j},\\
C_6 &: \quad \mu_{i,j} > 0,\\
C_7 &: \quad \nu_{i,j} < 1.
\end{aligned}$$

The seven conditions differ mainly in how they handle particular values. Condition C_1 requires complete validity, while C_2 and C_3 differ in whether a validity degree of exactly 0.5 is accepted. Similarly, C_4 and C_5 compare the degrees of validity and non-validity, with different results when the two are equal. Conditions C_6 and C_7 are less restrictive, requiring only a positive degree of validity or a non-validity degree below one, respectively.

For example, $\langle 0.4, 0.35\rangle$ passes conditions C_4 and C_5, but fails conditions C_2 and C_3. Let D_k denote the Boolean decision obtained by applying condition C_k to a predicate:

$$D_k(r_{i,j}) = \begin{cases} 1, & \text{if } \langle \mu_{i,j}, \nu_{i,j}\rangle \text{ satisfies } C_k, \\ 0, & \text{otherwise,} \end{cases} \qquad k \in \{1, \ldots, 7\}.$$

A value of $D_k = 1$ indicates that the predicate is satisfied; it does not guarantee a movement. Activation, capacities, token priorities, and splitting rules remain in force.

3 Implementation of intuitionistic fuzzy predicates in OnlineGN

3.1 OnlineGN architecture and modeling environment

OnlineGN uses Angular and TypeScript on the frontend, Node.js and Express on the backend, and MongoDB for model storage. D3.js and Graphviz support the graphical presentation of nets. The source code is available in the OnlineGN repository.

The model contains places, transitions, and tokens together with their properties. A place has a capacity, priority, and characteristic functions; a connection in a transition has a predicate and capacity; and a token carries characteristics that may change after movement. The editor exposes these settings, while the simulator lets the user follow the execution step by step.

OnlineGN uses different components to manage the simulation. The page component controls each simulation step, while the compiler service runs the functions defined by the user. The movement service handles token transfers, and the graph service updates their positions in the GN. The intuitionistic fuzzy predicate evaluation is added to the existing simulation process, so no changes to the graphical representation are needed.

Figure 1 is a screenshot of the OnlineGN editor and simulation environment. It shows the rendered GN alongside the global-function editor and simulation controls.

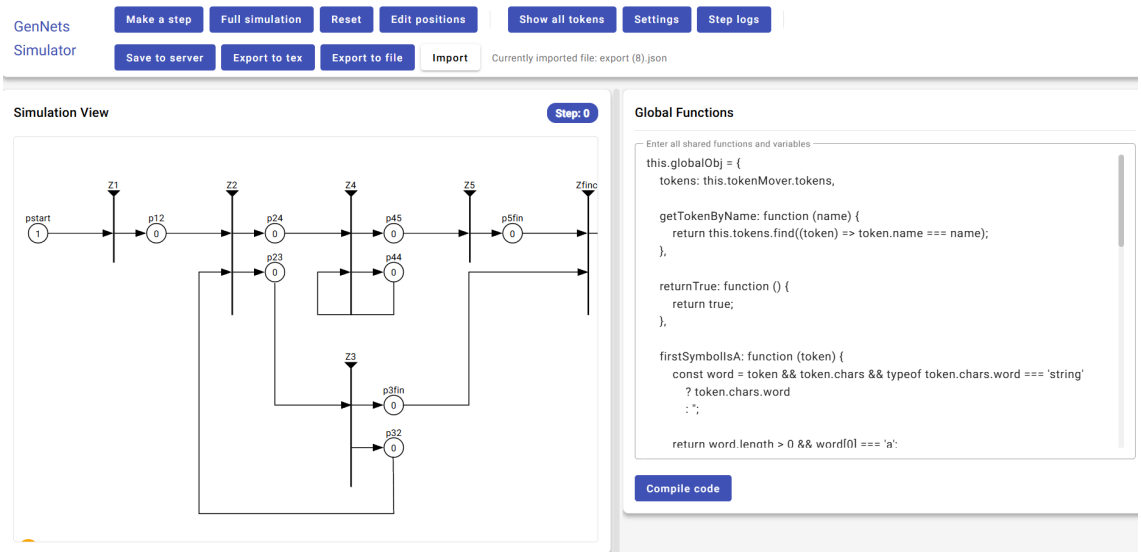


Figure 1. Screenshot of the OnlineGN editor and simulation interface, showing a GN model and its global functions.

3.2 Configuring intuitionistic fuzzy predicates

Each entry in a transition's predicate matrix is associated with a user-defined function, which may depend on the characteristics of the token being evaluated. In the original workflow, that function produces a Boolean result used to permit or reject the corresponding transfer.

With the implemented extension, a predicate may return $\langle \mu, \nu \rangle$ instead of a Boolean value. OnlineGN evaluates this pair using the condition C_k selected by the user. The resulting Boolean decision is passed to the normal movement procedure. The evaluation condition is configured through the interface, separately from the predicate function itself.

The interface offers both Boolean and intuitionistic fuzzy evaluation modes. In the latter, the predicate configuration includes fields for the degrees of validity and non-validity and a selector for the seven transfer conditions. Figure 2 is a screenshot of these controls in OnlineGN. In the configuration shown, $\mu = 0.4$ and $\nu = 0.4$, giving an indeterminacy degree of 0.2, and condition C_5 is selected. Keeping the condition separate from the predicate also allows the same net to be compared under different rules.

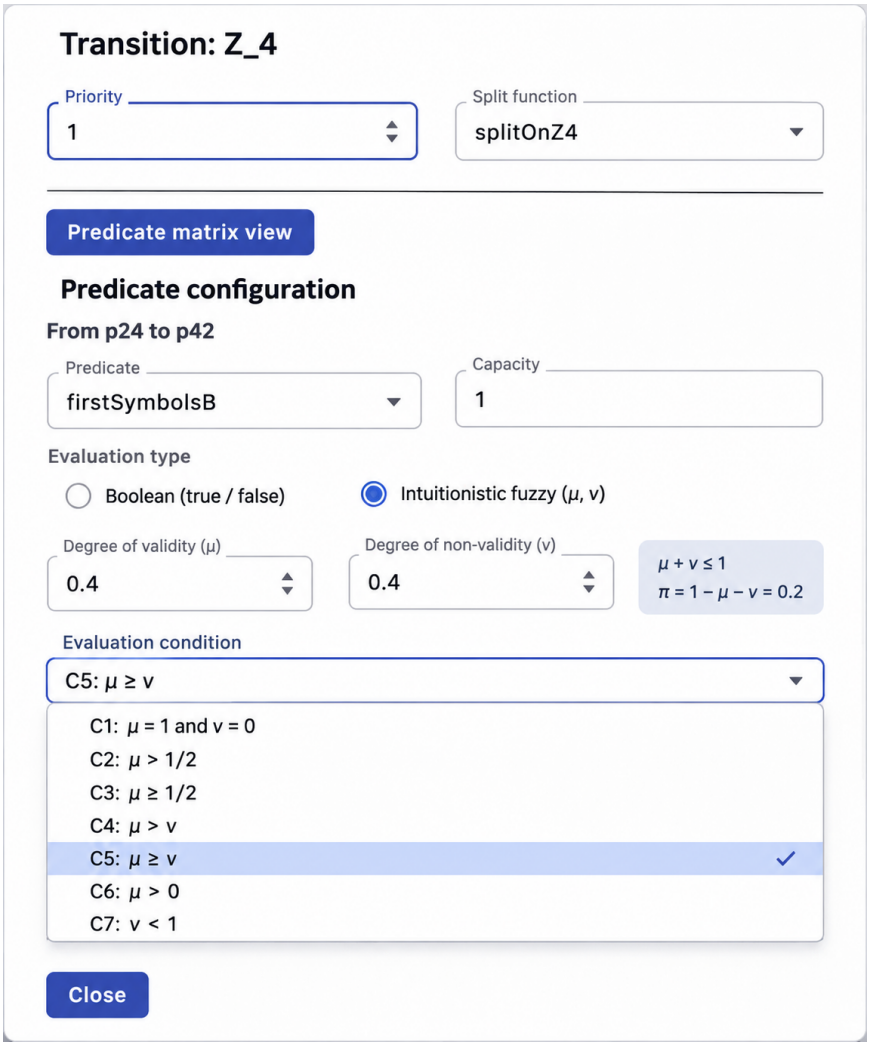


Figure 2. Screenshot from OnlineGN showing the transition predicate configuration, the intuitionistic fuzzy evaluation fields, and the selection of condition C_5.

3.3 Predicate evaluation during simulation

At each simulation step, OnlineGN processes transitions and tokens according to their priorities. The method `registerAllMoves()` gathers candidate transfers, while`canTokenContinue()` checks the connection capacity, destination capacity, and predicate result. To obtain that result, `shouldTokenContinue()` calls the user-defined predicate through `CodeCompilerService`.

For intuitionistic fuzzy predicates, the pair $\langle \mu, \nu \rangle$ is evaluated according to the selected condition C_k. The result is a Boolean value D_k, which determines whether the predicate permits token movement. The remaining rules of the simulation still apply. For example, a token cannot move if the corresponding arc has reached its capacity or the destination place is full.

Once the eligible transfers have been identified, OnlineGN follows its existing procedure for choosing and registering movements, including the rules for priorities and token splitting. The movement service applies the selected transfers, the graph service displays the resulting positions, and characteristic functions update tokens where appropriate. The added operation is the interpretation of a predicate result, not a replacement for the GN execution procedure.

To show the effect of the condition selector, we consider a single transition in which the predicate evaluation stays fixed.

4 Example and simulation views

Consider a transition Z with input place l_1, output place l_2, and a token α initially in l_1. The connection to l_2 is controlled by predicate r. We use the intuitionistic fuzzy evaluation

$$f(r) = \langle 0.4, 0.4 \rangle.$$

The associated indeterminacy is 0.2. In this comparison the transition is active, and the connection and destination place have sufficient capacity. Thus, the two conditions are evaluated against the same predicate pair; no other movement constraint is intended to distinguish the cases.

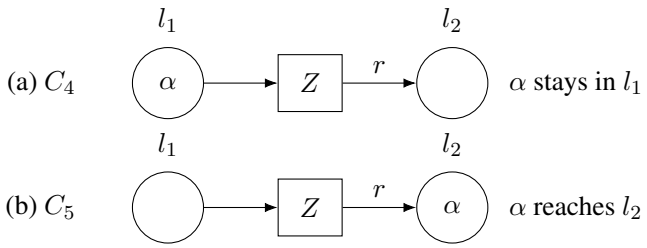


Figure 3. Scheme of the token positions corresponding to the C_4 and C_5 predicate decisions. The diagram is drawn from the GN model; the interface screenshots are shown separately.

Under condition C_4, the test is $\mu > \nu$. It fails because $0.4 > 0.4$ is false, so the predicate blocks the transfer. Under condition C_5, which permits equality, $0.4 \geq 0.4$ is true and the predicate permits the token to move to l_2. Table 1 compares the decisions, and Fig. 3 represents the resulting token positions.

Table 1. Predicate decisions for $f(r) = \langle 0.4, 0.4 \rangle$ under conditions C_4 and C_5.

Condition	Predicate test	Token movement
C_4	$0.4 > 0.4$ (false)	α remains in l_1
C_5	$0.4 \geq 0.4$ (true)	α may move to l_2

Figure 4 presents screenshots of the OnlineGN interface for these two configurations. Both panels show the same predicate function, returning the degrees $\mu = 0.4$ and $\nu = 0.4$. In the C_4 view, the step-log display reports that the predicate test is false and no token movement is registered. In the C_5 view, the test is true and the displayed transfer is $\alpha : l_1 \to l_2$. The two views illustrate how selecting a different evaluation condition changes the predicate decision while preserving the model and its input evaluation.

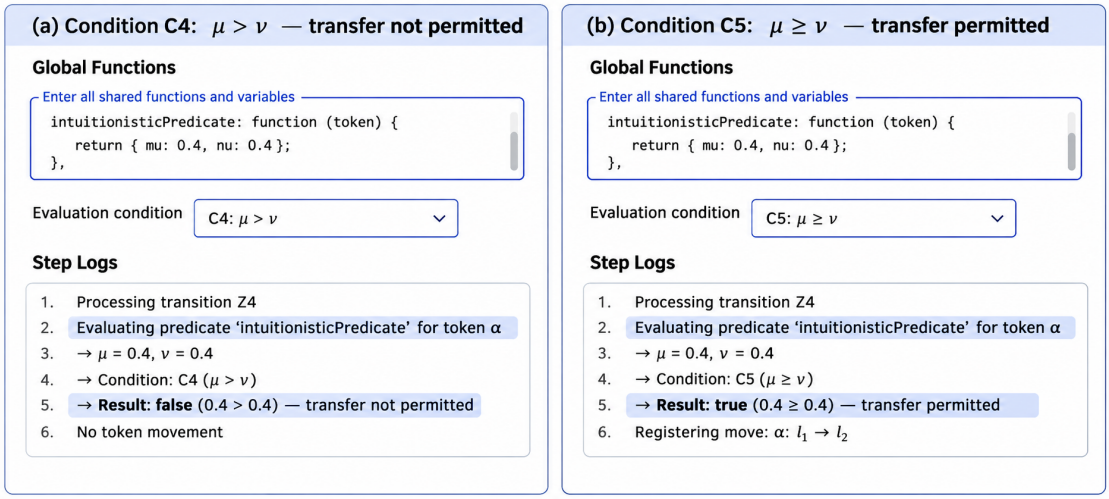


Figure 4. Screenshots of the OnlineGN interface for conditions C_4 and C_5, showing the predicate function, selected evaluation condition, and step-log displays of the corresponding transfer decisions.

5 Conclusion

OnlineGN supports intuitionistic fuzzy predicate evaluation through seven selectable transfer conditions. The additional evaluation step converts a pair of validity and non-validity degrees into the Boolean decision used by the existing token-transfer logic. Capacities, priorities, and the normal movement procedure remain part of the simulation, while the condition can be changed independently of the model's predicates.

The example with $\langle 0.4, 0.4 \rangle$ makes the difference between conditions C_4 and C_5 concrete. Strict comparison rejects the predicate; non-strict comparison accepts it. The included OnlineGN screenshots show the configuration controls and the displayed decisions for these cases. Further testing with larger nets, multiple eligible outputs, and capacity constraints would help characterize the behavior of the extension in more involved simulations. Other forms of intuitionistic fuzzy GNs, including fuzzy token characteristics and quantities, remain outside the present implementation.

References

[1] Atanassov, K. (1985). Generalized nets and their fuzzings. *AMSE Review*, *2*(3), 39–49.

[2] Dimitriev, A. (2026). *OnlineGN: Web-based Generalized Nets Simulator*. Available online at: `https://onlinegn.net/`

[3] Atanassov, K. T. (1986). Intuitionistic fuzzy sets. *Fuzzy Sets and Systems*, *20*(1), 87–96.

[4] Atanassov, K. T. (1991). *Generalized Nets*. World Scientific, Singapore.

[5] Atanassov, K. T. (1999). *Intuitionistic Fuzzy Sets: Theory and Applications*. Studies in Fuzziness and Soft Computing, Vol. 35. Physica-Verlag, Heidelberg.

[6] Atanassov, K., Dimitrov, D., & Atanassova, V. (2010). Algorithms for tokens transfer in different types of intuitionistic fuzzy generalized nets. *Cybernetics and Information Technologies*, 10(4), 22–35.

[7] Atanassov, K. T., & Nikolov, N. G. (2003). Intuitionistic fuzzy generalized nets: Definitions, properties, applications. In: *Melo-Pinto, P., Teodorescu, H.-N., & T. Fukuda (Eds.). Systematic Organisation of Information in Fuzzy Systems*, pp. 161–175. IOS Press, Amsterdam.

[8] Dimitriev, A., Atanassov, K., & Angelova, N. (2025). Parallel towers of Hanoi via generalized nets: Simulated with OnlineGN. *Software*, 4(4), Article ID 23.