

# Rule base identification in fuzzy networks by Boolean matrix equations

Alexander Gegov<sup>a,\*</sup>, Nedyalko Petrov<sup>a</sup> and Emil Gegov<sup>b</sup>

<sup>a</sup>*School of Computing Buckingham Building, University of Portsmouth, Portsmouth, UK*

<sup>b</sup>*School of Engineering and Design Kingston Lane, Brunel University, Uxbridge, UK*

**Abstract.** This paper proposes a novel approach for modelling complex interconnected systems by means of fuzzy networks. The nodes in these networks are interconnected rule bases whereby the outputs from some rule bases are fed as inputs to other rule bases. The approach allows any fuzzy network of this type to be presented as an equivalent fuzzy system by linguistic composition of its nodes. The composition process makes use of formal models for fuzzy networks and basic operations in such networks. These models and operations are used for defining several node identification cases in fuzzy networks. In this case, the unknown nodes are derived by solving Boolean matrix equations in a way that guarantees a pre-specified overall performance of the network. The main advantage of the proposed approach over other approaches is that it has better transparency and facilitates not only the analysis but also the design of complex interconnected systems.

**Keywords:** Fuzzy modelling, linguistic modelling, fuzzy networks, rule base identification, complex systems

## 1. Introduction

Complexity is a versatile feature of existing systems that cannot be described by a single definition. In this context, complexity is usually associated with a number of attributes such as uncertainty, dimensionality and structure, which make the modelling of systems with these attributes more difficult. Therefore, the complexity of a given system can be accounted for by identifying the complexity related attributes that are to be found in this system.

Fuzzy logic has proved itself as a powerful tool for dealing with uncertainty as an attribute of systemic complexity. In this context, fuzziness is quite suitable for reflecting non-probabilistic uncertainty such as imprecision, incompleteness and ambiguity [1, 14, 15, 18].

More recently, fuzzy logic has also become more effective in dealing with dimensionality as a systemic

complexity attribute by means of rule base reduction and compression. Dimensionality in rule base reduction is associated with the number of rules, which is an exponential function of the number of system inputs and the number of linguistic terms per input [6, 7, 35, 36]. In rule base compression, dimensionality is associated with the amount of on-line operations required during fuzzification, inference and defuzzification [3].

However, as far as structure is concerned, fuzzy logic is still unable to reflect adequately any interacting modules within a modelled process. This is due to the black-box nature of most fuzzy models that cannot take into account explicitly any interactions among sub-processes [9, 16, 20, 33]. In this respect, the following paragraphs discuss some of the main approaches in fuzzy modelling and their ability to deal with structure as a systemic complexity attribute.

The most common type of fuzzy system is with a single rule base [5, 17, 34]. This type of system is referred to here as Standard Fuzzy System (SFS). The latter is characterised by a black-box nature whereby the inputs are mapped directly to the outputs without

\*Corresponding author. Alexander Gegov, School of Computing Buckingham Building, University of Portsmouth, Portsmouth PO1 3HE, UK. E-mail: alexander.gegov@port.ac.uk.

the consideration of any internal connections. The operation of SFS is based on a single Fuzzification-Inference-Defuzzification (FID) sequence and it is usually quite accurate for output modelling as it reflects the simultaneous influence of all inputs on the output. However, the efficiency and transparency of SFS deteriorate with the increase of the number of rules. Therefore, as the number of rules increases, it not only takes longer to simulate the model output but it is also less clear how this output is affected by the model inputs.

Another type of fuzzy system is with multiple rule bases [10, 19, 39, 40]. This type of system is usually described by cascaded rule bases and referred to as Chained Fuzzy System (CFS) or Hierarchical Fuzzy System (HFS). Both CFS and HFS are characterised by a white-box nature whereby the inputs are mapped to the outputs by means of some internal variables in the form of connections. The operation of CFS and HFS is based on multiple FID sequences whereby each connection links the FID sequences for two adjacent rule bases.

CFS has an arbitrary structure in terms of subsystems and the connections among them [8, 21, 22]. In this case, each subsystem represents an individual rule base whereas each interaction is represented by a connection linking a pair of adjacent rule bases. This connection is identical with an output from the first rule base and an input to the second rule base in the pair. CFS is usually used as a detailed presentation of SFS for the purpose of improving transparency by explicitly taking into account all subsystems and the interactions among them. Also, efficiency is improved because of the smaller number of inputs to the individual rule bases. However, accuracy may be lost due to the accumulation of errors as a result of the multiple FID sequences.

HFS is a special type of CFS that has a specific structure [11, 23, 26, 30, 37]. Each subsystem in HFS has two inputs and one output. Some connections represent identical mappings, which may propagate across parts of the system. HFS is often used as an alternative presentation of SFS for the purpose of improving transparency by explicitly taking into account all subsystems and the interactions among them. Efficiency is also improved by the reduction of the overall number of rules, which is a linear function of the number of inputs to the subsystems and the number of linguistic terms per input. However, these improvements usually lead to loss of accuracy due to the accumulation of errors as a result of the multiple FID sequences.

A third type of fuzzy system is with networked rule bases and it is referred to here as Networked Fuzzy System (NFS). This type of system has been introduced recently in [4]. NFS is characterised by a white-box nature whereby the inputs are mapped to the outputs by means of connections. Subsystems in NFS are represented by nodes and the interactions among subsystems are the connections among these nodes. NFS is a hybrid between SFS and CFS/HFS. On one hand, the structure of NFS is similar to the structure of CFS/HFS due to the explicit presentation of subsystems and the interactions among them. On the other hand, the operation of NFS resembles the operation of SFS as the multiple rule bases are simplified to a linguistically equivalent single rule base. This simplification is based on the linguistic composition approach that is the main focus of this work. As a hybrid concept, NFS has the potential of combining the advantages of SFS and CFS/HFS.

Properties of fuzzy systems such as accuracy, efficiency and transparency are directly related to attributes of systemic complexity such as uncertainty, dimensionality and structure. In this respect, uncertainty is an obstacle to accuracy as it is harder to build an accurate model from uncertain data [12, 13, 27, 38]. Furthermore, dimensionality represents an obstacle to efficiency because it is more difficult to reduce the amount of computations in a FID sequence for a large number of rules [24, 28, 29]. Finally, structure is an obstacle to transparency as it is harder to understand the behaviour of a black-box model that does not reflect the interactions among subsystems [2, 25, 31, 32].

This paper introduces a theoretical framework for NFS as a novel type of fuzzy system. It also demonstrates the capabilities of NFS as a modelling and design tool for complex systems. For clarity and simplicity, NFS is referred to as Fuzzy Network (FN). The paper addresses mainly structure as a systemic complexity attribute and the associated property of transparency. The main reason for this choice is that transparency has always been given less attention in complex systems modelling as opposed to accuracy and efficiency. Besides this, transparency has recently turned out to be almost as important for complex systems modelling as accuracy and efficiency.

The remaining part of this paper is structured as follows. Section 2 introduces formal models for fuzzy networks. Section 3 presents basic operations in fuzzy networks. Section 4 discusses the theoretical fundamentals of several node identification cases in fuzzy networks. Section 5 illustrates the application of these theoretical fundamentals to several examples. Section

6 summarises the main advantages of the approach and highlights future research directions.

## 2. Formal models for fuzzy networks

A fuzzy system with  $r$  rules,  $m$  inputs  $x_1 \dots x_m$  taking linguistic terms from the input sets  $\{A_{11}, \dots, A_{1r}\}, \dots, \{A_{m1}, \dots, A_{mr}\}$  and  $n$  outputs  $y_1 \dots y_n$  taking linguistic terms from the output sets  $\{B_{11}, \dots, B_{1r}\}, \dots, \{B_{n1}, \dots, B_{nr}\}$  can be described by the following if-then rules:

*Rule 1 : If  $x_1$  is  $A_{11}$  and ... and  $x_m$  is  $A_{m1}$ ,  
then  $y_1$  is  $B_{11}$  and ... and  $y_n$  is  $B_{n1}$*

.....

*Rule  $r$  : If  $x_1$  is  $A_{1r}$  and ... and  $x_m$  is  $A_{mr}$ ,  
then  $y_1$  is  $B_{1r}$  and ... and  $y_n$  is  $B_{nr}$*  (1)

As a fuzzy network represents an extension of a fuzzy system, i.e. it can be viewed as a system of fuzzy systems or a network whose nodes are fuzzy systems, some of the general formal models for fuzzy systems can be used also for fuzzy networks. However, other formal models that are specific to fuzzy networks are required for the simplification of a fuzzy network to a linguistically equivalent fuzzy system. Most of these formal models contain compressed information about nodes in fuzzy networks and they are discussed further below.

If-then rules as the ones from Equation (1) are established formal models for fuzzy systems that can represent nodes in a FN without the connections. They are used here as a bridge between fuzzy systems and FNs. For example, a FN with four nodes  $N_{11}, N_{12}, N_{21}, N_{22}$  can be described by the if-then rules given in Equations (2)–(13).

*Rule 1 for  $N_{11}$  : If  $x_{11}$  is small, then  $y_{11}$  is low* (2)

*Rule 2 for  $N_{11}$  : If  $x_{11}$  is medium, then  $y_{11}$  is high* (3)

*Rule 3 for  $N_{11}$  : If  $x_{11}$  is big, then  $y_{11}$  is average* (4)

*Rule 1 for  $N_{12}$  : If  $x_{12}$  is low, then  $y_{12}$  is moderate* (5)

*Rule 2 for  $N_{12}$  : If  $x_{12}$  is average, then  $y_{12}$  is heavy* (6)

*Rule 3 for  $N_{12}$  : If  $x_{12}$  is high, then  $y_{12}$  is light* (7)

*Rule 1 for  $N_{21}$  : If  $x_{21}$  is small, then  $y_{21}$  is average* (8)

*Rule 2 for  $N_{21}$  : If  $x_{21}$  is medium, then  $y_{21}$  is low* (9)

*Rule 3 for  $N_{21}$  : If  $x_{21}$  is big, then  $y_{21}$  is high* (10)

*Rule 1 for  $N_{22}$  : If  $x_{22}$  is low, then  $y_{22}$  is heavy* (11)

*Rule 2 for  $N_{22}$  : If  $x_{22}$  is average, then  $y_{22}$  is light* (12)

*Rule 3 for  $N_{22}$  : If  $x_{22}$  is high, then  $y_{22}$  is moderate* (13)

For compactness, the linguistic terms of the inputs and the outputs for the four nodes above can also be represented by positive integers. In this case, ‘small’, ‘low’, ‘light’ can be represented by ‘1’, ‘medium’, ‘average’, ‘moderate’ can be represented by ‘2’ and ‘big’, ‘high’, ‘heavy’ can be represented by ‘3’.

If-then rules as the ones presented above are very suitable for formal modelling of fuzzy systems with a single rule base such as SFSs. However, they are not quite suitable for formal modelling of fuzzy systems with multiple or networked rule bases. This is due to the fact that if-then rules cannot take into account any connections among nodes in networked rule bases. Also, if-then rules do not lend themselves easily to manipulation for the purpose of simplifying networked rule bases to a linguistically equivalent single rule base using the linguistic composition approach.

Boolean matrices are novel formal models for fuzzy systems that can represent nodes in a FN. Similarly to if-then rules, these models can represent nodes without the connections.

A Boolean matrix compresses the information from a rule base that is represented by a node. In this case, the row and column labels of the Boolean matrix are all possible permutations of the positive integers representing the linguistic terms of the inputs and the outputs for this rule base. The elements of the Boolean matrix are either zeros or ones whereby each one and zero reflects an existing and missing rule, respectively.

The if-then rules for the fuzzy network nodes  $N_{11}, N_{12}, N_{21}, N_{22}$  from Equations (2)–(13) can be described by the Boolean matrices in Equations (14)–(17).

$$\begin{array}{l}
 N_{11} : \quad y_{11} \quad 1 \quad 2 \quad 3 \\
 \quad \quad x_{11} \\
 \quad \quad 1 \quad 1 \quad 0 \quad 0 \\
 \quad \quad 2 \quad 0 \quad 0 \quad 1 \\
 \quad \quad 3 \quad 0 \quad 1 \quad 0
 \end{array} \quad (14)$$

$$\begin{array}{l}
 \text{Layer1} \quad \text{Layer2} \\
 \text{Level1} \quad N_{11}(x_{11}, y_{11}) \quad N_{12}(x_{12}, y_{12}) \\
 \text{Level2} \quad N_{21}(x_{21}, y_{21}) \quad N_{22}(x_{22}, y_{22})
 \end{array} \quad (18)$$

$$\begin{array}{l}
 N_{12} : \quad y_{12} \quad 1 \quad 2 \quad 3 \\
 \quad \quad x_{12} \\
 \quad \quad 1 \quad 0 \quad 1 \quad 0 \\
 \quad \quad 2 \quad 0 \quad 0 \quad 1 \\
 \quad \quad 3 \quad 1 \quad 0 \quad 0
 \end{array} \quad (15)$$

$$\begin{array}{l}
 N_{21} : \quad y_{21} \quad 1 \quad 2 \quad 3 \\
 \quad \quad x_{21} \\
 \quad \quad 1 \quad 0 \quad 1 \quad 0 \\
 \quad \quad 2 \quad 1 \quad 0 \quad 0 \\
 \quad \quad 3 \quad 0 \quad 0 \quad 1
 \end{array} \quad (16)$$

$$\begin{array}{l}
 \text{Layer1} \\
 \text{Level1} \quad z_{11,12} = y_{11} = x_{12} \\
 \text{Level2} \quad z_{21,22} = y_{21} = x_{22}
 \end{array} \quad (19)$$

$$\begin{array}{l}
 N_{22} : \quad y_{22} \quad 1 \quad 2 \quad 3 \\
 \quad \quad x_{22} \\
 \quad \quad 1 \quad 0 \quad 0 \quad 1 \\
 \quad \quad 2 \quad 1 \quad 0 \quad 0 \\
 \quad \quad 3 \quad 0 \quad 1 \quad 0
 \end{array} \quad (17)$$

The location structure above is a formal model for a FN with a node set  $\{N_{11}, N_{12}, N_{21}, N_{22}\}$ , an input set  $\{x_{11}, x_{12}, x_{21}, x_{22}\}$  and an output set  $\{y_{11}, y_{12}, y_{21}, y_{22}\}$ . This structure specifies the location of nodes as well as their inputs and outputs.

If the nodes  $N_{11}, N_{12}, N_{21}, N_{22}$  from Equations (2)–(13) are connected, their connections can be described by the connection set  $\{z_{11,12}, z_{21,22}\}$ . In this case, the first connection is identical with the output from  $N_{11}$  and the input to  $N_{12}$  whereas the second connection is identical with the output from  $N_{21}$  and the input to  $N_{22}$ . These connections can be described by the connection structure with two levels and one layer in Equation (19).

Location and connection structures as the ones presented above are also quite suitable for formal modelling of fuzzy systems with multiple or networked rule bases. In particular, these structures are well suited for formal modelling of FNs at a higher level of abstraction whereby only locations, inputs, outputs and connections for individual nodes are specified.

Location and connection structures describe FNs at overall network level. They work well with Boolean matrices which describe FNs at individual node level. However, these structures do not lend themselves easily to manipulation for the purpose of simplifying networked rule bases to a linguistically equivalent single rule base using the linguistic composition approach.

Block schemes and topological expressions are also novel formal models that are like compressed images of a FN. Similarly to location and connection structures, these models describe the location of nodes and the connections among them. In this case, the subscripts of each node specify its location in the network whereby the first subscript gives the level number and the second subscript gives the layer number. Besides this, block schemes and topological expressions specify all inputs, outputs and connections with respect to the nodes. For example, the four-node FN from Equations (2)–(13) and Equations (18) and (19) can be described by the block scheme in Fig. 1.

Boolean matrices as the ones presented above are very suitable for formal modelling of fuzzy systems with multiple or networked rule bases. In particular, they are well suited for formal modelling of FNs at a lower level of abstraction whereby detailed input-output mappings are specified for isolated individual nodes. Besides this, Boolean matrices work well with other formal models which can take into account connections among nodes in FNs.

Location and connection structures are other novel formal models that are like compressed images of a FN. These models describe the location of nodes and the connections among them, respectively. For example, the four nodes  $N_{11}, N_{12}, N_{21}, N_{22}$  from Equations (2)–(13) can be described by the location structure with two levels and two layers in Equation (18).

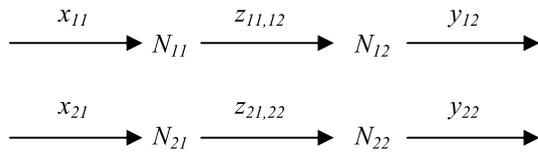


Fig. 1. Block scheme for a four-node fuzzy network.

The arrows in the block scheme above designate the input set  $\{x_{11}, x_{21}\}$  for the nodes in the first layer and the output set  $\{y_{12}, y_{22}\}$  for the nodes in the second layer. Also, the arrows designate the connection set  $\{z_{11,12}, z_{21,22}\}$  for connected pairs of nodes whereby for each pair of nodes the first node is in the first layer and the second node is in the second layer.

The FN from the four-node FN from Equations (2)–(13) and Equations (18) and (19) can also be described by the topological expression in Equation (20).

$$\begin{aligned} & \{[N_{11}](x_{11}|z_{11,12})H[N_{12}](z_{11,12}|y_{12})\}V \\ & \{[N_{21}](x_{21}|z_{21,22})H[N_{22}](z_{21,22}|y_{22})\} \end{aligned} \quad (20)$$

Each node in the topological expression above is placed within a pair of square brackets ‘[ ]’. The inputs and the outputs for each node are placed within a pair of simple brackets ‘( )’ right after the node. In this case, the inputs are separated from the outputs by a vertical slash ‘|’. Nodes in sequence are designated by the symbol ‘H’ for horizontal relative location whereas nodes in parallel are designated by the symbol ‘V’ for vertical relative location. In this case, the higher priority of horizontal relative location with respect to vertical relative location in Equation (20) is specified by pairs of curly brackets ‘{ }’.

Block schemes and topological expressions as the ones presented above are very suitable for formal modelling of fuzzy systems with multiple or networked rule bases. In particular, they are well suited for formal modelling of FNs at a higher level of abstraction whereby only locations, inputs, outputs and connections for individual nodes are specified.

Like location and connection structures, block schemes and topological expressions describe FNs at overall network level. They work well with Boolean matrices which describe FNs at individual node level. Besides this, block schemes and topological expressions lend themselves easily to manipulation for the purpose of simplifying networked rule bases to a linguistically equivalent single rule base using the linguistic composition approach.

This work focuses on Boolean matrices for formal modelling of nodes. As far as formal modelling of connections is concerned, the focus is on block schemes and topological expressions. The choice of these formal models is justified by their better suitability for the use of the linguistic composition approach in comparison to if-then rules, location and connection structures

### 3. Basic operations in fuzzy networks

The process of simplifying networked rule bases to a linguistically equivalent single rule base is central to the linguistic composition approach used in this work. This approach is based on three basic operations on nodes –horizontal, vertical and output merging. These operations are binary in that they can be applied to a pair of nodes, i.e. they are like elementary building blocks in the process of simplifying a FN to a fuzzy system. For simplicity, all basic operations are illustrated below by theoretical examples of nodes with scalar inputs, outputs and connections but their extension to the vector case is straightforward.

#### 3.1. Horizontal merging operation

Horizontal merging is a binary operation that can be applied to a pair of sequential nodes, i.e. nodes located in the same level of a FN. This operation merges the operand nodes from the pair into a single product node. The operation can be applied when the output from the first node is fed forward as an input to the second node in the form of an intermediate variable. In this case, the product node has the same input as the input to the first operand node and the same output as the output from the second operand node whereas the connection does not appear in the product node.

When Boolean matrices are used as formal models for the operand nodes, the horizontal merging operation is identical with Boolean matrix multiplication. The latter is similar to conventional matrix multiplication whereby each arithmetic multiplication is replaced by a ‘minimum’ operation and each arithmetic addition is replaced by a ‘maximum’ operation. In this case, the row labels of the product matrix are the same as the row labels of the first operand matrix whereas the column labels of the product matrix are the same as the column labels of the second operand matrix.

This example considers the sequential operand nodes  $N_{11}$  and  $N_{12}$  located in the first level of the four-node FN from Fig. 1. These nodes are described there by



Fig. 2. Two-node fuzzy network with operand nodes  $N_{11}$  and  $N_{12}$ .

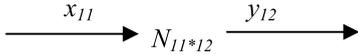


Fig. 3. One-node fuzzy network with product node  $N_{11*12}$ .

the Boolean matrices in Equations (14) and (15). The connections among these nodes are given by the connection structure in Equation (19). In this context, nodes  $N_{11}$  and  $N_{12}$  represent a two-node FN that is a subnetwork of the four-node FN. This two-node FN can be described by the block-scheme in Fig. 2 and the topological expression in Equation (21).

$$[N_{11}](x_{11}|z_{11,12}) * [N_{12}](z_{11,12}|y_{12}) \quad (21)$$

The use of the symbol ‘\*’ in Fig. 2 and Equation (21) implies that the horizontal merging operation can be applied to the operand nodes  $N_{11}$  and  $N_{12}$ . This operation results into a single product node  $N_{11*12}$  which represents a simplified image of the two-node FN in the form of a one-node FN. The latter can be described by the block scheme in Fig. 3 and the topological expression in Equation (22).

$$[N_{11*12}](x_{11}|y_{12}) \quad (22)$$

The use of the symbol ‘\*’ in Fig. 3 and Equation (22) implies that the application of the horizontal merging operation has resulted in the product node  $N_{11*12}$ . This node can be described by the Boolean matrix in Equation (23).

$$N_{11*12} : \begin{matrix} & y_{12} & 1 & 2 & 3 \\ x_{11} & & & & \\ 1 & 0 & 1 & 0 & \\ 2 & 1 & 0 & 0 & \\ 3 & 0 & 0 & 1 & \end{matrix} \quad (23)$$

### 3.2. Vertical merging operation

Vertical merging is a binary operation that can be applied to a pair of parallel nodes, i.e. nodes located in the same layer of a FN. This operation merges the operand nodes from the pair into a single product node. In this case, the inputs to the product node represent the union of the inputs to the operand nodes whereas the

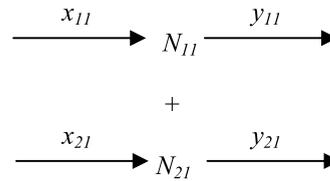


Fig. 4. Two-node fuzzy network with operand nodes  $N_{11}$  and  $N_{21}$ .

outputs from the product node represent the union of the outputs from the operand nodes. The operation of vertical merging can always be applied due to the ability to concatenate the inputs and the outputs of any pair of parallel nodes with independent inputs and outputs.

When Boolean matrices are used as formal models for the operand nodes, the vertical merging operation is like an expansion of the first operand matrix along its rows and columns. In particular, the product matrix is obtained by expanding each non-zero element from the first operand matrix to a block that is the same as the second operand matrix and by expanding each zero element from the first operand matrix to a zero block of the same dimension as the second operand matrix. In this case, the row labels of the product matrix are all possible permutations of row labels of the operand matrices whereas the column labels of the product matrix are all permutations of column labels of the operand matrices.

This example considers the parallel operand nodes  $N_{11}$  and  $N_{21}$  located in the first layer of the four-node FN from Fig. 1. These nodes are described there by the Boolean matrices in Equations (14) and (16). The connections of these nodes with the nodes in the second layer of this FN are given by the connection structure in Equation (19). In this context, nodes  $N_{11}$  and  $N_{21}$  represent a two-node subnetwork of this FN. This two-node FN can be described by the block-scheme in Fig. 4 and the topological expression in Equation (24).

$$[N_{11}](x_{11}|y_{11}) + [N_{21}](x_{21}|y_{21}) \quad (24)$$

The use of the symbol ‘+’ in Fig. 4 and Equation (24) implies that the vertical merging operation can be applied to the operand nodes  $N_{11}$  and  $N_{21}$ . This operation results into a single product node  $N_{11+21}$  which represents a simplified image of the two-node FN in the form of a one-node FN. The latter can be described by the block scheme in Fig. 5 and the topological expression in Equation (25).

$$[N_{11+12}](x_{11}, x_{21}|y_{11}, y_{21}) \quad (25)$$

The use of the symbol ‘+’ in Fig. 5 and Equation (25) implies that the application of the vertical merging

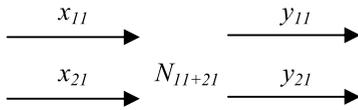


Fig. 5. One-node fuzzy network with product node  $N_{11+21}$ .

operation has resulted in the product node  $N_{11+12}$ . This node can be described by the Boolean matrix in Equation (26).

$$\begin{matrix}
 N_{11+12} : \\
 y_{11}, y_{21} & 11 & 12 & 13 & 21 & 22 & 23 & 31 & 32 & 33 \\
 x_{11}, x_{21} & & & & & & & & & \\
 11 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 12 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 13 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 21 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 22 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 31 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 32 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 33 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
 \end{matrix} \quad (26)$$

### 3.3. Output merging operation

Output merging is a binary operation that can be applied to a pair of parallel nodes with common inputs. This operation merges the operand nodes from the pair into a single product node. In this case, the inputs to the product node are the same as the common inputs to the operand nodes whereas the outputs from the product node represent the union of the outputs from the operand nodes. The operation of output merging can always be applied due to the ability to concatenate the outputs of any pair of parallel nodes with common inputs.

When Boolean matrices are used as formal models for the operand nodes, the output merging operation is like an expansion of the first operand matrix along its columns. In particular, the product matrix is obtained by expanding each non-zero element from the first operand matrix to a row-block that is the same as the corresponding row of the second operand matrix and by expanding each zero element from the first operand matrix to a zero row-block of the same dimension as the rows of the

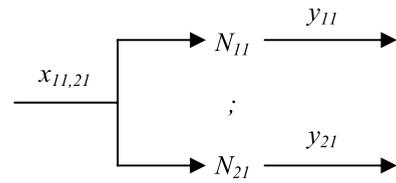


Fig. 6. Two-node fuzzy network with operand nodes  $N_{11}, N_{21}$  and common input.

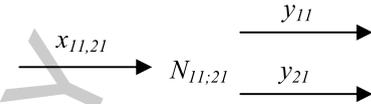


Fig. 7. One-node fuzzy network with product node  $N_{11;21}$ .

second product matrix. In this case, the row labels of the product matrix are the same as the identical row labels of the operand matrices whereas the column labels of the product matrix are all possible permutations of column labels of the operand matrices.

This example considers the parallel operand nodes  $N_{11}$  and  $N_{21}$  located in the first layer of the four-node FN from Fig. 1 in a modified context. In particular, the two independent inputs  $x_{11}$  and  $x_{21}$  to these nodes are replaced by a common input  $x_{11,21}$ . The nodes are described by the Boolean matrices in Equations (14) and (16). The connections of these nodes with the nodes in the second layer of this FN are given by the connection structure in Equation (19). In this context, the nodes  $N_{11}$  and  $N_{21}$  represent a modified two-node subnetwork of this FN. This two-node FN can be described by the block-scheme in Fig. 6 and the topological expression in Equation (27).

$$[N_{11}](x_{11,21}|y_{11}); [N_{21}](x_{11,21}|y_{21}) \quad (27)$$

The use of the symbol ‘;’ in Fig. 6 and Equation (27) implies that the output merging operation can be applied to the operand nodes  $N_{11}$  and  $N_{21}$ . This operation results into a single product node  $N_{11;21}$  which represents a simplified image of the two-node FN in the form of a one-node FN. The latter can be described by the block scheme in Fig. 7 and the topological expression in Equation (28).

$$[N_{11;12}](x_{11,21}|y_{11}, y_{21}) \quad (28)$$

The use of the symbol ‘;’ in Fig. 7 and Equation (28) implies that the application of the output merging operation has resulted in the product node  $N_{11;12}$ . This node can be described by the Boolean matrix in Equation (29).

$$\begin{array}{l}
 N_{11;12} : \\
 y_{11}, y_{21} \quad 11 \ 12 \quad 13 \ 21 \ 22 \quad 23 \ 31 \ 32 \ 33 \\
 x_{11}, x_{21} \\
 \begin{array}{cccccccc}
 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
 \end{array} \\
 \end{array} \tag{29}$$

**4. Theoretical fundamentals**

The assumption for all basic operations is that the two operand nodes are known and the result of the operation applied to them is the product node. This is typical for network analysis where the task is to see how the network behaves as a whole with all nodes given. However, in the case of network design some nodes are not known and the task is to identify them such that the overall network behaves in a pre-specified way. In this context, the remaining paragraphs of this section consider the theoretical fundamentals of this type of rule base identification in the cases of horizontal, vertical and output merging whereby the first operand node and the product node are known but the second operand node is unknown.

*4.1. Horizontal merging case*

Node identification is usually applied in horizontal merging when one or more nodes in the same level of a FN are unknown but the node for the equivalent fuzzy system for this level is known. In this case, it is necessary to find the unknown nodes from the other nodes in this level and the node for the equivalent fuzzy system. The purpose of this type of node identification is to ensure that once the unknown nodes have been identified and horizontally merged with the known nodes, the resultant node is identical with the node for the equivalent fuzzy system. In this context, node identification in horizontal merging can not always be applied as there is no guarantee for a solution to exist in accordance with the above requirement. However, when a solution can be found it may not be unique and this allows the node identification process to be optimised with respect to the performance of the equivalent fuzzy system.

When Boolean matrices are used as formal models during node identification in horizontal merging, this process is based on solving systems of Boolean

equations. In this case, the known coefficients in these systems of equations are the elements of the Boolean matrices for the known individual nodes in the associated level of the FN and the elements of the Boolean matrix for the known node of the equivalent fuzzy system whereas the unknown variables are the elements of the Boolean matrices for the unknown nodes.

Here, node identification is considered in the context of horizontal merging of two nodes *A* and *U* into a node *C* whereby only *A* and *C* are given. Therefore, it is necessary to identify the unknown node *U* on the basis of the known nodes *A* and *C*. This problem can be described in a general form by the Boolean matrix equation in Equation (30) where *A*, *U* and *C* represent the Boolean matrices for the above three nodes. The dimensions and the detailed descriptions of these Boolean matrices are given in Equations (31)–(33).

$$A * U = C \tag{30}$$

$$a_{11} \dots a_{1q} \tag{31}$$

$$\begin{array}{l}
 A_{p \times q} = \dots \dots \dots \\
 a_{p1} \dots a_{pq}
 \end{array}$$

$$u_{11} \dots u_{1r} \tag{32}$$

$$\begin{array}{l}
 U_{q \times r} = \dots \dots \dots \\
 u_{q1} \dots u_{qr}
 \end{array}$$

$$c_{11} \dots c_{1r} \tag{33}$$

$$\begin{array}{l}
 C_{p \times r} = \dots \dots \dots \\
 c_{p1} \dots c_{pr}
 \end{array}$$

The Boolean matrix equation in Equation (30) can be represented as a set of *r* systems of Boolean equations whereby each system consists of *p* equations and *q* unknowns. This set of systems of Boolean equations is given by Equation (34).

$$\begin{array}{l}
 \max[\min(a_{11}, u_{11}), \dots, \min(a_{1q}, u_{q1})] = c_{11} \\
 \dots \dots \dots \\
 \max[\min(a_{p1}, u_{11}), \dots, \min(a_{pq}, u_{q1})] = c_{p1} \\
 \dots \dots \dots \\
 \max[\min(a_{11}, u_{1r}), \dots, \min(a_{1q}, u_{qr})] = c_{1r} \\
 \dots \dots \dots \\
 \max[\min(a_{p1}, u_{1r}), \dots, \min(a_{pq}, u_{qr})] = c_{pr}
 \end{array} \tag{34}$$

The solution for each system of Boolean equations in Equation (34) represents a column in the unknown Boolean matrix  $U$  from the Boolean matrix equation in Equation (30). The most trivial way of solving each system of Boolean equations is to generate all possible permutations of 0s and 1s for the unknowns. In this case, a multiple solution is very likely to exist, especially if some columns in the Boolean matrix  $A$  contain only zero elements. These zero elements will have an overriding effect on the elements in the corresponding rows of the Boolean matrix  $U$ , i.e. the latter can be taken as either 0s or 1s in the solution. In particular, the variation  $V$  in the number of solutions for the Boolean matrix equation in Equation (30) is given by the general formula in Equation (35) where  $r$  is the number of columns in  $U$  and  $s$  is the number of zero columns in  $A$ .

$$V = r^s + 1 \tag{35}$$

#### 4.2. Vertical merging case

Node identification is usually applied in vertical merging when one or more nodes in the same layer of a FN are unknown but the node for the equivalent fuzzy system for this layer is known. In this case, it is necessary to find the unknown nodes from the other nodes in this layer and the node for the equivalent fuzzy system. The purpose of this type of node identification is to ensure that once the unknown nodes have been identified and vertically merged with the known nodes, the resultant node is identical with the node for the equivalent fuzzy system. In this context, node identification in vertical merging can not always be applied as there is no guarantee for a solution to exist in accordance with the above requirement. However, when a solution can be found, it is usually unique.

When Boolean matrices are used as formal models during node identification in vertical merging, this process is based on examining the structure of the known matrices. In this case, a location based correspondence is sought between the non-zero elements of the known Boolean matrices for the individual nodes and any identical non-zero blocks of the known Boolean matrix for the equivalent fuzzy system. If such a correspondence is to be found, then the unknown Boolean matrix is equal to these non-zero blocks or to a compressed image of this Boolean matrix whereby all non-zero and zero blocks are represented by 1s and 0s, respectively.

Here, node identification is considered in the context of vertical merging of two nodes  $A$  and  $U$  into a node  $C$  whereby only  $A$  and  $C$  are given. Therefore, it is neces-

sary to identify the unknown node  $U$  on the basis of the known nodes  $A$  and  $C$ . This problem can be described in a general form by the Boolean matrix equation in Equation (36) where  $A$ ,  $U$  and  $C$  represent the Boolean matrices for the above three nodes. The dimensions and the detailed descriptions of these Boolean matrices are given in Equations (37)–(39).

$$A + U = C \tag{36}$$

$$a_{11} \dots a_{1q} \tag{37}$$

$$Ap \times q = \dots \dots \dots a_{p1} \dots a_{pq}$$

$$u_{11} \dots u_{1s} \tag{38}$$

$$Ur \times s = \dots \dots \dots u_{r1} \dots u_{rs}$$

$$c_{11} \dots c_{1,r,s} \tag{39}$$

$$Cp.r \times q.s = \dots \dots \dots c_{p,r,1} \dots c_{p,r,q,s}$$

If the Boolean matrix  $C$  in Equation (36) contains only one set of not more than  $p$  identical non-zero blocks  $C^k$ ,  $1 \leq k \leq p$  whereby there is not more than one such block in any block row of this matrix, the elements with a location in  $A$  corresponding to the location of non-zero blocks  $C^k$  in  $C$  are all equal to 1 and all other elements with a location in  $A$  corresponding to the location of zero blocks  $C^0$  in  $C$  are equal to 0, then the Boolean matrix  $U$  is equal to  $C^k$ . In this case, the location of the elements of the non-zero identical blocks  $C^k$  in  $C$  can be described by Equation (40) which also shows the admissible initial values for the subscripts  $i$  and  $j$  of the elements of  $C^k$ . As far as the zero blocks  $C^0$  are concerned, they are with the same number of rows and columns as the non-zero blocks  $C^k$ .

$$c_{ij}^k \dots c_{i,j+s-1}^k \tag{40}$$

$$C^k = \dots \dots \dots, \quad i = 1, 1+r, \dots, 1+(p-1)r,$$

$$c_{i+r-1,j}^k \dots c_{i+r-1,j+s-1}^k, \quad j = 1, 1+s, \dots, 1+(q-1)s$$

#### 4.3. Output merging case

Node identification is usually applied in output merging when one or more nodes with common inputs in the

same layer of a FN are unknown but the node for the equivalent fuzzy system for this layer is known. In this case, it is necessary to find the unknown nodes from the other nodes in this layer and the node for the equivalent fuzzy system. The purpose of this type of node identification is to ensure that once the unknown nodes have been identified and output merged with the known nodes, the resultant node is identical with the node for the equivalent fuzzy system. In this context, node identification in output merging can not always be applied as there is no guarantee for a solution to exist in accordance with the above requirement. However, when a solution can be found, it is usually unique.

When Boolean matrices are used as formal models during node identification in output merging, this process is based on examining the structure of the known matrices. In this case, a location based correspondence is sought between the non-zero elements of the known Boolean matrices for the individual nodes and any non-zero row blocks of the known Boolean matrix for the equivalent fuzzy system. If such a correspondence is to be found, then the rows of the unknown Boolean matrix are equal to the non-zero row blocks or to a compressed image of this Boolean matrix whereby all non-zero and zero row blocks are represented by 1s and 0s, respectively.

Here, node identification is considered in the context of output merging of two nodes  $A$  and  $U$  with common inputs into a node  $C$  whereby only  $A$  and  $C$  are given. Therefore, it is necessary to identify the unknown node  $U$  on the basis of the known nodes  $A$  and  $C$ . This problem can be described in a general form by the Boolean matrix equation in Equation (41) where  $A$ ,  $U$  and  $C$  represent the Boolean matrices for the above three nodes. The dimensions and the detailed descriptions of these Boolean matrices are given in Equations (42)–(44).

$$A; U = C \tag{41}$$

$$A_{p \times q} = \begin{matrix} a_{11} \dots a_{1q} \\ \dots \\ a_{p1} \dots a_{pq} \end{matrix} \tag{42}$$

$$U_{p \times r} = \begin{matrix} u_{11} \dots u_{1r} \\ \dots \\ u_{p1} \dots u_{pr} \end{matrix} \tag{43}$$

$$C_{p \times q,r} = \begin{matrix} c_{11} \dots c_{1,q} \\ \dots \\ c_{p,1} \dots c_{p,q,r} \end{matrix} \tag{44}$$

If the Boolean matrix  $C$  in Equation (41) contains only one set of not more than  $p$  non-zero row blocks  $C^k$ ,  $1 \leq k \leq p$  whereby there is not more than one such block in any row of this matrix, the elements with a location in  $A$  corresponding to the location of non-zero row blocks  $C^k$  in  $C$  are all equal to 1 and all other elements with a location in  $A$  corresponding to the location of zero row blocks  $C^0$  in  $C$  are equal to 0, then the rows of the Boolean matrix  $U$  are equal to  $C^k$ . In this case, the location of the elements of the non-zero row blocks  $C^k$  in  $C$  can be described by Equation (45) which also shows the admissible initial values for the subscripts  $i$  and  $j$  of the elements of  $C^k$ . As far as the zero row blocks  $C^0$  are concerned, they are with the same number of elements as the non-zero row blocks  $C^k$ .

$$C^k = c_{ij}^k \dots c_{i+r-1,j}^k, i = 1, p, j = 1, 1+r, \dots, 1+(q-1).r \tag{45}$$

### 5. Application examples

The theoretical fundamentals from the three cases in the previous section are illustrated below by application examples. In particular, one example is shown for each case.

#### 5.1. Horizontal merging example

This example considers a FN with two sequential nodes  $A$  and  $U$  whereby  $\{x_{A1}, x_{A2}\}$  is the input set for  $A$ ,  $z_{A,U}$  is the intermediate variable and  $\{y_{U1}, y_{U2}\}$  is the output set for  $U$ . These nodes are horizontally merged into node  $C$  that represents the equivalent fuzzy system for this FN, as shown by the block-scheme in Fig. 8 and the topological expression in Equation (46). The known nodes  $A$  and  $C$  are described by the Boolean matrices in Equations (47) and (48).

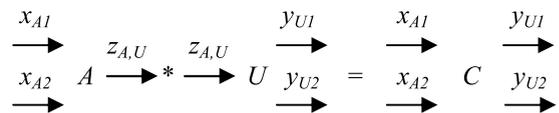


Fig. 8. Fuzzy network with nodes  $A$  and  $U$  horizontally merged into node  $C$ .

$$[A](x_{A1}, x_{A2} | z_{A,U}) * [U](z_{A,U} | y_{U1}, y_{U2}) = [C](x_{A1}, x_{A2} | y_{U1}, y_{U2}) \quad (46)$$

A :	$z_{A,U}$	1	2	3	
	$x_{A1}, x_{A2}$				
	11	1	0	0	(47)
	12	1	0	0	
	21	0	0	1	
	22	0	0	1	

C :	$y_{U1}, y_{U2}$	11	12	21	22	
	$x_{A1}, x_{A2}$					
	11	0	0	0	1	(48)
	12	0	0	0	1	
	21	1	0	0	0	
	22	1	0	0	0	

The columns of the Boolean matrix for the unknown node  $U$  can be found from Equation (34) which is presented in a detailed form for this example by Equations (49)–(52).

$$\begin{aligned} \max[\min(1, u_{11}), \min(0, u_{21}), \min(0, u_{31})] &= 0 \\ \max[\min(1, u_{11}), \min(0, u_{21}), \min(0, u_{31})] &= 0 \\ \max[\min(0, u_{11}), \min(0, u_{21}), \min(1, u_{31})] &= 1 \\ \max[\min(0, u_{11}), \min(0, u_{21}), \min(1, u_{31})] &= 1 \end{aligned} \quad (49)$$

$$\begin{aligned} \max[\min(1, u_{12}), \min(0, u_{22}), \min(0, u_{32})] &= 0 \\ \max[\min(1, u_{12}), \min(0, u_{22}), \min(0, u_{32})] &= 0 \\ \max[\min(0, u_{12}), \min(0, u_{22}), \min(1, u_{32})] &= 0 \\ \max[\min(0, u_{12}), \min(0, u_{22}), \min(1, u_{32})] &= 0 \end{aligned} \quad (50)$$

$$\begin{aligned} \max[\min(1, u_{13}), \min(0, u_{23}), \min(0, u_{33})] &= 0 \\ \max[\min(1, u_{13}), \min(0, u_{23}), \min(0, u_{33})] &= 0 \\ \max[\min(0, u_{13}), \min(0, u_{23}), \min(1, u_{33})] &= 0 \\ \max[\min(0, u_{13}), \min(0, u_{23}), \min(1, u_{33})] &= 0 \end{aligned} \quad (51)$$

$$\begin{aligned} \max[\min(1, u_{14}), \min(0, u_{24}), \min(0, u_{34})] &= 1 \\ \max[\min(1, u_{14}), \min(0, u_{24}), \min(0, u_{34})] &= 1 \\ \max[\min(0, u_{14}), \min(0, u_{24}), \min(1, u_{34})] &= 0 \\ \max[\min(0, u_{14}), \min(0, u_{24}), \min(1, u_{34})] &= 0 \end{aligned} \quad (52)$$

The variation in the number of solutions for the set of 4 systems of Boolean equations in Equations (49)–(52) with 4 equations and 3 unknowns each can be found from Equation (35) which is presented in a specific form for this example by Equation (53).

$$4^1 + 1 = 5 \quad (53)$$

The solution for node  $U$  is given by the Boolean matrices in Equations (54)–(58). In this case, each subscript  $i = 1, 5$  for  $U$  represents an individual solution from the solution set. Each individual solution represents a consistent rule base, i.e. a rule base whose Boolean matrix has not more than one non-zero element in each row.

$U_1 :$	$y_{U1}, y_{U2}$	11	12	21	22	
	$z_{A,U}$					
	1	0	0	0	1	(54)
	2	0	0	0	0	
	3	1	0	0	0	

$U_2 :$	$y_{U1}, y_{U2}$	11	12	21	22	
	$z_{A,U}$					
	1	0	0	0	1	(55)
	2	1	0	0	0	
	3	1	0	0	0	

$U_3 :$	$y_{U1}, y_{U2}$	11	12	21	22	
	$z_{A,U}$					
	1	0	0	0	1	(56)
	2	0	1	0	0	
	3	1	0	0	0	

$$\begin{array}{l}
 U_4 : \quad y_{U1}, y_{U2} \quad 11 \quad 12 \quad 21 \quad 22 \\
 z_{A,U} \\
 1 \quad \quad \quad 0 \quad 0 \quad 0 \quad 1 \quad (57) \\
 2 \quad \quad \quad 0 \quad 0 \quad 1 \quad 0 \\
 3 \quad \quad \quad 1 \quad 0 \quad 0 \quad 0
 \end{array}$$

$$\begin{array}{l}
 U_5 : \quad y_{U1}, y_{U2} \quad 11 \quad 12 \quad 21 \quad 22 \\
 z_{A,U} \\
 1 \quad \quad \quad 0 \quad 0 \quad 0 \quad 1 \quad (58) \\
 2 \quad \quad \quad 0 \quad 0 \quad 0 \quad 1 \\
 3 \quad \quad \quad 1 \quad 0 \quad 0 \quad 0
 \end{array}$$

$$\begin{array}{l}
 C : \quad y_A, y_U \quad 11 \quad 12 \quad 13 \quad 21 \quad 22 \quad 23 \quad 31 \quad 32 \quad 33 \\
 x_A, x_U \\
 11 \quad \quad \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \\
 12 \quad \quad \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \\
 13 \quad \quad \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \\
 21 \quad \quad \quad 1 \quad 0 \\
 22 \quad \quad \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \\
 23 \quad \quad \quad 0 \quad 1 \quad 0 \\
 31 \quad \quad \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \\
 32 \quad \quad \quad 0 \quad 1 \\
 33 \quad \quad \quad 0 \quad 1 \quad 0
 \end{array} \quad (61)$$

5.2. Vertical merging example

This example considers a FN with two parallel nodes A and U whereby  $x_A$  is the input to A,  $y_A$  is the output from A,  $x_U$  is the input to U and  $y_U$  is the output from U. These nodes are vertically merged into node C that represents the equivalent fuzzy system for this FN, as shown by the block-scheme in Fig. 9 and the topological expression in Equation (59). The known nodes A and C are described by the Boolean matrices in Equations (60) and (61).

The Boolean matrix C in Equation (61) contains only one set of 3 identical non-zero blocks  $C^k$  whereby there is not more than one such block in any block row of this matrix. Also, the elements with a location in A corresponding to the location of non-zero blocks  $C^k$  in C are all equal to 1 and all other elements with a location in A corresponding to the location of zero blocks  $C^0$  in C are equal to 0.

Therefore, the Boolean matrix U is equal to  $C^k$ , as shown by the Boolean matrix in Equation (62). In this case, the location of the elements of  $C^k, k = 1, 2, 3$  in C is given by Equations (63)–(65).

$$\begin{aligned}
 & [A](x_A|y_A) + [U](x_U|y_U) \\
 & = [C](x_A, x_U|y_A, y_U) \quad (59)
 \end{aligned}$$

$$\begin{array}{l}
 A : \quad y_A \quad 1 \quad 2 \quad 3 \\
 x_A \\
 1 \quad \quad \quad 0 \quad 1 \quad 0 \quad (60) \\
 2 \quad \quad \quad 1 \quad 0 \quad 0 \\
 3 \quad \quad \quad 0 \quad 0 \quad 1
 \end{array}$$

$$\begin{array}{l}
 U : \quad y_U \quad 1 \quad 2 \quad 3 \\
 x_U \\
 1 \quad \quad \quad 1 \quad 0 \quad 0 \quad (62) \\
 2 \quad \quad \quad 0 \quad 0 \quad 1 \\
 3 \quad \quad \quad 0 \quad 1 \quad 0
 \end{array}$$

$$\begin{aligned}
 & c_{14}^1 c_{15}^1 c_{16}^1 \quad (63) \\
 C^1 = & c_{24}^1 c_{25}^1 c_{26}^1 \\
 & c_{34}^1 c_{35}^1 c_{36}^1
 \end{aligned}$$

$$\begin{aligned}
 & c_{41}^2 c_{42}^2 c_{43}^2 \quad (64) \\
 C^2 = & c_{51}^2 c_{52}^2 c_{53}^2 \\
 & c_{61}^2 c_{62}^2 c_{63}^2
 \end{aligned}$$

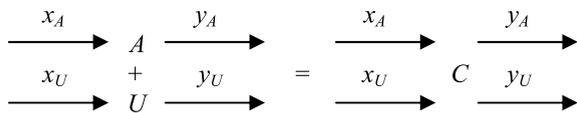


Fig. 9. Fuzzy network with nodes A and U vertically merged into node C.

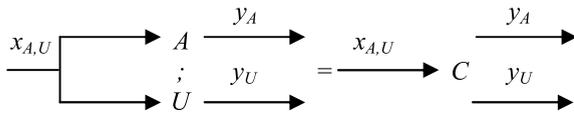


Fig. 10. Fuzzy network with nodes A and U output merged into node C.

$$C^3 = \begin{matrix} c_{77}^3 c_{78}^3 c_{79}^3 \\ c_{87}^3 c_{88}^3 c_{89}^3 \\ c_{97}^3 c_{98}^3 c_{99}^3 \end{matrix} \quad (65)$$

### 5.3. Output merging example

This example considers a FN with two nodes A and U with common input  $x_{A,U}$  whereby  $y_A$  is the output from A and  $y_U$  is the output from U. These nodes are output merged into node C that represents the equivalent fuzzy system for this FN, as shown by the block-scheme in Fig. 10 and the topological expression in Equation (66). The known nodes A and C are described by the Boolean matrices in Equations (67) and (68).

$$[A](x_{A,U}|y_A); [U](x_{A,U}|y_U) = [C](x_{A,U}|y_A, y_U) \quad (66)$$

$$A : \begin{matrix} y_A & 1 & 2 & 3 \\ x_{A,U} & & & \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 \end{matrix} \quad (67)$$

$$C : \begin{matrix} y_A, y_U & 11 & 12 & 13 & 21 & 22 & 23 & 31 & 32 & 33 \\ x_{A,U} & & & & & & & & & \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{matrix} \quad (68)$$

The Boolean matrix C in Equation (68) contains only 3 non-zero row blocks  $C^k$  whereby there is not more than one such block in any row of this matrix. Also, the elements with a location in A corresponding to the location of non-zero row blocks  $C^k$  in C are all equal to 1 and all other elements with a

location in A corresponding to the location of zero row blocks  $C^0$  in C are equal to 0.

Therefore, the rows of the Boolean matrix U are equal to  $C^k$ , as shown by the Boolean matrix in Equation (69). In this case, the location of the elements of  $C^k$ ,  $k = 1, 2, 3$  in C is given by Equations (70)–(72).

$$U : \begin{matrix} y_U & 1 & 2 & 3 \\ x_{A,U} & & & \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 \\ 3 & 0 & 1 & 0 \end{matrix} \quad (69)$$

$$C^1 = c_{14}^1 c_{15}^1 c_{16}^1 \quad (70)$$

$$C^2 = c_{21}^2 c_{22}^2 c_{23}^2 \quad (71)$$

$$C^3 = c_{37}^3 c_{38}^3 c_{39}^3 \quad (72)$$

## 6. Conclusion

The proposed approach for complex systems modelling by fuzzy networks with interconnected rule bases improves the transparency of the models used. This allows the structure of a fairly complex interconnected process to be reflected explicitly in the model. As a result, any complex process can be modelled by a fuzzy network in a more transparent way than by a fuzzy system due to the better visibility inside the process. This facilitates not only the analysis but also the design of the modelled process.

The proposed approach is based on formal models for fuzzy networks and basic operations in such networks. The formal models used are Boolean matrices, block-schemes and topological expressions. The basic operations are binary and they include horizontal, vertical and output merging of rule bases.

The basic operations are used for defining several node identification cases in fuzzy networks by means of Boolean matrix equations. The solution to some of these equations always exists. The exceptions are node identification cases that may not have a solution. In this context, when a solution exists, node identification in horizontal merging is likely to have multiple solutions whereas node identification in vertical and output merging usually has a unique solution.

The proposed approach is illustrated for fuzzy networks with a fairly small number of inputs, outputs

and connections. However, it can be easily extended to fuzzy networks with an arbitrarily large number of inputs, outputs and connections. In this case, as all binary merging operations are associative, they can be applied repetitively in a flexible way.

## Acknowledgments

The first author would like to thank the Faculty of Technology at the University of Portsmouth for the granted research sabbatical that made possible the writing of this paper.

## References

- [1] A. Fernandez-Caballero, Contribution of fuzziness and uncertainty to modern artificial intelligence, *Fuzzy Sets and Systems* **160**(2) (2009), 129–129.
- [2] A. Fiordaliso, A constrained Takagi-Sugeno fuzzy system that allows for better interpretation and analysis, *Fuzzy Sets and Systems* **118**(2) (2001), 307–318.
- [3] A. Gegov, *Complexity management in fuzzy systems: A rule base compression approach*, Springer, Berlin, 2007.
- [4] A. Gegov, *Fuzzy networks for complex systems: A modular rule base approach*, Springer, Berlin, 2010.
- [5] A. Piegat, *Fuzzy modelling and control*, Springer, Berlin, 2001.
- [6] B. Chen and X. Liu, Fuzzy approximate disturbance decoupling of MIMO nonlinear systems by backstepping and application to chemical processes, *IEEE Transactions on Fuzzy Systems* **13**(6) (2005), 832–847.
- [7] B. Chen, S. Tong and X. Liu, Fuzzy approximate disturbance decoupling of MIMO nonlinear systems by back stepping approach, *Fuzzy Sets and Systems* **158**(10) (2007), 1097–1125.
- [8] C. Igel and K.H. Temme, The chaining syllogism in fuzzy logic, *IEEE Transactions on Fuzzy Systems* **12**(6) (2004), 849–853.
- [9] C. Xu and Y.C. Shin, Interaction analysis of MIMO nonlinear systems based on fuzzy basis function network model, *Fuzzy Sets and Systems* **158**(18) (2007), 2013–2015.
- [10] D. Ji-Cheng and C. Fu-Lai, Cascaded fuzzy network model based on syllogistic fuzzy reasoning, *IEEE Transactions on Fuzzy Systems* **9**(2) (2001), 293–306.
- [11] F. Cheong and R. Lai, Designing a hierarchical fuzzy logic controller using the differential evolution approach, *Applied Soft Computing* **7**(2) (2007), 481–491.
- [12] F.J. Moreno-Velo, I. Baturone, A. Barriga and S. Sanchez-Solano, Automatic tuning of complex fuzzy systems with X-fuzzy, *Fuzzy Sets and Systems* **158**(18) (2007), 2026–2038.
- [13] F. Wan, H. Shang, L.X. Wang and Y.X. Sun, How to determine the minimum number of fuzzy rules to achieve given accuracy: A computational geometric approach to SISO case, *Fuzzy Sets and Systems* **150**(2) (2005), 199–209.
- [14] G. Aluja, *Fuzzy sets in the management of uncertainty*, Springer, Berlin, 2004.
- [15] G. Bojadziev and M. Bojadziev, *Fuzzy logic for business, finance and management*, World Scientific, Singapore, 2007.
- [16] H. Zhang, C. Dang and C. Li, Decentralized H-infinity filter design for discrete-time interconnected fuzzy systems, *IEEE Transactions on Fuzzy Systems* **17**(6) (2009), 1428–1440.
- [17] J.J. Buckley, *Simulating fuzzy systems*, Springer, Berlin, 2005.
- [18] J.M. Garibaldi and T. Ozen, Uncertain fuzzy reasoning: A case study in modelling expert decision making, *IEEE Transactions on Fuzzy Systems* **15**(1) (2007), 16–30.
- [19] J. Mar and H.T. Lin, A car-following collision prevention control device based on the cascaded fuzzy inference system, *Fuzzy Sets and Systems* **150**(3) (2005), 457–473.
- [20] K.J. Kim and S.B. Cho, Personalized mining of web documents link structures and fuzzy concept networks, *Applied Soft Computing* **7**(1) (2007), 398–410.
- [21] L.O. Hall, Rule chaining in fuzzy expert systems, *IEEE Transactions on Fuzzy Systems* **9**(6) (2001), 822–828.
- [22] M. Bucolo, L. Fortuna and M. La Rosa, Complex dynamics through fuzzy chains, *IEEE Transactions on Fuzzy Systems* **12**(3) (2004), 289–295.
- [23] M.G. Joo and J.S. Lee, A class of hierarchical fuzzy systems with constraints on the fuzzy rules, *IEEE Transactions on Fuzzy Systems* **13**(2) (2005), 194–203.
- [24] M.G. Joo and T. Sudkamp, A method of converting a fuzzy system to a two-layered hierarchical fuzzy system and its runtime efficiency, *IEEE Transactions on Fuzzy Systems* **17**(1) (2009), 93–103.
- [25] M. Kumar, R. Stoll and N. Stoll, A robust design criterion for interpretable fuzzy models with uncertain data, *IEEE Transactions on Fuzzy Systems* **14**(2) (2006), 314–328.
- [26] M.L. Lee, H.Y. Chung and F.M. Yu, Modelling of hierarchical fuzzy systems, *Fuzzy Sets and Systems* **135**(2) (2003), 343–361.
- [27] M. Landago, M.J. Rio and A. Perez, A note on smooth approximation capabilities of fuzzy systems, *IEEE Transactions on Fuzzy Systems* **9**(2) (2001), 229–237.
- [28] N.R. Pal, V.K. Eluri and G.K. Mandal, Fuzzy logic approaches to structure preserving dimensionality reduction, *IEEE Transactions on Fuzzy Systems* **10**(3) (2002), 277–286.
- [29] N. Xiong and L. Litz, Reduction of fuzzy control rules by means of premise learning –method and case study, *Fuzzy Sets and Systems* **132**(2) (2002), 217–231.
- [30] S. Aja-Fernandez and C. Alberola-Lopez, Matrix modelling of hierarchical fuzzy systems, *IEEE Transactions on Fuzzy Systems* **16**(3) (2008), 585–599.
- [31] S. Guillaume, Designing fuzzy inference systems from data: An interpretability-oriented review, *IEEE Transactions on Fuzzy Systems* **9**(3) (2001), 426–443.
- [32] S.M. Zhou and J.Q. Gan, Low-level interpretability and high-level interpretability: A unified view of data-driven interpretable fuzzy system modelling, *Fuzzy Sets and Systems* **159**(23) (2008), 3091–3131.
- [33] T.P. Zhang, Stable adaptive fuzzy sliding mode control of interconnected systems, *Fuzzy Sets and Systems* **122**(1) (2001), 5–19.
- [34] T. Ross, *Fuzzy logic with engineering applications*, Wiley Chichester, 2004.
- [35] W.J. Wang and L. Luoh, Stability and stabilization of fuzzy large-scale systems, *IEEE Transactions on Fuzzy Systems* **12**(3) (2004), 309–315.
- [36] W.J. Wang and W.W. Lin, Decentralized PDC for large-scale fuzzy systems, *IEEE Transactions on Fuzzy Systems* **13**(6) (2005), 779–786.
- [37] Y. Chen, B. Yang, A. Abraham and L. Peng, Automatic design of hierarchical Takagi-Sugeno type fuzzy systems using

- evolutionary algorithms, *IEEE Transactions on Fuzzy Systems* **15**(3) (2007), 385–397.
- [38] Y.M. Li, Z.K. Shi and Z.H. Li, Approximation theory of fuzzy systems based upon genuine many-valued implications –MIMO cases, *Fuzzy Sets and Systems* **130**(2) (2002), 159–174.
- [39] Z. Lendek, R. Babuska and B. De Schutter, Stability of cascaded fuzzy systems and observers, *IEEE Transactions on Fuzzy Systems* **17**(3) (2009), 641–653.
- [40] Z.M. Yeh and K.H. Li, A systematic approach for designing multistage fuzzy control systems, *Fuzzy Sets and Systems* **143**(2) (2004), 251–273.

AUTHOR COPY