An algebraic model of the production type distributed intelligent system

S D Makhortov, S Yu Bolotova
Department of Applied Mathematics, Informatics and Mechanics, Voronezh State University, 1, University Square, Voronezh, 394018, Russia
E-mail: makhortov@amm.vsu.ru, bolotova.svetlana@gmail.com

Abstract. The LP structures theory contains a lattice-based algebraic approach for modeling and researching a wide range of production and similar systems in computer science. The article studies an algebraic model (LP structure) which has a number of significant capabilities that cover distributed knowledge systems. The authors formulate some results that create a theoretical basis for optimizing distributed knowledge management. The paper considers a production logical equations class, which opens up new possibilities for accelerating distributed backward logical inference.

1. Introduction
Modern information technologies are characterized by a significant increase in the scale and critical importance of the tasks being solved. Accordingly, the significance of rigorous rationale of the correctness and reliability of information processing increases. Similar critical areas include distributed knowledge-based systems. For example, here we can name such areas as searching and intellectual analysis of thematic information in large data stores that support each node’s local knowledge base; medical diagnostics, when a "virtual consilium" of knowledge bases developed by different specialists participates in the process of diagnosing; educational systems in the global network; intelligent information systems of corporations and others [1].

An effective approach to the construction and investigation of distributed artificial intelligence is represented by the concept of multi-agent systems [2]. There are other methods that have individual advantages in solving specific problems associated with distributed intelligent systems [1, 3].

In recent years, the authors have obtained a number of results in the field of knowledge management. They are associated with the logical systems of the production type [3, 4] which are widespread in computer science. There has been developed an algebraic theory of LP structures [5], which provides the rationale and effective solution of the problems of equivalent transformations, verification, minimization of knowledge bases, alongside with acceleration of logical inference. The theory also proved to be applicable in the studies of other systems which had not previously been considered as production ones [6].

Article [7] made another step in the development of the LP structures theory by introducing an algebraic model embracing distributed production-logical systems and complemented this model with the corresponding production-logical equations apparatus.
The present paper combines and generalizes the results of article [7], thus creating a theoretical basis for optimizing distributed knowledge systems and accelerating distributed logical inference.

The paper consists of the following sections. Part 2 gives necessary notions and notation. Part 3 introduces a distributed LP structure and establishes its basic properties. Part 4 determines the apparatus of production logical equations in a distributed LP structure, discusses their properties and methods of solution. Part 5 presents ideas on practical application of distributed LP structures. The conclusion summarizes the results of the research and indicates some perspectives.

Due to the article volume limitation, mathematical statements are given without proof.

2. Fundamentals of the LP structures theory

The notations and definitions needed for the further presentation are described, for example, in [5, 6]. Recall some of the main ones.

A binary relation \( R \) on an arbitrary set \( F \) is said to be reflexive if, for any \( a \in F \), \((a, a) \in R\) is true; transitive if, for any \( a, b, c \in F \) from \((a, b), (b, c) \in R\) it follows that \((a, c) \in R\). It is known that there exists a closure \( R^* \) of an arbitrary relation \( R \) with respect to reflexivity and transitivity properties (a reflexive-transitive closure). The inverse problem is the finding of a transitive reduction: by the given \( R \) and transitivity properties (a reflexive-transitive closure).

A lattice [9] is a set with a partial order \( \leq \) ("no more than", "is contained"), in which for any pair of elements operations are defined \( \land \) ("meet") — an exact lower bound of this pair and \( \lor \) ("join") — its exact upper bound. Lattice \( F \) is called bounded, if it contains common lower and upper bounds, i.e. such two elements \( O, I \), that for any \( a \in F \). The atom of a bounded (from below) lattice \( F \) is a minimal element of its subset \( F \setminus \{O\} \). A lattice is called atomistic if each of its elements can be decomposed as a join of atoms.

An LP (lattice production) structure is an algebraic system representing a lattice \( F \) with a defined additional binary relation \( R \) on it that has some ("production-logical") properties. Relation \( R \) models a set of intelligent system rules (productions). The lattice elements correspond to the premises and conclusions of these rules, containing elementary facts of the knowledge base and expressions over them.

Relation \( R \) is called production-logical if it has reflexivity, that is, it contains all pairs of the form \((a, a)\), transitivity, and other properties that are determined by a particular model. One of these properties is distributivity. Informally, distributivity of a relation means the possibility of logical inference in parts and the union of its results on the basis of lattice operations \( \land \) and \( \lor \).

This paper considers lattices with semantics of sets \( \lambda(F) \) (a set of finite subsets of the universe \( F \)) as LP structures basis. Therefore, instead of the symbols \( \leq, \geq, \land \) and \( \lor \), the signs of set-theoretic operations \( \subseteq, \supseteq, \cap \) and \( \cup \), are used, and the lattice elements are usually denoted by capital letters. The exception is atoms, denoted by small letters. Atoms here are all subsets consisting of exactly one element of the universe. In such a notation, \( U \)-distributivity is treated in the following sense: from \((A, B_1), (A, B_2) \in R\) follows \((A, B_1 \cup B_2) \in R\).

For LP structures the following main questions are relevant [5]: on closure, equivalent transformations, canonical form and logical reduction. Solving these issues provides opportunities for substantiation, automated research and optimization of production sets in various subject areas. The logical closure of an arbitrary binary relation \( R \) is the least production logical relation containing \( R \). Two relations \( R_1, R_2 \) are called (logically) equivalent if their logical closures coincide. For such relations the notation \( R_1 \sim R_2 \), is used. An equivalent transformation of a given relation is a partial replacement of some subset of its pairs which leads to an equivalent relation.
In [5] the author proves the theorems on a logical closure of an arbitrary binary relation existence and on the principle of locality of equivalent transformations of logical relations.

**Theorem 1.** For an arbitrary binary relation $R$, there exists a logical closure $\overrightarrow{R}$ on the lattice.

**Theorem 2.** Let $R_1, R_2, R_3, R_4$ be relations on common lattice. If $R_1 \sim R_2$ and $R_3 \sim R_4$ then $R_1 \cup R_3 \sim R_2 \cup R_4$.

Relation $R$ on an atomistic lattice $\mathbb{F}$ is called canonical, if it is given by a set of pairs of the form $(A, a)$, where $A \in \mathbb{F}$, $a$ is an atom in $\mathbb{F}$. (that is, an indecomposable element).

**Assertion 1.** For any relation on an atomistic lattice there exists an equivalent canonical relation.

We have also considered the question of minimizing binary relations with preservation of their properties. A logical reduction of the relation $R$ on a lattice is any minimal relation which is equivalent to $R$. The theorem on the existence of a logical reduction and the method of its construction is valid.

**Theorem 3.** For an arbitrary binary relation $R$ on the lattice, there exists a logical reduction.

The proposed approach to the research of production and similar systems, as well as the improvement of the inference, is based on the representation of sets of facts and rules by the LP structure. Each elementary fact is reflected by a lattice atom, the premise and conclusion of the rule — by the corresponding elements of the lattice, and the rules themselves are represented by pairs of the binary relation $R$.

### 3. Distributed LP structure and its properties

As noted in [10], different sources provide various definitions of distributed systems, many of which are not complete and often not consistent with other definitions. For the purposes of this article it is sufficient to consider a free and intuitive interpretation of this term. Let there be a distributed computer network, where each node can store a set of facts (database), a subset of production rules (knowledge base), and also there can function a logical inference machine with local working memory. Thus, there is a distributed production system, consisting of a set of local production systems (subsystems), coordinated with each other.

Each fact and each rule of a distributed production system is associated with a subset of computer network nodes where this fact and this rule are stored. It is assumed that the subsystems are more or less independent, but require some (relatively expensive) interaction, the intensity of which in the general case is substantially lower than the intensity of local calculations.

Denote by $N$ — a finite set of nodes of the computer network, $\mathbb{N}$ — a power set generated by it, that is, a set of all its subsets. Elementary facts and rules of the production system will be marked with lattice $\mathbb{N}$ elements as attributes.

So, let two lattices be given: $\mathbb{F}$ (basic for the LP structure) and $\mathbb{N}$ (lattice of nodes). On lattice $\mathbb{F}$, a mapping $\text{Nodes}()$ is defined, that assigns to each atom $a \in \mathbb{F}$ a single non-empty element $X \in \mathbb{N}$ (that is, $\text{Nodes}(a) = X, X \neq \emptyset$). On the lattice $\mathbb{F}$, a binary relation $R$ is given. Besides, each pair $(A, B) \in R$ is also associated with a nonempty element $Y \in \mathbb{N}$ ($\text{Nodes}(A, B) = Y, Y \neq \emptyset$).

In the object interpretation (that is, in the modeled production system), the $\text{Nodes}()$ function determines for each elementary fact or rule a set of nodes of the distributed computing system where this fact or rule is stored. The range of the function $\text{Nodes}()$ does not contain the empty element $\emptyset$, which means that each fact of the database and every knowledge base rule is stored on at least one node of the distributed system.

**Definition 1.** The algebraic system described above is called a distributed LP structure.
Note that in the presence of the $\text{Nodes}(\cdot)$ map, the lattice $F$ and the relation $R$ can also be considered as distributed. Next, for a distributed LP structure, the standard questions (i.e. closure, equivalent transformations, canonical form, logical reduction) are considered.

**Definition 2.** The distributed relation $R$ on a distributed lattice $F$ is called production-logical if it is reflexive, transitive, and distributive, and for each $(A, B) \in R$, $\text{Nodes}(A, B) = N$. The smallest production-logical relation containing $R$ is called the logical closure of the relation $R$.

The existence of a logical closure for an arbitrary distributed relation follows from the corresponding base result (Theorem 1, part 2).

**Theorem 4.** For an arbitrary distributed binary relation $R$ on a distributed lattice $F$ there exists a logical closure $\overline{R}$.

Similarly, the concept of equivalent transformations and its proof is carried over to the distributed LP structure.

**Definition 3.** Two distributed relations $R_1, R_2$ on a distributed lattice $F$ are called equivalent ($R_1 \sim R_2$) if their logical closures (in the sense of Definition 3.2) coincide. An equivalent transformation of a given relation is a replacement of a subset of its pairs, which leads to an equivalent relation.

From Theorem 2, part 2 and Theorem 4 of this part we immediately get the following result.

**Theorem 5.** Let $R_1, R_2, R_3, R_4$ be distributed relations on a common distributed lattice. If $R_1 \sim R_2$ and $R_3 \sim R_4$ then $R_1 \cup R_3 \sim R_2 \cup R_4$.

We now proceed to the discussion of questions of the distributed LP structures optimization. For this purpose along with the $\text{Nodes}(\cdot)$ function, we introduce two more mappings from $F$ to $N$. Let an arbitrary element $A \in F$ be taken. Since the lattice $F$ is atomistic, there is a representation $A = \bigcup_i a_i$ in the form of a union of atoms. We denote by

$$\text{NodesMeet}(A) = \bigcap_i \text{Nodes}(a_i); \quad \text{NodesJoin}(A) = \bigcup_i \text{Nodes}(a_i).$$

The $\text{NodesMeet}(\cdot)$ mapping defines nodes, where each one contains all the atoms of the $A$ element. The $\text{NodesJoin}(\cdot)$ mapping generates all nodes that contain at least one of these atoms.

A question arises: what value of a nonempty attribute $\text{Nodes}(A, B)$ should be called preferable. On the one hand, an important goal of modeling distributed logic systems is the rationale for traffic reduction between nodes. On the other hand, unnecessary duplication of information on various nodes is impractical.

Let a local inference machine operating with the facts of some node run on its node. It will be costly for it to look for the inference rules for derivation of its own facts on other nodes. Therefore, we consider it expedient to store on the node all the rules where the facts of that node are used. The following concepts characterize the "good" distribution of rules over the computer network nodes.

**Definition 4.** A distributed binary relation $R$ on a distributed lattice $F$ is called relevant-correct if for any $(A, B) \in R$, $\text{Nodes}(A, B) \supseteq \text{NodesJoin}(A) \cup \text{NodesJoin}(B)$; relevant-normalized if $\text{Nodes}(A, B) = \text{NodesJoin}(A) \cup \text{NodesJoin}(B)$.

The latter relation more closely matches the considered inference strategy. Whereas the first one allows storing such rules on a node that are not connected with any fact of the same node.

**Definition 5.** A distributed relation $R$ on a lattice $F$ is called canonical if it is relevant-normalized and given by a set of pairs of the form $(A, a)$, where $A \in F$, $a$ is an atom in $F$. 

4
On the basis of Assertion 1 and the results of this part, the following fact is deduced.

**Assertion 2.** For any distributed relation on a distributed lattice, there is an equivalent canonical relation.

Next, consider the question of equivalent minimization for the distributed binary relations.

**Definition 6.** A logical reduction of a distributed relation $R$ on a distributed lattice is any minimal relation equivalent to $R$ and at that relevant-normalized.

Here we emphasize that in Definition 6 we are talking exactly about a minimum relation, which may be not the only one. By the minimum equivalent relation we mean such that the exclusion from it of any single pair will lead to an inequivalent relation.

**Theorem 6.** For an arbitrary distributed binary relation $R$ on a distributed lattice, there exists a logical reduction.

### 4. Production-logical equations in a distributed LP structure

In this section, we introduce a relational equations class associated with the distributed LP structures. The process of finding the solution corresponds to the backward logical inference on the distributed lattice. Let a distributed relation $R$ be given on a distributed lattice $F$ and $A \xrightarrow{R} B$. Then $B$ is called an image of $A$ and $A$ is a preimage of $B$ under the relation $A \xrightarrow{R} B$.

For the given $B \in F$ the minimal preimage under the relation $\xrightarrow{R}$ is such element $A \in F$ that $A \xrightarrow{R} B$ and $A$ is minimal, i.e. it does not contain any other $A_1 \in F$, for which $A_1 \xrightarrow{R} B$.

**Definition 7.** Atom $x \in F$ is called initial under the distributed relation $R$, if in $R$ there is no such pair $(A; B)$ where $x$ is contained in $B$ and not contained in $A$. Element $X$ is called initial, if all its atoms are initial. A subset $F_0(R)$ (sometimes denoted as $F_0$) consisting of all the initial elements of $F$ is called an initial set of lattice $F$.

Let us consider the equation

$$R^L(X) = B,$$

where $B \in F$ is a given element and $X \in F$ is an unknown one.

**Definition 8.** Any minimal preimage of element $B$ in $F_0$ is called a partial (distributed) solution to equation (1). Any preimage of element $B$ contained in $F_0$ is called an approximate partial solution to equation (1). A collection of all partial solutions $\{X_s\}$, $s \in S$ is called a general solution to equation (1).

The equations of form (1) can be called production logical equations on a distributed lattice. According to [5] it is possible to describe changes in the general solution of equations of the form (1) when their right-hand sides unite. With this purpose let us consider a production logical equation of the following form

$$R^L(X) = B_1 \cup B_2$$

**Theorem 7.** Let $\{X_s\}$, $s \in S_1$ be a general solution to the equation of the form (1) with the right-hand side $B_1$ and $\{Y_p\}$, $p \in S_2$ be a general solution to the equation of the same form with the right-hand side $B_2$. Then a general solution to the equation (2) is a set of all elements of the form $X_s \cup Y_p$, from which the elements containing other elements of the same set are eliminated.

Let us consider ways of solving distributed equations of the form (1). We will assume that $R$ is a finite canonical distributed relation on a distributed lattice $F$ that does not contain the relation $\subseteq$ pairs, and the right-hand side $B$ of equation (1) is a finite join of atoms.
Introduce the splitting of the initial distributed relation $R$ on virtual layers $\{R_t|t \in T\}$. The purpose of this splitting is to simplify the analyzing of the logical closure $R \rightarrow L$ properties. In addition, in separate layers the construction and research of algorithms associated with the production-logical equations solution is simplified. Split $R$ into disjoint subsets, each of which is formed by pairs $(A, x_p)$ with the same atom $x_p$ as the right-hand side. Such splitting is possible, since $R$ is canonical. Denote these subsets $R^p$ by their atom $x_p, p \in P$.

**Definition 9.** A layer in a distributed relation $R$ is a subset $R_t$ formed by ordered pairs taken one pair from each nonempty $R^p, p \in P$.

For the splitting of a distributed relation a number of useful properties, similar to those described in [5] is preserved.

**Remark 1.** Any subset of pairs in $R$ with unique right-hand sides belongs to some single layer. A total number of layers $M$ is defined by the equality $M = \prod_{p \in P} M_p$, where $M_p$ is the cardinality of a subset of pairs $R$ with the right-hand side $x_p$.

**Corollary 1.** The logical closure of a canonical distributed relation $R$ on a distributed lattice $F$ is equal to the union of logical closures of its layers, that is $R^L = \bigcup_{t \in T} R^L_t$.

We say that a solution $X$ of equation (1) (exact or approximate) is generated in $R$ by some layer $R_t$ if $X \rightarrow R B$.

**Remark 2.** Similarly to [5], it is possible to show that any partial solution of equation (1) is generated by some layer $R_t$.

**Remark 3.** It follows from Corollary 1 that to find a solution of equation (1) in a certain layer $R_t$ it is sufficient to solve a similar equation with a relation $R_t$ instead of (1).

**Corollary 2.** One layer cannot contain two different partial solutions of equation (1).

From corollaries 1 and 2 formulated above there immediately follows theorem 8.

**Theorem 8.** To find the general solution of the distributed equation (1) it is sufficient to find a partial solution $X_t$ in each layer $R_t$, where it exists. Further, from the resulting set of solutions, it is necessary to exclude the elements containing other elements of the same set.

Theorem 8 and Remark 3 make it possible to reduce the problem of solving equation (1) to finding a partial solution of the distributed equation

$$R^L_t(X) = B,$$

where $B$ is not the initial element of the distributed lattice $F$, $R_t$ — is an arbitrary layer in $R$.

On the basis of Theorem 7, to solve the distributed equation (3) it is sufficient to solve an equation with each atom of the element $B$ as the right-hand side. Therefore, consider the following equation:

$$R^L_t(X) = b,$$

where $b$ is not the initial atom of lattice $F$, $R_t$ — is an arbitrary layer in $R$.

Let us describe the method of solving this problem, generalizing the analogous method in [5]. It is equivalent to enumerating the distributed oriented graph input vertices, from which a given vertex is reachable. Each atom of the distributed lattice $F$ involved in the relation $R$ is associated with the vertex of the graph. Then for each pair $(A, a)$ of a fixed layer $R_t$, there are constructed arcs leading from all the vertices, corresponding to atoms $A$, to the vertex corresponding to the given $a$. In brief, it is possible to identify the lattice atoms and the corresponding vertices in the graph.
In the graph \( G_{R_t} \), choose a vertex \( b \). Consider a subgraph \( G_{R_t;b} \subseteq G_{R_t} \) containing all vertices from which the vertex \( b \) (including \( b \) itself) and all the arcs connecting such vertices are accessible. Formulate a theorem that completes the substantiation for the step-by-step process for solving the initial production logical equation (1).

**Theorem 9.** Equation (4) has no more than one solution. If the graph \( G_{R_t;b} \) does not contain cycles, then the unique solution of the equation consists of all the atoms corresponding to the input vertices of the graph. If \( G_{R_t;b} \) contains at least one cycle, then the equation does not have solutions in the layer.

5. **The possibilities of using distributed LP structures**

The distributed LP structure represents a formal model of the distributed intelligent system. Thus, the theorems in part 3 substantiate the automated equivalent transformations of such systems for optimization and minimization purposes. The practical application of the apparatus of production-logical equations (part 4) is the distributed backward inference optimizing.

The relevant inference strategy [11] is aimed at minimizing the number of slow queries (to a database or a user). The backward inference based on equation solving starts with the construction of all minimal initial preimages in LP structure for the atoms corresponding to the values of the object of expertise. Further, in the constructed set it is enough to find the preimage, which contains only true facts, after which it is immediately possible to make a conclusion about the corresponding value of the object of expertise.

There exists an effective way — priority viewing of preimages containing the values of the most relevant objects. These are objects, the values of which are present in a maximal number of preimages. Then the only negative answer to a given question at once excludes a large number of preimages from consideration. The second indicator of the relevance of the tested object is the presence of its values in the preimages of minimal cardinality. Thus, preference is given to those preimages, the validation of which will require a smaller number of questions to the user (or queries to the database). As experiments show [11], when applying the relevant LP-inference it is possible to achieve a decrease of the number of slow queries performed by an average of 20%.

The distributed nature of the intelligent system creates additional difficulties in implementing an effective logical inference. One of the purposes of modeling such a system is the reduction of traffic between the nodes of the computer network during the inference process. Thus, when developing a strategy of the distributed relevant LP-inference, along with the mentioned above two characteristics, it is necessary to take the additional relevance parameters into account. They are associated with the attributes of storing facts and rules, which are described by the mappings \( \text{Nodes}() \), \( \text{NodesMeet}() \), \( \text{NodesJoin}() \), introduced in part 3 of this work.

6. **Conclusion**

In this paper we consider an LP structure that extends the field of the application of the given theory up to distributed intelligent systems of production type.

For a distributed LP structure, a standard set of questions — closure, equivalent transformations, canonical form, logical reduction, equations, and their solutions — is studied. Thus, a theoretical basis for formalizing knowledge management and optimizing distributed logical inference is created. The next step in the research may be the development of strategies of application of new LP inference relevance parameters connected with the distributed nature of intelligent systems.

The presented results allow to use the advantages of the LP structures theory in construction and research of distributed intelligent systems of production type in various subject areas. Since a lot of models in computer science have a production character, further studies in the direction
considered can also be associated with LP structures of more complex types and corresponding subject areas, and with transferring the concepts of this work to these models.

Acknowledgments
This work was funded by RFBR according to the research project 19-07-00037

References
[1] Soshnikov D V 2002 Methods and means of constructing distributed intelligent systems on the basis of production-frame of knowledge representation Ph.D. thesis Moscow diss. kand.fiz.-mat.nauk; 05.13.11 (In Russian)
[2] Makarov V L, Bakhtizin A R, Sushko E D, Vasenin V A, Borisov V A and Roganov V A 2016 Herald of the RAS, Pleiades Publishing 86 248–257
[3] Ginkul G P 2013 Programmnyye sistemy i instrumenty 14 172–175 (In Russian)
[4] Ishida T 1995 Proc. of the First Intern. Conf. on Multiagent Systems 416–422
[5] Makhortov S D 2012 Automation and Remote Control 73 1937–1943 doi: 10.1134/S0005117912110161
[6] Makhortov S D and Shurlin M D 2013 Automation and Remote Control 74 1211–1217 doi:10.1134/S0005117913070126
[7] Makhortov S D 2016 Programmnaya inzheneriya 324–329 (In Russian)
[8] Aho A V, Garey M R and Ulman J D 1972 SIAM J. Computing 1 : 2 131–137
[9] Birkhoff G 1967 Lattice Theory (Providence: Am. Math. Soc)
[10] Tanenbaum A S and Steen M V 2006 Distributed Systems. Principles and Paradigms (Prentice-Hall, Inc)
[11] Bolotova S Y and Makhortov S D 2011 Iskusstvennyy intellekt i prinyatiye resheniy 40–50 (In Russian)