Armenian Journal of Mathematics
Volume 1, Number 2, 2008, 10–24

Automorphic Equivalence of Multi-models Recognition

Knyazhansky Marina
Plotkin Tatjana [1]
Bar-Ilan University, Ramat Gan 52900, Israel
[1]plotkin.evgeny@gmail.com

Abstract. In this paper we provide an implementable formal algorithm for knowledge bases equivalence verification based on the formal definition of knowledge base given by B. I. Plotkin in his works and also study some important properties of automorphic equivalence of models. In addition we show that notion of automorphic equivalence is much wider than the notion of isomorphism.

Key words: Knowledge base, implementable formal algorithm, automorphic equivalence of models

Mathematics Subject Classification 2000: 68M20, 68T99, 08C05

1. Introduction and Motivation

This is a mathematical paper whose subject is originated from databases (DB) and knowledge bases (KB) theory particularly, and from Theoretical Computer Science in general. We proved the following theorem: the problem of informational equivalence of two knowledge bases with finite corresponding multimodels is decidable.

On the other hand, a part of this theorem is effective and provides an implementable formal algorithm for KB equivalence recognition. Thus, we think that it is of practical importance and can find applications in KB theory. This feedback is one of our principal aims.

*** Presented at the First International Algebra and Geometry Conference in Armenia, 16–20 May, 2007, Yerevan**

More precisely, the goal of this paper is to construct an algorithm for knowledge bases equivalence verification based on the formal definition of knowledge base given in [**16, 18, 20, 21**] and to study some important properties of automorphic equivalence of models. We will describe the concept of equivalence and formulate the criterion for the equivalence of knowledge bases defined over finite models. Further we will define multi-models and automorphic equivalence of models and multi-models that are generalization of automorphic equivalence of algebras.

The recent advances in knowledge base research and the growing importance of effective knowledge management emphasized the significance of the question of knowledge bases equivalence verification. This problem has not been stated earlier, at least in a way that allows speaking about algorithms for verification of informational equivalence, because it requires a formal mathematical definition of the corresponding objects, while the majority of the existing approaches rely on informal or descriptive definitions which make the exact solution impossible.

1.1. Knowledge Bases. Descriptive Definitions. There is no single agreed definition of knowledge presently, and there remain numerous competing theories. In any case knowledge is some essence which requires representation of knowledge. Various artificial languages and notations have been proposed for representing knowledge. They are typically based on logic and mathematics, and have easily parsed grammars to ease machine processing [**3, 6, 11, 12, 13, 23, 26, 27**].

In general, a knowledge base is not a static collection of information (like a database), but a dynamic resource that may itself have the capacity to learn, as part of an artificial intelligence component. These kinds of knowledge bases can suggest solutions to problems sometimes based on feedback provided by the user, and are capable of learning from experience (like an expert system). Knowledge representation, automated reasoning, argumentation and other areas of artificial intelligence are tightly connected with knowledge bases.

1.2. Equivalence Problem. We study an equivalence of knowledge bases in respect to their informational abilities. In other words, we would like to discuss informational equivalence of knowledge bases. It means, we expect to get the same information but may be in different formats.

The principal task here is to find out whether the problem of informational equivalence verification is algorithmically solvable. If we concentrate on finite objects then the reasonable answer is yes, we can build the step-by-step procedure used to solve the problem. But when we consider infinite objects it may be problematic. Evidently, knowledge bases are the example of this case. On the other side, if we could find some finite invariant (or system of invariants) of a knowledge base, such that equivalence of those invariants would involve equivalence of corresponding knowledge bases, then the problem would turn to algorithmically solvable.

1.3. Problem in Question and the Main Results. Knowledge base systems combine features of database management systems with artificial intelligence techniques. The existing knowledge bases are defined using informal description of internal relationships and as a consequence they do not allow to identify equivalent knowledge represented in different ways by different knowledge base implementations. This problem can be solved by providing a formal mathematical model of a knowledge base. An algebraic model was presented in [**17**]. In the papers [**15, 16, 18, 19, 20, 21**] it was proposed a solution for the knowledge bases equivalence problem using an algebraic geometry approach [**7, 16**], category theory [**2, 10, 14**] and group-theoretic methods [**1, 4, 5, 8, 10, 22**].

In particular the following theorem had been obtained:
Two knowledge bases are informationally equivalent if and only if the corresponding subjects of knowledge are automorphically equivalent.

This result introduces the notion of automorphic equivalence as a key tool of the theory. Study of this notion is one of the main objectives of this paper.

We hope that the ability to verify informational equivalence of two different knowledge bases can be used to increase efficiency of knowledge retrieval and detection of hidden knowledge. If retrieving information from one knowledge base may be problematic, the same information can be possibly easily accessible in another informationally equivalent knowledge base. Another application of knowledge bases equivalence verification is the disambiguation of information that arrived from different sources or was encoded in different formats. In this case information that is considered equivalent can be skipped.

2. Algebraic Background

In this section we will discuss notions of model, multi-model and automorphic equivalence of models and multi-models.

2.1. Models.

DEFINITION 2.1. *We define a model as a triple* (D, Φ, f)*, where* D *is a data domain, that is, an algebra in a variety of algebras* Θ *(for example, vector space over a field),* Φ *is a set of symbols of relations,* f *is one of possible interpretations of these symbols as real relations in* D*, i. e., if* $\phi \in \Phi$ *is an n-ary relation in* Φ*, then* $f(\phi)$ *is a subset of the Cartesian product* D^n*. Moreover,* D *may be a multi-sorted set, i. e. ,* $D = \{D_i, i \in \Gamma\}$*, where* Γ *is a set of sorts* [**18, 25**].

2.2. Multi-Models.

DEFINITION 2.2. *A multi- model is a triple* (D, Φ, F)*, where* D *is a data domain (an algebra),* Φ *is a set of symbols of relations,* F *is a set of different interpretations of* Φ *on* D*.*

A model (D, Φ, f) is a particular case of a multi-model (D, Φ, F). The definition of multi-model takes into account that the instance (interpretation) f can change, for example over time or under some other circumstances. All these f constitute the set F. In general multi-models may be infinite but we consider only the finite ones [**18**].

2.3. Automorphic Equivalence of Models and Multi-Models. For the given model (D, Φ, f) we have a group $Aut(f)$ consisting of all bijections $s : D \to D$ compatible with the interpretation of symbols of relations. This means that for every n-ary relation $\phi \in \Phi$ and every element $(a_1, a_2, ..., a_n) \in f(\phi)$ the element $(sa_1, sa_2, ..., sa_n)$ belongs to $f(\phi)$ as well. In the case of a multi-sorted model $s = (s_i, i \in \Gamma)$. The set of all such s form the group of automorphisms of the model denoted by $Aut(f)$ (it should be noted that there are some cases when this group is trivial).

Recall that two models (A, Φ_1, f_1) and (B, Φ_2, f_2) are called isomorphic if the sets Φ_1 and Φ_2 coincide and there is a bijection $\sigma : A \to B$, which is an isomorphism of algebras, and for any n-ary relation $\phi \in \Phi$ we have $(a_1, a_2, ..., a_n) \in f_1(\phi)$ if and only if $(\sigma a_1, \sigma a_2, ..., \sigma a_n) \in f_2(\phi)$.

DEFINITION 2.3. *Let us consider two models* (A, Φ_1, f_1) *and* (B, Φ_2, f_2). *Assume that A and B are algebras with the same operations, defined by the variety of algebras* Θ. *They are called automorphically equivalent, if there is an isomorphism* $\mu : A \to B$ *such that groups of automorphisms are conjugated by this isomorphism, i.e.,* $Aut(f_2) = \mu Aut(f_1)\mu^{-1}$.

The difference between automorphic equivalence of models and automorphic equivalence of algebras is the transformation $\mu : A \to B$. For models we use the notion of isomorphism of algebras μ while for algebras we referred to it as a bijection of sets δ.

DEFINITION 2.4. *Two multi-models* (A, Φ_1, F_1) *and* (B, Φ_2, F_2) *are called automorphically equivalent, if there is a bijection* $\alpha : F_1 \to F_2$, *such that the models* (A, Φ_1, f) *and* (B, Φ_2, f^α) *are automorphically equivalent for every* $f \in F_1$.

This means that it is possible to correlate the instances of these multi-models in such a way that the corresponding models turn to be automorphically equivalent.

3. Multi-Models Automorphically Equivalent to a given one

Let us consider two ways to construct a multi-model (A, Φ, F_1), which is automorphically equivalent to the given multi-model (A, Φ, F).

We will use the multi-model to investigate automorphic equivalence properties.

We will investigate two possible approaches:

(1) In order to get $f' \in F_1$ some transformation σ will be applied to $f \in F$.
(2) We will define $f' \in F_1$ as a complement to $f \in F$.

Let us look at these two cases in more detail.

3.1. Construction of Automorphically Equivalent Multi-Models Using Transformation σ. Let σ be an automorphism of algebra A. For every interpretation $f \in F$ we will construct another interpretation f^σ by the following rule: for n-ary relation $\varphi \in \Phi$ and row $(a_1, a_2, ..., a_n) \in A^n$, we set: $(a_1, a_2, ..., a_n) \in f^\sigma(\varphi)$ if and only if $(a_1^{\sigma^{-1}}, a_2^{\sigma^{-1}}, ..., a_n^{\sigma^{-1}}) \in f(\varphi)$.

Now we build a mapping $\mu : A \to A$, such that $\mu(a) = \sigma(a)$ for every $a \in A$, and $\alpha : F \to F_1$, such that $f^\alpha = f^\sigma$ for every $f \in F$.

Here we define a multi-model (A, Φ, F_1), which is isomorphic to the given multi-model (A, Φ, F). Moreover, $Aut(f^\alpha) = \sigma Aut(f)\sigma^{-1}$, which means that these two multi-models are automorphically equivalent.

Thus, any automorphism of the algebra A induces a multi-model which is isomorphic and, consequently, automorphically equivalent to the given one.

3.2. Construction of Automorphically Equivalent Multi-Models Using $\overline{f}$. Let (A, Φ, F) be a multi-model. For every $f \in F$ we build $\overline{f}$ using the following rule: $\overline{f}(\varphi) = \overline{f(\varphi)}$, where bar denotes the complement in the corresponding Cartesian product. Denote by $\overline{F}$ the set of all $\overline{f}$. Let us consider a mapping $\alpha : F \to \overline{F}$ defined by $f^\alpha = \overline{f}$. It is clear that this map is a bijection. Now we use an identity transformation mapping $\mu : A \to A$. Obviously, we have $Aut(f) = Aut(\overline{f})$, and the equality $Aut(\overline{f}) = \mu Aut(f)\mu^{-1}$, where $f \in F$, takes place.

Thus, the multi-models (A, Φ, F) and $(A, \Phi, \overline{F})$ are automorphically equivalent. Note that they are not isomorphic.

4. Some Properties of Automorphic Equivalence

Our next aim is to investigate some properties of automorphic equivalence. With this end we consider graphs as a particular example of models.

4.1. Graphs. To each graph $G = (V, E)$, where V is a set of vertices and E is a set of edges, corresponds a model (V, φ, E) where V is a domain of the model, φ is the only relation that exists on the graph and defines edges between vertices, E is an interpretation of the relation φ on the domain V, i.e., $E \subseteq V \times V$.

As usual, an automorphism of a graph is a permutation on the set of vertices preserving edges. All automorphisms of a graph constitute a group, which is a subgroup of symmetric group acting on vertices. The automorphism group of a graph characterizes its symmetries, and, therefore, is very useful in determining some of its properties.

4.2. Investigation of a Tree Structure Preservation. In graph theory a tree is a graph in which any two vertices are connected by exactly one path. Alternatively, any connected graph with no cycles is a tree. We show that automorphic equivalence of graphs does not preserve tree structure of a graph.

4.2.1. *Building Automorphically Aquivalent Multi-Models Using Algebra Automorphism.* Assume that $G_1 = (V_1, E_1)$ is a tree and $G_2 = (V_2, E_2)$ is an arbitrary graph. Let G_1 and G_2 be automorphically equivalent graphs. It means that there exists a bijection α that transforms E_1 to E_2, a bijection μ that transforms V_1 to V_2 and $Aut(G_1)$ and $Aut(G_2)$ are conjugated: $Aut(G_2) = \mu Aut(G_1)\mu^{-1}$.

Now let us consider the following example. We have two graphs on Figure 1:

.

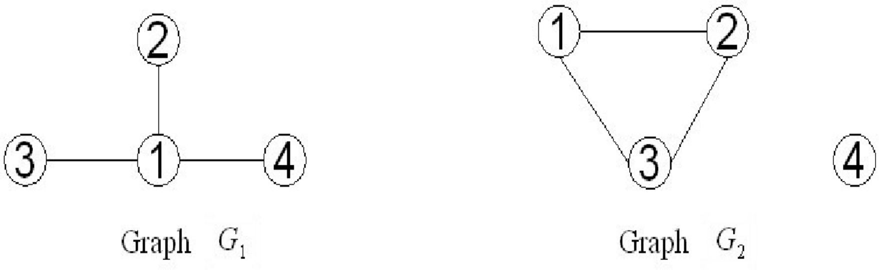

Figure 1.

The set of vertices for graph G_1 is $V_1 = \{1, 2, 3, 4\}$ and set of edges is $E_1 = \{e_1^1 = (1, 2), e_1^2 = (2, 1), e_1^3 = (1, 3), e_1^4 = (3, 1), e_1^5 = (1, 4), e_1^6 = (4, 1)\}$. The automorphisms group consists of all permutations of 2, 3 and 4.

For graph G_2 we have $V_2 = \{1, 2, 3, 4\}$ and the set of edges is $E_2 = (e_2^1 = (1, 2), e_2^2 = (2, 1), e_2^3 = (1, 3), e_2^4 = (3, 1), e_2^5 = (2, 3), e_2^6 = (3, 2))$. The automorphisms group consists of all permutations of 1, 2 and 3.

Let us demonstrate that these two graphs are automorphically equivalent.

(1) There exists a bijection $\alpha : E_1 \to E_2$, $\alpha = \begin{pmatrix} e_1^1 & e_1^2 & e_1^3 & e_1^4 & e_1^5 & e_1^6 \\ e_2^1 & e_2^2 & e_2^3 & e_2^4 & e_2^5 & e_2^6 \end{pmatrix}$

(2) There exists a bijection μ as defined below (here we use cyclic representation of permutations):

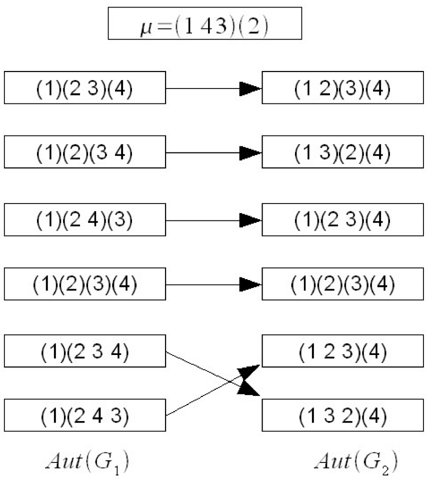

Figure 2.

Groups of automorphisms are conjugated by bijection μ. Therefore, graphs are automorphically equivalent. This example illustrates that automorphic equivalence of two graphs does not preserve the basic characteristics of those graphs, like being a tree.

Let us consider an additional example with directed graphs (Figure 3).

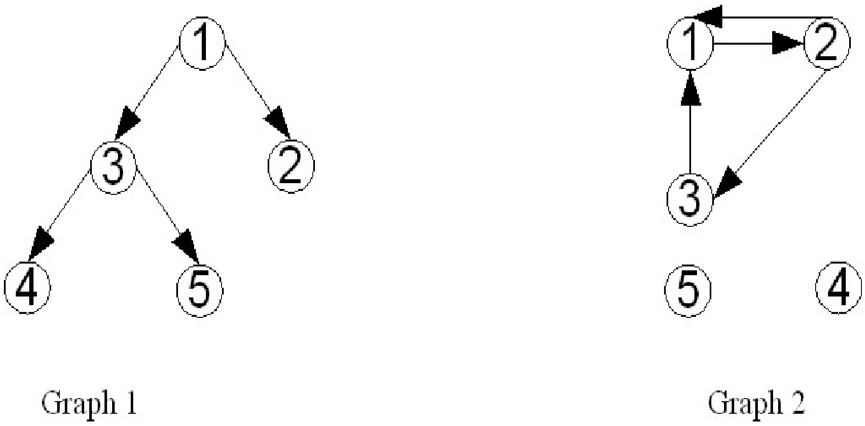

Figure 3.

The set of vertices for graph G_1 is $V_1 = \{1, 2, 3, 4, 5\}$ and the set of edges is $E_1 = (e_1^1 = (1, 2), e_1^2 = (1, 3), e_1^3 = (3, 4), e_1^4 = (3, 5))$. The automorphisms group consists of the following permutations $\binom{12345}{12354}$ and $\binom{12345}{12345}$.

The vertices of graph G_2 are $V_2 = \{1, 2, 3, 4, 5\}$ and edges of this graph are $E_2 = (e_2^1 = (1, 2), e_2^2 = (2, 3), e_2^3 = (3, 1), e_2^4 = (2, 1))$. The automorphisms of the graph are $\binom{12345}{12354}$ and $\binom{12345}{12345}$.

We can see that these two graphs are automorphically equivalent.

(1) There exists bijection $\alpha : E_1 \to E_2$

$$\alpha = \begin{pmatrix} e_1^1 & e_1^2 & e_1^3 & e_1^4 & e_1^5 & e_1^6 \\ e_2^1 & e_2^2 & e_2^3 & e_2^4 & e_2^5 & e_2^6 \end{pmatrix}$$

(2) There exists bijection μ as defined below (here we use cyclic representation of permutations):

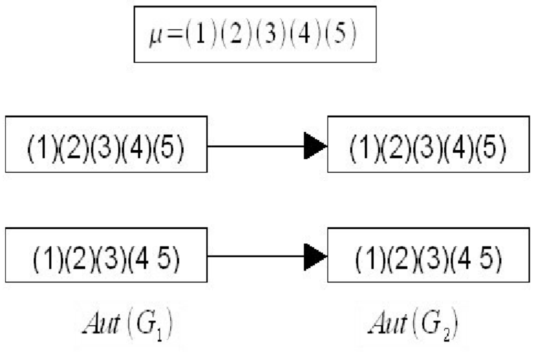

Figure 4.

Groups of automorphisms are conjugated by bijection μ. Therefore, graphs are automorphically equivalent. Thus, the graphs on Figure 3 are automorphically equivalent but the property of being a tree is not preserved since G_2 is not a tree.

4.2.2. *Constructing Automorphically Equivalent Multi-Models Using Complement Interpretation.* For multi-model of the given graph $(A, \Phi, F) = (V, \Phi, E)$ we will build an automorphically equivalent multi-model $(A, \Phi, \overline{F}) = (V, \Phi, \overline{E})$ using the approach from Section 3.2.

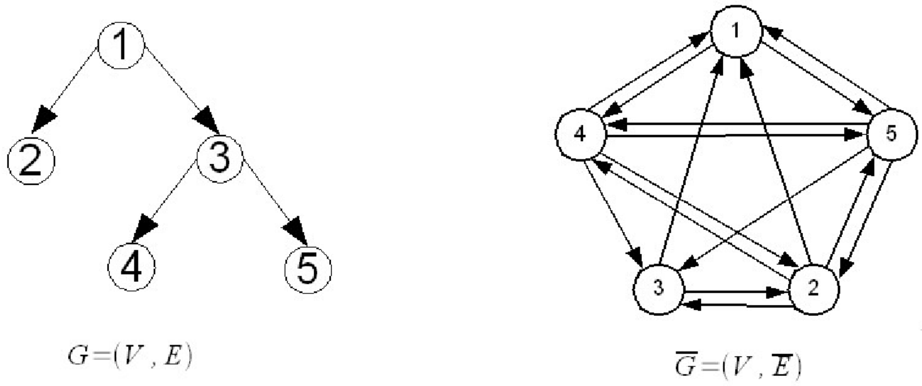

Figure 5.

In order to build the second graph we connect two vertices that were not connected in the original graph and, vice versa, if vertices were connected in the original graph then we do not connect them in the new one.

Here we have just one interpretation in each multi-model, so there is no problem to build a bijection $\alpha : F \to \overline{F}$. This bijection is $\alpha(E) = \overline{E}$.

The second bijection $\mu : A \to A$ is the identity transformation and in this example it is the permutation $\mu = \binom{1\,2\,3\,4\,5}{1\,2\,3\,4\,5}$. According to the criterion of automorphic equivalence two our multi-models are automorphically equivalent.

Since the second graph is not a tree, automorphic equivalence does not preserve this property.

4.3. Graph Connectedness and Automorphic Equivalence.

4.3.1. *Building Automorphically Equivalent Multi-Models Using Algebra Automorphism.* From the example in subsection 4.1.1, we can see that the first graph is connected (tree) but the second one is not. As we proved already two corresponding multi-models are automorphically equivalent. So, automorphic equivalence of multi-models does not keep connectedness of the graphs.

4.3.2. *Building Automorphically Equivalent Multi-Models Using Inverse Interpretation.* Again we start from multi-model of connected graph $(A, \Phi, F) = (V, \Phi, E)$ and we build automorphically equivalent multi-model by the second approach, $(A, \Phi, \overline{F}) = (V, \Phi, \overline{E})$.

Figure 6.

Here we use bijection $\alpha : F \to \overline{F}$, where $\alpha(E) = \overline{E}$, and identical bijection $\mu : A \to A$, i. e., $\mu = \binom{1\,2\,3\,4}{1\,2\,3\,4}$. Then the following equality takes place $Aut(\overline{f}) = \mu Aut(f)\mu^{-1}$. This shows that two multi-models are automorphically equivalent.

In this second example we also see that automorphic equivalence does not save connectedness of the graph.

5. Knowledge Bases Informational Equivalence Verification Algorithm Outline

It is not clear *a priori* whether the information equivalence problem for knowledge bases is algorithmically decidable. By its nature a knowledge base is an infinite object since it possesses an infinite number of queries (see [**17**] for precise definitions).

We assume that every knowledge under consideration is represented by three components:

1) *The description of knowledge.* It is a syntactical part of knowledge, written out in the language of the given logic.

2) *The subject of knowledge* which is an object in the given applied field, i.e., an object for which we determine knowledge.

3) *The content of knowledge* (its semantics).

Let (A, Φ, F) be a multi-model. To every such multi-model corresponds a knowledge base $KB = KB(A, \Phi, F)$. Roughly speaking,

A knowledge base $KB = KB(A, \Phi, F)$ *consists of the category of knowledge description* $L_\Theta(\Phi)$ *and the categories of knowledge content* $K_{\Phi\Theta}(f)$. *They are related by the functors*

$$Ct_f : L_\Theta(\Phi) \to K_{\Phi\Theta}(f).$$

These functors Ct_f transform knowledge description to content of knowledge.

We view the description T as a *query* to a knowledge base, and $A = T^f$ as a *reply to this query.*

Let $\mathrm{KB}(A, \Phi_1, F_1)$ and $\mathrm{KB}(B, \Phi_2, F_2)$ be two knowledge bases.

THEOREM 5.1. *The problem of informational equivalence of two knowledge bases is algorithmically decidable if the the corresponding multimodels are finite.*

The proof relies on the following theorem [**17**]

THEOREM 5.2. *Two knowledge bases* $KB(A, \Phi_1, F_1)$ *and* $KB(B, \Phi_2, F_2)$ *are informationally equivalent if and only if their finite multi-models are automorphically equivalent.*

In other words two knowledge bases are informationally equivalent if and only if the corresponding multi-models are automorphically equivalent.

This theorem provides a possibility to build an effective algorithm for knowledge bases informational equivalence verification. That is:

LEMMA 5.3. *For two given multi-models* (A, Φ_1, F_1) *and* (B, Φ_2, F_2) *there exists algorithm for their automorphic equivalence verification.*

We prove this lemma by constructing a formal algorithm:

begin verification
if algebras$A$ and B are not isomorphic
  exit with **not automorphically equivalent**
else
  find group of automorphisms $Aut(A)$
  find group of automorphisms $Aut(B)$
  find set Aut_{FA} of all subgroups $Aut(f)$ from $Aut(A)$, where $f \in F_1$
  find set Aut_{FB} of all subgroups $Aut(f)$ from $Aut(A)$, where $f \in F_2$
  build bipartite graph (V, E) of conjugated (f_i, f_j) from $F_1 \times F_2$
  if exists f_i without corresponding f_j
    exit with **not automorphically equivalent**
  else
    find bijection $\alpha : F_1 \to F_2$ so that $(f, f^\alpha) \in E$ for all $f \in F_1$
    if α exists
      exit with **automorphically equivalent**
    else
      exit with **not automorphically equivalent**
end verification

References

[1] Alperin J. L., and Bell R. B. (1995). Groups and Representations. New York: Springer-Verlag.

[2] Awodey S. (2006). Category Theory (Oxford Logic Guides 49). Oxford University Press.

[3] Bernus P., Mertins K., and Schmidt G. (1998). Conceptual Graphs. Handbook on Architectures of Information Systems, Springer-Verlag, Berlin, pp. 287-311.

[4] Burrow M. (1993). Representation Theory of Finite Groups. New York: Dover, 1993.

[5] Conway J. H., Curtis R. T., Norton S. P., Parker R. A., and Wilson R. A. (1985). Atlas of finite groups. Oxford University Press.

[6] Davis R., Shrobe H., and Szolovits P. (1993). What Is a Knowledge Representation? AI Magazine, 14(1):17-33.

[7] Grabmeier J., Kaltofen E., and Weispfenning V. (2003). Computer Algebra Handbook. Springer Verlag.

[8] Holt D. F. (2005). Handbook of Computational Group Theory. In the series 'Discrete Mathematics and its Applications', Chapman and Hall/CRC, xvi + 514 p.

[9] Lawvere W., and Rosebrugh R. (2003). Sets for mathematics. Cambridge University Press.

[10] Lawvere W., and Schanuel S. (1997). Conceptual mathematics: a first introduction to categories. Cambridge University Press.

[11] Lenat, D. B. (1995). Steps to Sharing Knowledge. In Toward Very Large Knowledge Bases, edited by N.J.I. Mars. IOS Press.

[12] Markman A. B. (1998). Knowledge Representation, Lawrence Erlbaum Associates.

[13] Mineau G., Moulin B., and Sowa J. F. (1993). Relating diagrams to logic. Conceptual Graphs for Knowledge Representation, Springer-Verlag, New York.

[14] Pitts A. M. (2000). Categorical Logic. Handbook of Logic in Computer Science, Vol.5, Oxford: Oxford Unversity Press, 39128.

[15] Plotkin B.I. (1999). Algebra, categories and databases, Handbook of algebra, v.2, Else-vier, Springer, pp. 81-148.

[16] Plotkin B.I. (2002). Seven Lectures in Universal Algebraic Geometry, Preprint, Arxiv math, GM/0204245, pp. 87.

[17] Plotkin B.I., and Plotkin T. (2001). Geometrical aspect of databases and knowledge bases, vol. 46, Algebra Universals, pp. 40394070.

[18] Plotkin B.I. (1993). Universal algebra, algebraic logic and databases, Kluwer.

[19] Plotkin T. (1995). Equivalence of relational databases. Proceeding of the Second International Workshop on Advances in Databases and Informational Systems (ADBIS95), Moscow, June 1995. Volume 1: Regular papers. Moscow: Phasis.

[20] Plotkin T. (1996). Relational databases equivalence problem, Advances of databases and information systems, Springer, pp. 391-404.

[21] Plotkin T., and Knyazhansky M. (2004). Verification of knowledge bases informational equivalence, Journal Scientific Israel Technological Advantages, Issues 1-2, Vol. 6.

[22] Seress A. (2003). Permutation Group Algorithms. Cambridge Tracts in Mathematics, vol 152, Cambridge University Press.

[23] Sowa John F. (1999). Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA.

[24] Tarski A. (1931). Sur les ensembles definissables de nombres reels, Fundamenta Mathematicae 17, 210-239; English translation in Tarski (1983), pp. 110-142.

[25] Tarski A. (1983). Logic, Semantics, Metamathematics. Oxford University Press, Oxford. Second edition. (First edition 1956)

[26] Tepfenhart W. and Cyre W. (1999). Conceptual graphs: draft proposed American National Standard. Conceptual Structures: Standards and Practices, Lecture Notes in AI 1640, Springer-Verlag, Berlin, pp. 1-65.

[27] Ullman J. D. (1988). Principles of Database and Knowledge-base systems. Computer Science Press, pp 24-29.