Isotypeness of models and knowledge bases equivalence

Elena Aladova, Eugene Plotkin and Tatjana Plotkin

Abstract. The aim of this work is to study the notions of elementarily equivalent and isotypic knowledge bases. We prove that isotypic knowledge bases are informationally equivalent.

Mathematics Subject Classification (2010). Primary 08A70; Secondary 68T30.

Keywords. Knowledge base, model, algebra, elementary equivalence, isotypic knowledge bases.

1. Introduction

1.1. Motivation

The Knowledge Bases world is extremely diverse and we would like to think over the question about some classes of equivalent bases, yet not defining what do we exactly mean by equivalence by now.

The outcome we expect of equivalence is that the bases work equally. The knowledge need not be represented in exactly the same way but there should be a transition allowing to infer the replies in one knowledge base using the knowledge kept in another one. The same observation relates also to queries, that is, queries to a knowledge base need not be literally the same but they have to be considered up to a certain syntactical/semantical equivalence. So, the knowledge bases are *informationally equivalent* if they produce the equivalent knowledge under equivalent queries. One can also say that two knowledge bases are *informationally equivalent* if the whole information that can be retrieved from one of them could be also obtained from the other one and vice versa.

1.2. Equivalence of databases

Genetically, the equivalence problem for knowledge bases goes back to the similar one for databases. To the best of our knowledge, it was first posed in [6] and [1] and gave rise to the notion of databases schemes equivalence. Algorithms for verification of databases equivalence using database schemes were proposed by Beniaminov, Beeri-Mendelzon-Sagiv-Ullman and others. In this setting two relational database schemes are equivalent if their sets of fixed points coincide. Correspondingly, two relational databases are equivalent if their sets of all fixed points intersected with the sets of feasible instances coincide. This and other approaches to the database equivalence problem had been studied in numerous papers (see [7], [2], [31], [14], [4], [3], [29], etc.).

1.3. Symmetries and automorphic equivalence of knowledge bases

Informational equivalence for knowledge bases requires a more extended technique (see [30] and references therein). Being inspired by Galois theory of algebras of relations invented by M. Krasner [17], we used the notion of automorphic equivalence of knowledge bases in [30], [28].

In plane words the general idea behind this approach is as follows.

We are very grateful to the anonymous reviewers for their valuable and helpful comments and suggestions.

We would like to have a calculable invariant of a knowledge base, i.e., an object assigned, which can be used for distinguishing the properties of knowledge bases in question. Moreover, we expect coincidence of these invariants as a criterion of the informational equivalence of knowledge bases. Such a criterion is provided by the universal ideas of the Galois theory and should use a properly defined group of symmetries of a knowledge base, a.k.a. the group of automorphisms of a knowledge base.

Relying rather on human perception than on mathematics let us call an object rigid, if it can be "rigidly" defined by its group of symmetries. So, if a knowledge base is, in this sense, rigid, it can be characterized by its group of automorphisms, which is a calculable object, much simpler than the initial structure of a knowledge base.

The above observations resulted in the following definition (see [27], [30] for details). Let $\mathfrak{A}$, $\mathfrak{B}$ be two knowledge bases, and $Aut(\mathfrak{A})$, $Aut(\mathfrak{B})$ be their groups of automorphisms.

Definition 1.1. The knowledge bases $\mathfrak{A}$ and $\mathfrak{B}$ are called automorphically equivalent if their Galois groups $Aut(\mathfrak{A})$ and $Aut(\mathfrak{B})$ are conjugated.

It turns out, that knowledge bases are rigid ([16],[27], [30]):

Theorem 1.2. *Two finite knowledge bases $\mathfrak{A}$ and $\mathfrak{B}$ are informationally equivalent if and only if they are automorphically equivalent.*

In fact, this theorem reduces the problem of informational equivalence of knowledge bases to the well-developed conjugation problem for groups ([9], [22], [32]).

1.4. The main objective of the paper

Theorem 1.2 provides an algebraic criterion for the informational equivalence of knowledge bases. It indicates the fact that knowledge bases have a formidable group of symmetries which makes it possible to calculate how far is a knowledge base from the other one.

The current paper makes another accent and emphasizes logical characteristics instead of computability. In fact, we pursue the goal of the characterization of the measure of "sameness" of two different knowledge bases more from the positions of logic and geometry than from the side of algebra. Clearly, this is yet another incarnation of the same essence, and ideally in the end one should come up with the similar result.

So, our aim is to study the logical invariants of knowledge bases and to use for that the ideas of logical geometry. We will see that the notion of *elementary equivalence* of knowledge bases is not enough for this aim, and that the appropriate notion which works out is the notion of *isotypic knowledge bases*.

The main result of the paper is the following

Theorem. Isotypic knowledge bases are informationally equivalent.

Its proof is based on several preliminary facts and is contained in the last section of the paper.

1.5. Preliminaries about the model of a knowledge base

In order to operate with mathematical characteristics of a knowledge base we use the model, described in Section 2. Beforehand, one should underline that due to the complexity of the object each model of a knowledge base is just a rough approximation to reality, suffering from a variety of assumptions. Our model is not an exception. Nevertheless we dare state the results about arbitrary knowledge bases in terms of our model. In any case it allows one to use the machinery of algebra and logic and to treat the common sense notions in a proper formal way.

1.6. Mathematical engine

We assume that queries to a knowledge base are written in a first-order language. The corresponding logic has syntax and semantics which go side by side. We express syntax and semantics by adequate algebraic notions allowing to represent all transitions by some functions preserving operations. This approach naturally requires an application of structures of algebraic logic [13], [23]. In particular, a consistent application of Halmos algebras is of ultimate importance.

The key point about Halmos algebras is that they are related to first-order logic in a way analogous to the relationship between Boolean algebras and propositional logic. The immediate advantage of this phenomenon is that we can view queries to a knowledge base and replies to these queries as objects of the same nature, i.e., elements of Halmos algebras. Then the transition query-reply is a homomorphism of such algebras. This insight attracts to the play the notions of a kernel, of a quotient algebra, of a Galois connection and gives rise to a developed algebraic machinery.

There are two peculiarities of the algebraic logic characteristic for the aims of knowledge bases. First of all we need a multi-sorted variant of logic, which corresponds to data and information of the different nature. Secondly, we need to construct an algebraic logic and the corresponding logical geometry over an arbitrary set of predicates (cf. [10]). This construction is presented in Sections 4 and 5 and takes up a valuable part of the paper.

Finally, it is not realistic to implement all the needed steps of this route in a short paper, but we will formulate these steps and prove the main theorem on equivalence.

2. Preliminaries

For the sake of self-completeness we recall here the basic definitions and sketch the model of a knowledge base in use. This material is mostly known (see [30] and references therein). We begin with the definition of a model in the form of [23].

Definition 2.1. We define a model as a triple $\mathcal{M} = (H, \Psi, f)$, where H is a data domain, that is, an algebra in a variety of algebras Θ, Ψ is a set of symbols of relations, and f is an interpretation of these symbols as relations in H. Each symbol $\varphi \in \Psi$ has an arity n_φ. An interpretation of the relations is a map f which takes every $\varphi \in \Psi$ to the subset $f(\varphi)$ of the Cartesian product H^{n_φ} . Further on we sometimes denote the model (H, Ψ, f) by (f).

Remark 2.2. Note the misleading coincidence of the terms "model" and "model of a knowledge base". The first one is the main subject of model theory (see the textbook [19] and many others). The second term is used in the sense, which is customary for applications: "a mathematical model of something".

A knowledge under consideration is presented in three components:

- *Description of knowledge* is its syntactical component. It relies on a language in the given logic and describes what kind of information we would like to retrieve. Knowledge description consists of the set of sentences in the chosen language. Assuming our logic to be first-order, the description of knowledge is presented by a set of first-order formulas. The logic is given for the fixed variety Θ which serves as a data type in applications. We consider a first-order logic equipped with predicates (relations). We also assume that our language contains the predicate $\equiv$ and its interpretation is always equality, that is we deal with normal models.

  For the algebraic description of knowledge we use the free in Θ algebra $W(X)$ with the finite $X = \{x_1, \dots, x_n\}$ and the algebra of formulas $\Phi(X)$. Then the description of knowledge is a set of formulas T in $\Phi(X)$ or, more precisely, in the multi-sorted algebra of formulas $\tilde{\Phi}$ (see Subsection 4.6).

- The next component is the *subject area of knowledge* for which the content of knowledge should be computed. The subject of knowledge is represented by a model $\mathcal{M} = (H, \Psi, f)$. This triple presents a subject area where all replies to all queries are searched. We call the knowledge base finite if the algebra H is finite.
- The third component is the *content of knowledge*. Given a model (H, Ψ, f), to each description of knowledge $T \subset \Phi(X)$, there corresponds a content of knowledge A, where A is a subset in H^n. If we regard H^n as an affine space, then the content of knowledge A can be treated geometrically as the set of points in the affine space which satisfy the description T of the searched knowledge. The most of geometry is involved in this third component.

Now we are able to complete the definition of a knowledge base. Let us define a multi-model (H, Ψ, F) as a set of models (H, Ψ, f), where f runs some set F of interpretations of symbols of relations Ψ in H. In fact, multi-models are needed in order to determine knowledge bases with changing interpretations of the description of knowledge.

We need to introduce the following categories.

The first one is the category of logical knowledge description $\widetilde{\Phi}_{(f)}$. Here, the notation $\widetilde{\Phi}_{(f)}$ with subscript (f) indicates the fact that knowledge descriptions are considered under the given model (H, Ψ, f). Objects of this category are pairs of the form (X, T), where X is a finite set of variables and T is a set of first order formulas written in the variables from X. Morphisms in $\widetilde{\Phi}_{(f)}$ will be defined in Section 6. The categories $LK_\Theta(f)$ of knowledge content, where f runs over the set F have the objects of the form (X, A), where A is a subset in an affine space over the given model. Their morphisms are naturally defined (see Section 6 for details).

Let us denote the knowledge base over the given multi-model (H, Ψ, F) by $KB(H, \Psi, F)$.

Definition 2.3. A knowledge base $KB = KB(H, \Psi, F)$ consists of the category of knowledge description $\widetilde{\Phi}_{(f)}$ and the categories of knowledge content $LK_\Theta(f)$. They are related by the contravariant functors

$$Ct_f : \widetilde{\Phi}_{(f)} \to LK_\Theta(f).$$

These functors Ct_f transform knowledge descriptions to contents of knowledge and will be defined in Subsection 6.1.

In the three subsequent sections we present mathematical apparatus needed to formalize the intuitive definition of a knowledge base.

3. Logic for knowledge bases

In this Section we give a brief account of the logic which will be used for knowledge description and which will be subject to algebraization. Note, however, that the precise definition of the logic in use is left till Section 4, and is given in terms of algebraic logic.

We will start with the formal syntactic description of a first-order language used for the description of knowledge. Each language assumes some stock of variables, which serve as an alphabet, and a number of rules which allow to construct words from a given alphabet. Formalizing all this:

Definition 3.1. A language $\mathbb{L}$ is given by specifying the following data.

1. A set of variables $X = \{x_1, x_2, \dots, x_n, \dots\}$. This set can be finite or infinite. The generic situation is an infinite $X = X^0$.
2. A set $\mathcal{F}$ of function symbols p given together with their arities $n_p \geq 0$.
3. A set Ψ of relation symbols φ given together with their arities $n_\varphi \geq 1$. Relation symbols $\varphi \in \Psi$ are also called predicate symbols.
4. A set $\mathcal{C}$ of constant symbols. These symbols are also treated as function symbols of zero arity.
5. The symbols of logical connectives $\neg$, $\vee$.

6. The symbol of existential quantifier $\exists$.
7. The punctuation symbols "(", ")", ",".

The universal quantifier $\forall$ can be defined in terms of the existential quantifier $\exists$ and the connective $\neg$ as $(\forall x_i u) = \neg(\exists x_i(\neg u))$.

Now we need to define the set of formulas $\mathbb{F}$ of $\mathbb{L}$.

Definition 3.2. Terms in a language $\mathbb{L}$ are defined inductively:

1. Variables are terms.
2. Constant symbols of $\mathbb{L}$ are terms.
3. If $t_1, \ldots, t_{n_p}$ are terms and p is a functional symbol of arity n_p, then $p(t_1, \ldots, t_{n_p})$ is a term.
4. There are no other terms.

Definition 3.3. An atomic formula is a formula of the form $\varphi(t_1, \ldots, t_{n_\varphi})$, where φ is a relation symbol of arity n_φ and $t_1, \ldots, t_{n_\varphi}$ are terms.

Definition 3.4. Formulas in a language $\mathbb{L}$ are defined inductively:

1. Atomic formulas are formulas.
2. If u_1 and u_2 are formulas, then $\neg u_1$, $(u_1 \vee u_2)$ are formulas.
3. If u is a formula, then $\exists x_i u$ is a formula, where x_i is a variable.
4. There are no other formulas.

The set of axioms and the set derivation rules are peculiar to each first-order theory. Their logical core is described in many textbooks (see, for instance [20]).

Definition 3.5. A formula u is derivable from a set of formulas T if and only if there exists a finite sequence of formulas

$$u_0, u_1, \ldots, u_n = u,$$

whose last term u_n is u, such that u_0 either belongs to T or is an axiom, and every formula u_i, $1 \leq i \leq n$, is either an axiom, or an element of T, or the result of applying a derivation rule to some of preceding formulas in the sequence.

If u is derivable from axioms, we say that u is a *theorem* of the logical calculus and denote this by $\vdash u$. If we fix a set of formulas T, then by theory T we mean the set of all formulas derivable from T.

The next step is related to the semantical part of knowledge, namely to a subject of knowledge. The corresponding logical notion is a model, which was defined as a triple $\mathcal{M} = (H, \Psi, f)$ where H is an algebra in some variety Θ, Ψ is a set of symbols of relations and f is a realization which makes symbols $\varphi \in \Psi$ relations in H^{n_φ}. In this triple, realization of functional symbols from $\mathcal{F}$ is hidden in the signature of operations related to the variety Θ. Actually, the realization is hidden not only in a set of symbols, but in the algebra H. As for the system Ψ of symbols of relations, its choice is determined by problems in applications.

The traditional definition of what does it mean for a tuple $\bar{a} = (a_1, \ldots, a_n) \in H^n$ to satisfy a formula $u(x_1, \ldots, x_n)$ on the model $\mathcal{M}$ is given inductively (see [19]). We say that u is valid on $\mathcal{M}$ if every $\bar{a} \in H^n$ satisfies u. Later on we define the value of the formula $u(x_1, \ldots, x_n)$ in the point $\bar{a} = (a_1, \ldots, a_n)$ by means of algebraic logic.

Let us have a model $\mathcal{M} = (H, \Psi, f)$. The set of all sentences valid on $\mathcal{M}$ is called *elementary theory* of $\mathcal{M}$. Denote the elementary theory of $\mathcal{M}$ by $Th(\mathcal{M})$.

Definition 3.6. Two models $\mathcal{M}_1 = (H_1, \Psi, f_1)$ and $\mathcal{M}_2 = (H_2, \Psi, f_2)$ are called elementarily equivalent, if their elementary theories coincide:

$$Th(\mathcal{M}_1) = Th(\mathcal{M}_2).$$

Coincidence of elementary theories of models $\mathcal{M}_1$ and $\mathcal{M}_2$ means semantically that every sentence valid on one model is valid on another one and vice versa.

Since an algebra $H \in \Theta$ itself can be viewed as a model, one can speak also about elementary theory of the algebra H.

4. Algebraic Logic for knowledge bases

4.1. Algebraic logic: an informal look at the subject

Our nearest aim is to convert logic to algebra not loosing all interactions between the syntax and semantics. This leads to a geometrical intuition, which in turn yields the description of the content of knowledge.

The idea of algebraization of logic goes back to E. Schroeder, who has published three volumes of "Lectures on the algebra of logic" at the turn of the 19th and 20th centuries. In its present form this idea appeared in the works of A. Tarski and P. Halmos. Our main tool will be the Halmos algebras which were presented under the name of polyadic algebras in [13], and introduced in [23] for the multi-sorted case needed for the logical geometry and database/knowledge base applications.

In the very plain words, Halmos algebras are intended to replace the logical system by an algebraic system, equivalent to the original logic. There are several advantages of this idea. We distinguish only the ability to make all the passages between syntax and semantics to be homomorphisms.

This way of thinking is well-known for Boolean algebras. Indeed, to think about propositional sentences as elements in a Boolean algebra is already a folklore. It is not so common to emphasize that this Boolean algebra is the free Boolean algebra, that is, it can be mapped homomorphically to any other Boolean algebra. However, this freeness property is crucial from the point of view of representation of Boolean formulas as elements in the Boolean algebra of all subsets of some set. This passage is explicitly stated in the Stone's celebrated theorem. Philosophically, this means that the formulas with the logical connectives as operations - a.k.a. syntax, are represented by the sets with intersections and unions as operations - a.k.a. semantics.

The *principal goal of algebraization of a first order logic* is to do essentially the same for the logic enriched by quantifiers and predicates. With this end it is clear that the appropriate (in the sense of algebraization of a first-order logic) algebra should be a Boolean algebra, equipped with quantifiers, defined in an abstract way as operations on Boolean algebras. This organism has to enjoy some conditions, which are going to be the defining axioms of the algebra. Note that operations of replacement of variables require a special attention. This was not the case for the propositional calculus algebraization, since all replacements of variables in this calculus respect logical connectives and are converted in Boolean algebras to homomorphisms of such algebras. This is the case with the appearance of quantifiers, since replacements and quantifiers do not commute and are subject to more complex rules. When this difficulty is treated by additional axioms the resulting object is called a Halmos algebra.

4.2. Extended Boolean algebras

This subsection and the consequent ones deal with a formal implementation of the ideas outlined in 4.1. Let $\mathcal{B}$ be a Boolean algebra.

Definition 4.1. An existential quantifier on a Boolean algebra $\mathcal{B}$ is a map $\exists : B \to B$ subject to conditions:

1. $\exists 0 = 0$.
2. $b \leq \exists b$.
3. $\exists(b_1 \wedge \exists b_2) = \exists b_1 \wedge \exists b_2$, where b_1, b_2 are elements of $\mathcal{B}$, 0 is the zero element of $\mathcal{B}$.

Let $W(X)$ be the free algebra in the variety Θ over the set of free generators X. Elements $w \in W(X)$ play the role of terms in our logic which is associated with the variety Θ, that is, all functional symbols are incorporated in the elements w.

Definition 4.2. We call a Boolean algebra $\mathcal{B}$ an extended Boolean algebra over the free in Θ algebra $W(X)$, if

1. There are defined quantifiers $\exists x$ for all $x \in X$ in $\mathcal{B}$ with $\exists x \exists y = \exists y \exists x$ for all $x, y \in X$.

2. To every atomic formula $\varphi(w_1, \ldots, w_{n_\varphi}) \in \mathbb{L}$, where n_φ is the arity of φ, there corresponds a constant $\varphi(w_1, \ldots, w_{n_\varphi})$ in $\mathcal{B}$.

For example, if φ is a binary symbol $\equiv$ which stands for the equality predicate (see [20] for the definition), then the corresponding extended Boolean algebra possesses the following properties:

E. To every pair $w, w' \in W(X)$, there corresponds a constant (called an equality) in $\mathcal{B}$, denoted by $w \equiv w'$. Here,

E1. $w_1 \equiv w_1' \leq w_1' \equiv w_1$.

E2. $w \equiv w$ is the unit of the algebra $\mathcal{B}$.

E3. $w_1 \equiv w_2 \wedge w_2 \equiv w_3 \leq w_1 \equiv w_3$.

E4. For every n-ary operation $\omega \in \Omega'$, where Ω' is a signature of the variety Θ, we have

$$w_1 \equiv w_1' \wedge \ldots \wedge w_n \equiv w_n' \leq w_1 \ldots w_n \omega \equiv w_1' \ldots w_n' \omega.$$

Condition E4 really means that equalities respect all operations on $W(X)$. Since $\varphi(w_1, \ldots, w_{n_\varphi})$ is a constant in $\mathcal{B}$, all endomorphisms of Boolean algebras leave it unchanged.

4.3. Algebras $\Phi(X)$ and $Hal_\Theta^X(f)$

Fix a set of variables X, take the free algebra $W(X)$ in the variety Θ and denote by M_X the set of all atomic formulas $\varphi(w_1, \ldots, w_{n_\varphi})$, $w_i \in W(X)$. Define the signature of operations Ω_X by

$$\Omega_X = \{\vee, \wedge, \neg, \exists x, M_X\}, \text{ for all } x \in X.$$

Denote by $\mathfrak{L}_X$ the absolutely free algebra (term algebra) in this signature over the atomic formulas $\varphi(w_1, \ldots, w_{n_\varphi})$, where each φ is a relational symbol from Ψ, and $w_1, \ldots, w_{n_\varphi}$ lie in $W(X)$ (we assume that all punctuation symbols belong to each signature).

For example, let Θ be the variety of semigroups, $X = \{x_1, x_2\}$, $W = W(X)$. Suppose that "$\equiv$" is the only symbol of relations. Then the formula $(x_1 x_2 x_1^2 \equiv x_2 x_1) \wedge (x_1 \equiv x_2)$ belongs to $\mathfrak{L}_X$, while the formula $x_1 x_2 x_1^2 = (x_2 x_1 \vee x_1)$ does not belong.

Define the relation τ_X on $\mathfrak{L}_X$ by: $u \tau_X v$ if and only if $\vdash (u \to v) \wedge (v \to u)$, where $u, v \in \mathfrak{L}_X$. In other words, two formulas u and v are claimed equivalent if each of them is derivable from the other (cf. Definition 3.5). It is easy to see that τ_X is a congruence on $\mathfrak{L}_X$ called Lindenbaum-Tarski equivalence.

Definition 4.3. The quotient algebra $\Phi(X) = \mathfrak{L}_X / \tau_X$ is called the algebra of one-sorted formulas associated with a first-order calculus.

All Boolean operations and quantifiers on $\Phi(X)$ are naturally inherited from $\mathfrak{L}_X$.

Proposition 4.4. *Algebra of formulas $\Phi(X)$ is an extended Boolean algebra.*

We skip the proof of this proposition, which amounts to a direct check.

So, we have an algebra of formulas over X, which makes us free in use of the syntax of a language, and, correspondingly, we have a necessary algebraic tool for a knowledge description. Our next concern is semantics, that is, the subject area of knowledge and the content of knowledge.

First of all note that there is a bijection between H^n and the set $Hom(W(X), H)$ of all homomorphisms from $W(X)$ to H. Here n is the cardinality of X, that is $|X| = n$. Take the Boolean algebra of all subsets in $Hom(W(X), H)$ and denote it by $Bool(W(X), H)$.

Note that from now on we will use the language of homomorphisms, since it allows formulation of many algebraic notions in the most natural way. However, keeping in mind the duality between the languages of homomorphisms and n-tuples is quite important for applications.

Define existential quantifiers $\exists x, x \in X$ to be operations in the given Boolean algebra. For each set of points $A \subset Hom(W(X), H)$ define a set $B = \exists x A$ by the rule: $\mu \in B$ if and only if there exists $\nu \in A$, satisfying the condition $\mu(x') = \nu(x')$ for every $x' \in X$, $x' \neq x$. The operations $\exists x$ meet the general idea of quantifier and the corresponding Definition 4.1.

Suppose we are given a model (H, Ψ, f). Each point $\bar{a} = (a_1, \ldots, a_n)$ in H^n can be viewed as a point $\mu : W(X) \to H$ with $\mu(x_i) = a_i$, $i = 1, \ldots, n$, $|X| = n$. Define what does it mean that the point μ satisfies the formula $\varphi(w_1, \ldots, w_m)$ under the interpretation f. It means that the tuple $(w_1^\mu, \ldots, w_m^\mu)$ belongs to the set $f(\varphi) \subset H^m$.

Denote by $[\varphi(w_1, \ldots, w_m)]_{(f)}$ the set of points in the space $Hom(W(X), H)$ satisfying the corresponding atomic formula. We consider the algebra $Bool(W(X), H)$ together with quantifiers and specified constants $[\varphi(w_1, \ldots, w_m)]_{(f)}$ as an extended Boolean algebra and denote such an algebra by $Hal_\Theta^X(H, \Psi, f)$ or, shortly, $Hal_\Theta^X(f)$.

So, we constructed two extended Boolean algebras $\Phi(X)$ and $Hal_\Theta^X(f)$ which are responsible for syntax and semantics, respectively. In order to enforce a knowledge base work, we have to relate them and, according to the general philosophy, this relation should be a homomorphism. The latter is possible, because both algebras have the same signature.

Later on we will define this important value homomorphism

$$Val_H^X : \Phi(X) \to Hal_\Theta^X(f).$$

Here,

$$Val_H^X(\varphi(w_1, \ldots, w_m)) = [\varphi(w_1, \ldots, w_m)]_{(f)}.$$

Note that the set $[\varphi(w_1, \ldots, w_m)]_{(f)}$ may be empty and in this case we identify it with the zero of the algebra $Bool(W(X), H)$.

4.4. Multi-sorted Halmos algebras $Hal_\Theta^X(f)$

The necessary step is to move to the case of many-sorted algebras. There are several reasons to do that. First of all our variables x_i can have quite a different nature and it would be illogical to dump them into one pile. The second reason is not so evident. The point is that once we want a geometric pattern, we need to work in a situation of finite-dimensional spaces. Namely, a multi-sorted glance at variables allows us to consider a bunch of interacting finite-dimensional spaces over finite X, running the set of sorts Γ, instead of one infinite-dimensional affine space over the infinite set X^0.

The first stage on the way is the algebra-category $Hal_\Theta(f)$. Define the category of all algebras $Hal_\Theta^X(f)$. Its objects are algebras of the form $Hal_\Theta^X(f)$, where only X is varied, while the morphisms s_* in this category are defined as follows.

To every homomorphism $s : W(X) \to W(Y)$, there corresponds a morphism

$$s_* : Hal_\Theta^X(f) \to Hal_\Theta^Y(f).$$

Let A be a subset in $Hom(W(X), H)$. Define $B = s_* A$ by the rule: $\mu \in B$ if $\mu s \in A$. This determines the morphism s_*. This s_* is a homomorphism of Boolean algebras

$$s_* : Bool(W(X), H) \to Bool(W(Y), H).$$

Interaction of s_* with quantifiers and atomic formulas and the corresponding axioms is a separate topic (cf. [26], Definition 2.7 or [25], Definition 1.3). Denote the list of the axioms by $\diamondsuit$. These axioms are technical and a part of them is deliberately omitted at the moment. Note only that the interaction of the morphisms s_* with atomic formulas is controlled by

$$s_*[\varphi(w_1, \ldots, w_m)]_{(f)} = \varphi[(sw_1, \ldots, sw_m)]_{(f)},$$

and $s_* s'_* = (ss')_*$.

We assume some familiarity with the notion of a multi-sorted algebra (see, for example [23]). The defined category $Hal_\Theta(f)$ can be viewed as a multi-sorted algebra

$$Hal_\Theta(f) = (Hal_\Theta^X(f),\ X \in \Gamma),$$

with objects as domains and morphisms

$$s_* : Hal_\Theta^X(f) \to Hal_\Theta^Y(f)$$

as multi-sorted operations. The axioms of $Hal_\Theta(f)$ are presented by the list $\diamondsuit$.

4.5. The variety of multi-sorted Halmos algebras

Denote the variety of multi-sorted algebras generated by all $Hal_\Theta(f)$, where f runs all models (H, Ψ, f), $H \in \Theta$, by Hal_Θ. Thus, the identities of the variety Hal_Θ are the identities of all possible $Hal_\Theta(f)$ (see $\diamondsuit$ and [26]). Algebras from $Hal_\Theta(f)$ are called multi-sorted Halmos algebras.

Any algebra in the variety Hal_Θ is a multi-sorted algebra of the form $\mathfrak{H} = (H_X,\ X \in \Gamma)$, where all H_X are extended Boolean algebras, and the operation $s_* : H_X \to H_Y$ corresponds to every $s : W(X) \to W(Y)$.

4.6. Multi-sorted algebra of formulas

Let us now move to the multi-sorted algebra of formulas. Recall that M_X is the set of all atomic formulas of the sort X. Take further

$$M = (M_X, X \in \Gamma).$$

Take the multi-sorted algebra of formulas $\tilde{\Phi} = (\Phi(X), X \in \Gamma)$ over M. It is constructed as the quotient-algebra $\mathfrak{L}/\tau$, where $\mathfrak{L}$ is the absolutely free over M algebra in the signature Ω, $\tau = (\tau_X, X \in \Gamma)$ is the multi-sorted Lindenbaum-Tarski congruence. Here the signature $\Omega = (\Omega_X, X \in \Gamma)$ is a multi-sorted signature, consisting of various Ω_X and symbols of operations of the form s_* which take the formulas from the domain $\Phi(X)$ to formulas from $\Phi(Y)$. So every domain $\Phi(X)$ contains elements of the form $s_* v$, where $v \in \Phi(Z)$, $Z \in \Gamma$, and in this sense, the domains $\Phi(X)$ are richer than the one-sorted algebras $\Phi(X)$. From now on $\Phi(X)$ always means a domain of $\tilde{\Phi}$.

One can consider multi-sorted algebras of different types, say, algebras H can be multi-sorted. Recall that in our definition of multi-sorted Halmos algebras only the set Γ of variables is multi-sorted.

Theorem 4.5 ([26]). *The multi-sorted algebra of formulas $\tilde{\Phi} = (\Phi(X), X \in \Gamma)$ is the free over the multi-sorted set of atomic formulas M algebra in the variety Hal_Θ.*

The proof of this theorem uses the semisimplicity of the algebras $Hal_\Theta^X(f)$ (cf. [13]) and extends, in fact, the proof of the theorem of Stone for the Boolean algebras.

Theorem 4.5 has a principal consequence. To each atomic formula $\varphi(w_1, \ldots, w_m)$ corresponds its value in $Hal_\Theta^X(f)$ denoted by $[\varphi(w_1, \ldots, w_m)]_{(f)}$. This determines the mapping $M \to Hal_\Theta(f)$. Since the algebra $\tilde{\Phi}$ is freely generated by the set M, we have a homomorphism

$$Val_{(f)} : \tilde{\Phi} \to Hal_\Theta(f).$$

We have also componentwise homomorphisms

$$Val_{(f)}^X : \Phi(X) \to Hal_\Theta^X(f),$$

which lead to the commutative diagram

$$\begin{array}{ccc} \Phi(X) & \xrightarrow{s_*} & \Phi(Y) \\ {\scriptstyle Val_{(f)}^X}\downarrow & & \downarrow{\scriptstyle Val_{(f)}^Y} \\ Hal_\Theta^X(f) & \xrightarrow{s_*} & Hal_\Theta^Y(f). \end{array}$$

It follows from Theorem 4.5 that the congruence τ coincides with the verbal congruence of the identities of the variety Hal_Θ. Thus, the mappings s_* satisfy all identities of Hal_Θ. In particular, they respect Boolean operations and interact with quantifiers and formulas from M_X according to $\diamondsuit$. Moreover, take $u \in \Phi(X)$. We want to calculate the value of $s_*u \in \Phi(Y)$. This can be done according to diagram above, since the mappings $s_* : Hal^X_\Theta(f) \to Hal^Y_\Theta(f)$ are defined explicitly (see Section 4.4).

We conclude with the fact that the upper row reflects syntax and knowledge description, while the lower one reflects semantics and knowledge content. Vertical arrows link description of knowledge with its content.

4.7. What is the logic for knowledge description?

For the sake of transparency we start from the one-sorted case. So, assume X is fixed and we want to understand, how the formulas from $\Phi(X)$ look like. First of all, we consider the classes of equivalent formulas with respect to Lindenbaum-Tarski equivalence. In simple words this means that we identify formulas which take the same values under each interpretation, or, what is the same, they are derivable each from the other. So, we do not care about appearance of the formulas, it is important that they are indistinguishable syntactically and semantically.

The next question is how to survey the representatives of the classes of equivalence. According to definition, these are the formulas constructed on the base of atomic formulas with the help of Boolean operations and quantifiers.

Suppose, for simplicity, that we have only one relational symbol, namely, the symbol of equality. Then, the atomic formulas of our theory are of the form $w \equiv w'$, where w and w' are taken from $W(X)$. Then the formula

$$\forall x_2(\neg(w_1 \equiv w_2) \wedge (w_3 \equiv w_4)) \vee \exists x_1(w_2 \equiv w_3)$$

belongs to $\Phi(X)$, while the formula $(w_1 \vee w_2) \equiv w_3$ is not. If we have a set of relational symbols Ψ, then all atomic formulas $\varphi(w_1, \dots, w_m)$, $\varphi \in \Psi$ play the role of the foundation of $\Phi(X)$.

However, our formulas are, generically, multi-sorted and belong to $\widetilde{\Phi} = (\Phi(X),\ X \in \Gamma)$. Each $\Phi(X)$ contains just defined one-sorted formulas, but they do not exhaust the scope of $\Phi(X)$, since it contains also formulas of the form $u = s_*v$, where $v \in \Phi(Z)$. The meaning of s_* operations is as follows. The substitutions of variables, or more generally, of the words $w_i \to w_j$ play a crucial part in logic (associated with some variety of algebras Θ). The role of operations s_* is to represent these substitutions as operations on the algebra of formulas.

The operations s_* are, evidently, permutable with all Boolean operations. On the other hand, their interplay with quantifiers and relations is regulated by axioms $\diamondsuit$ and can be very sophisticated.

5. Logical Geometry

In this section we deal with the Galois correspondence between sets of formulas T in $\Phi(X)$ and sets of points A in the affine space $Hom(W(X), H)$. We constantly use the term "affine space" for H^n and $Hom(W(X), H)$ because of the following reason. Suppose we take Θ to be the variety of all associative commutative algebras over a fixed field K. Instead of arbitrary sets of formulas T, confine ourselves to sets of equations. Then we arrive to the area of classical algebraic geometry, since a free algebra in Θ is just the algebra $K[X]$ of polynomials over the set of variables $X = \{x_1, \dots, x_n\}$ with coefficients in K. The sets of equations lie in $K[X]$, while the sets of solutions (that is, the algebraic sets) are situated in the affine space K^n. If $K = \mathbb{C}$ or $K = \mathbb{R}$, then we obtain the cases of classical and real geometry, respectively. Although the variety Θ is arbitrary in our setting, we keep for H^n and $Hom(W(X), H)$ the term "affine space", since this is where solutions of equations are located (see, for example, [5]).

The correspondence between sets of equations in the polynomial algebra and algebraic sets in the affine space is one of the basic points for classical algebraic geometry (see [15] for the nice exposition). If Θ is arbitrary, the similar correspondence gives rise to universal algebraic geometry (see [11], [24]). If we consider arbitrary first-order formulas instead of equations, then arising system of notions can be called logical geometry.

5.1. Galois correspondence

Let a homomorphism $Val^X_{(f)} : \Phi(X) \to Hal^X_\Theta(f)$, a point $\mu : W(X) \to H$ and a formula $u \in \Phi(X)$ be given.

Definition 5.1. We say that the point μ satisfies the formula u if

$$\mu \in Val^X_{(f)}(u).$$

So, if $u \in \Phi(X)$, then $Val^X_{(f)}(u)$ is the set of points μ in $Hom(W(X), H)$ satisfying the formula u. It is easy to prove that the standard inductive definition of validity corresponds to this relation between a point and a formula.

For a point $\mu : W(X) \to H$ define $LKer(\mu)$ to be a system of all $u \in \Phi(X)$ satisfied by the point μ. We call $LKer(\mu)$ the *logical kernel* of the point μ. It follows immediately from the definitions:

Proposition 5.2. *A formula $u \in \Phi(X)$ belongs to the logical kernel $LKer(\mu)$ of a point μ if and only if $\mu \in Val^X_{(f)}(u)$.*

Straightforward check shows that $LKer(\mu)$ is a Boolean ultrafilter in $\Phi(X)$. We consider this ultrafilter as LG-type of the point μ:

$$LKer(\mu) = LG^H(\mu).$$

There is a closely related notion of a type $tp^H(\mu)$ in model theory.

$$Ker(Val_{(f)}) = Th(f), \qquad Ker(Val^X_{(f)}) = Th^X(f).$$

Hence, $Th(f)$ is the elementary theory of the model (f). Note that previously (Section 3) the elementary theory was defined as the set of all closed formulas (sentences) valid on (f). Thus, we allow some abuse of notation and slightly enlarge here the elementary theory, without going into the reasons. We call $Th^X(f)$ the *X-theory of the model* (f). We can also present the X-theory of the model (f) as

$$Th^X(f) = \bigcap_{\mu: W(X) \to H} LKer(\mu).$$

Now we are ready to define the Galois correspondence between sets of formulas T in $\Phi(X)$ and sets of points A in the affine space $Hom(W(X), H)$.

Let T be a subset in the set $\Phi(X)$. Assign to the set T a subset $T^L_{(f)} = A$ in the set of points $Hom(W(X), H)$ defined by the rule:

$$A = T^L_{(f)} = \{\mu : W(X) \to H \mid T \subset LKer(\mu)\}.$$

The latter means that the point μ satisfies each formula $u \in T$, i.e., $\mu \in Val^X_{(f)}(u)$. We call a set A of such kind *definable (by T) set*. The definable sets can be also presented as

$$A = T^L_{(f)} = \bigcap_{u \in T} Val^X_{(f)}(u).$$

Let, on the other hand, A be a set of points. Define the corresponding set of formulas by

$$T = A^L_{(f)} = \bigcap_{\mu \in A} LKer(\mu).$$

Here, T is a Boolean filter in the algebra $\Phi(X)$. We call Boolean filters T of the form $T = A^L_{(f)}$, (f)-closed filters. Such filters can be also presented as

$$T = A^L_{(f)} = \{u \in \Phi(X) \mid A \subset Val^X_{(f)}(u)\}.$$

The two-sided correspondence

$$T \Longrightarrow T^L_{(f)} = A, \text{ and } A \Longrightarrow A^L_{(f)} = T,$$

is a Galois correspondence. This means that $A_1 \subset A_2$ implies $A^L_2 \subset A^L_1$ and $T_1 \subset T_2$ implies $T^L_2 \subset T^L_1$. Moreover, $A \subset A^{LL}$ and $T \subset T^{LL}$. Here A^{LL} and T^{LL} are Galois-closed objects, the closures of the sets A and T, respectively.

5.2. Categories $\widetilde{\Phi}_{(f)}$ and $LK_\Theta(f)$

Let us return to the category of knowledge content $LK_\Theta(f)$. The category of knowledge content $LK_\Theta(f)$ is just a category of definable sets whose objects are pairs (X, A), where A is a definable set for the given f.

Define morphisms $[s] : (Y, B) \to (X, A)$ of $LK_\Theta(f)$. Let $s : W(X) \to W(Y)$ be given. If (Y, B) and (X, A) are objects in $LK_\Theta(f)$, $B \subset Hom(W(Y), H)$ and $A \subset Hom(W(X), H)$, then we have a mapping $[s] : B \to A$ determined by the condition $\mu s = \nu \in A$ for $\mu \in B$. Mappings of the form $[s]$ are taken as morphisms of $LK_\Theta(f)$.

Take $T_1 = A^L_{(f)}$ and $T_2 = B^L_{(f)}$ for A and B.

Proposition 5.3. *The inclusion $\mu s \in A$ takes place for each $\mu \in B$ if and only if $s_* u \in T_2$ for every $u \in T_1$.*

Proof. Let $\mu s \in A$. According to the assumption, $T^L_1 = A$. This means that if $u \in T_1$, then every point $\nu \in A$ satisfies the formula u. In particular, the point μs satisfies the formula u. We have $\mu s \in Val^X_{(f)}(u)$, $\mu \in s_* Val^X_{(f)}(u) = Val^Y_{(f)}(s_* u)$ (see the rule for s_* in Section 4.4 and the commutative diagram in Section 4.6). This is valid for each $\mu \in B$. This means that $s_* u \in B^L_{(f)} = T_2$. The proposition is proved in one direction.

Let now $s_* : \Phi(X) \to \Phi(Y)$ and $s_* u \in T_2$ for each $u \in T_1$. We have $T_2{}^L_{(f)} = B$. The point $\mu \in B$ satisfies every formula v from T_2. In particular, each point μ satisfies each formula $s_* u$, $u \in T_1$. We have $\mu \in Val^X_{(f)}(s_* u)$, $\mu s \in Val^X_{(f)}(u)$, $\mu s \in T_1{}^L_{(f)} = A$. $\square$

Now we can define the category of knowledge descriptions $\widetilde{\Phi}_{(f)}$. Its objects have the form (X, T) where T is a closed filter in the algebra of formulas $\Phi(X)$ with respect to model (f) . Define morphisms $s_* : (X, T_1) \to (Y, T_2)$ in $\widetilde{\Phi}_{(f)}$ as restrictions of $s_* : \Phi(X) \to \Phi(Y)$ on T_1. This definition is correct in view of Proposition 5.3.

6. Knowledge base model and isotypic knowledge bases

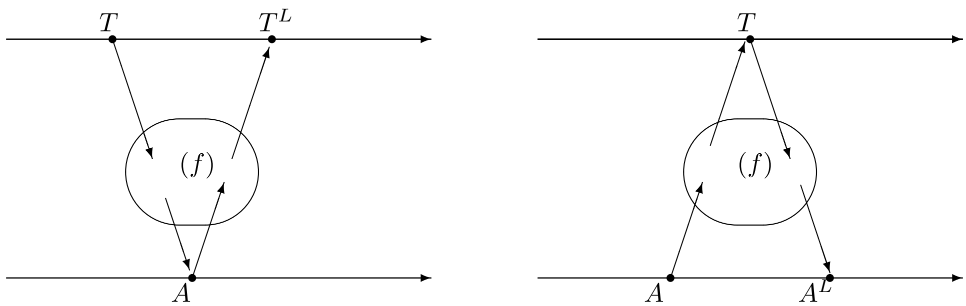
6.1. Informationally equivalent knowledge bases

In fact, a knowledge base defined in 2.3 is represented now as a sort of automata model, where queries are objects of the defined category of descriptions of knowledge $\widetilde{\Phi}_{(f)}$. The requesting information is searched in the subject area presented by a model (H, Ψ, f), where all relations are implemented by the interpretation f in the Cartesian powers of H. The replies have the form of objects of the category of knowledge content $LK_\Theta(f)$. The objects of the latter category are definable sets in an affine space, which means that they are geometric figures, described by sets of first-order sentences, like a unitary disk is described by the sentence $\forall x \forall y (x^2 + y^2 \leq 1)$ in the affine space R^2, where R is the set of real numbers. Of course, the format of the output data is irrelevant and can be chosen to be tables, graphs, etc.

In order to vitalize the whole structure we need to add a dynamical passage which connects queries and replies. Thus, one needs a functor connecting the categories $\widetilde{\Phi}_{(f)}$ and $LK_\Theta(f)$.

Since we have obtained a well-defined Galois correspondence between objects (see Picture 1),

Description (Syntax)



Picture 1 *Content (Semantics)*

the Galois connections between the (f)-closed filters in the syntax and definable sets in semantics can be taken for the definition of a functor between the categories of knowledge description and knowledge content.

Define the knowledge functor

$$Ct_f : \widetilde{\Phi}_{(f)} \to LK_\Theta(f)$$

on objects by

$$Ct_f(T) = T^L_{(f)}.$$

This is the way how the functor Ct_f transforms a description of knowledge to its content. Take now $s : W(X) \to W(Y)$ and the closed filters $T_1 \subset \Phi(X)$ and $T_2 \subset \Phi(Y)$. Then we have $s_* : (X, T_1) \to (Y, T_2)$. Define Ct_f on morphisms by

$$Ct_f(s_*) = [s] : {T_2}^L_{(f)} \to {T_1}^L_{(f)}.$$

So, Ct_f is a contravariant functor and this observation completes Definition 2.3 of a knowledge base.

Now we are in a position to define the notion of knowledge bases informational equivalence in terms of the knowledge base model. Given knowledge bases KB(H_1, Ψ_1, F_1) and KB(H_2, Ψ_2, F_2), consider the diagrams

$$\begin{array}{ccc} \widetilde{\Phi}^1_{(f)} & \xrightarrow{\beta} & \widetilde{\Phi}^2_{(f)} \\ \downarrow{\scriptstyle Ct_f} & & \downarrow{\scriptstyle Ct_{f^\alpha}} \\ LK^1_\Theta(f) & \xrightarrow{\gamma} & LK^2_\Theta(f^\alpha) \end{array} \qquad \begin{array}{ccc} \widetilde{\Phi}^1_{(f)} & \xleftarrow{\beta'} & \widetilde{\Phi}^2_{(f)} \\ \downarrow{\scriptstyle Ct_f} & & \downarrow{\scriptstyle Ct_{f^\alpha}} \\ LK^1_\Theta(f) & \xleftarrow{\gamma^{-1}} & LK^2_\Theta(f^\alpha) \end{array}$$

where $\alpha : F_1 \to F_2$ is a bijection of the sets of interpretations, β, β' are functors of the categories of knowledge description, γ is an isomorphism of the categories of knowledge content.

Definition 6.1. [30] The knowledge bases $KB_1 = KB(H_1, \Psi_1, F_1)$ and $KB_2 = KB(H_2, \Psi_2, F_2)$ are called informationally equivalent if it is possible to choose α, β, β' and γ such that they match the commutative diagrams above.

Existence of $\alpha : F_1 \to F_2$ means that the knowledge bases have the same subject areas and can be represented by the same models. Functors β, β' say that everything that can be asked from one knowledge base can be asked from another. Finally, isomorphism γ gives the possibility to compare contents of knowledge obtained by the first base and by the second one and to conclude that this is the same information.

Suppose we have a query T to the first knowledge base. The functor β transfers it to a query (T^{β}) to the second knowledge base. Functors Ct_f and $Ct_{f^{\alpha}}$ calculate replies T^f and $(T^{\beta})^{f^{\alpha}}$ in the first and second knowledge bases, respectively. Commutativity of both diagrams precisely means that there is one-to-one correspondence between the replies to queries in knowledge bases in question. This means that any information obtained with the help of the first base can be obtained by means of the second one and vice versa. If we identify the queries by the correspondences β and β' and the replies by γ and γ^{-1}, then once again using some abuse of language, we can say that the same queries to two knowledge bases in question return the same replies.

6.2. Elementarily equivalent knowledge bases

From now on the goal is to recognize the informational equivalence of knowledge bases in the sense of Definition 6.1. Our main interest is to find out how the informational equivalence is related to the logical description of knowledge. In this concern, define elementarily equivalent knowledge bases.

Definition 6.2. The knowledge bases $KB_1 = KB(H_1, \Psi, F_1)$ and $KB_2 = KB(H_2, \Psi, F_2)$ are called elementarily equivalent if there exists a bijection $\alpha : F_1 \to F_2$ and for every $f \in F_1$ the models (H_1, Ψ, f) and (H_2, Ψ, f^{α}) are elementarily equivalent.

The elementary equivalence of models ([19], [23], Section 3, Section 5, etc.) means that any first-order formula (sentence) u which is true on one model takes the same value on another one. In other words these models have the same logical description. It is used to say that two models (H_1, Ψ, f_1) and (H_2, Ψ, f_2) are elementarily equivalent if the elementary theories $Th(H_1)$ and $Th(H_2)$ coincide (see Definition 3.6).

So, the question is:

Problem 6.3. Whether the elementary equivalence of knowledge bases implies their informational equivalence.

In more down-to-earth terms Problem 6.3 asks if the notion of elementary equivalence is powerful enough to distinguish the knowledge bases. Theorem 1.2 says that the symmetries of a knowledge base are powerful in this sense and "rigidly" determine the base. It would be desirable to have the same fact with respect to the logical formulas valid on the knowledge base. However, this is not the case:

Proposition 6.4. *There exist two knowledge bases which are elementarily equivalent, but not informationally equivalent.*

Proof. The proof is, in fact, a variation of the Łoś's theorem [18] which, in particular, states that every model $\mathcal{M}$ is elementarily equivalent to its ultrapower $\overline{\mathcal{M}}$. Thus, take $KB_1 = KB(H_1, \Psi, F_1)$ and consider a model $\mathcal{M} = (H_1, \Psi, f_1)$, $f_1 \in F_1$. Let $\overline{\mathcal{M}}$ be an ultrapower of $\mathcal{M}$ corresponding to some Boolean ultrafilter. We can do the same procedure for every $f \in F_1$ and come up with the multi-model (H_2, Ψ, F_2) which can be viewed as an ultrapower of (H_1, Ψ_1, F_1). By Łoś's theorem the associated knowledge bases $KB_1(\mathcal{M})$ and $KB_2(\overline{\mathcal{M}})$ are elementarily equivalent. However, they are not informationally equivalent, since $\mathcal{M}$ and $\overline{\mathcal{M}}$ are not necessarily isomorphic, and, thus, one can show that the isomorphism γ from Definition 6.1 does not exist (for the details of the latter statement see [21], [24]). □

The ultraproduct construction used in Proposition 6.4 assumes a substantial enlargement of the knowledge base. This is not the case for finite algebras, but if, for example, D is a finite-dimensional vector space, then its ultrapower is not, anymore, finite-dimensional. However, one can construct more difficult examples which are free from this drawback.

6.3. Isotypic knowledge bases

Recall (Subsection 5.1) that from a geometric viewpoint, the category of subject content coincides with the category of definable sets. Definable sets are geometric objects defined by sets of first order formulas exactly in the same way as curves, surfaces, etc., are defined by systems of equations. This observation suggests the following definition:

Definition 6.5. The models (H_1, Ψ, f_1) and (H_2, Ψ, f_2) are called LG-equivalent if for any $X \in \Gamma$ and $T \subset \Phi(X)$ we have

$$T^{LL}_{(f_1)} = T^{LL}_{(f_2)}.$$

Referring to Picture 1, one can say that LG-equivalence means that if we start with any description of knowledge T, jump to the knowledge content T^L and then return to knowledge description, we should arrive to the same set of formulas T^{LL} regardless of with respect to which model (H, Ψ, f) the closure $T^{LL}_{(f)}$ is taken. Allowing some abuse of language two models (H_1, Ψ, f_1) and (H_2, Ψ, f_2) are LG-equivalent if the subject area algebras H_1 and H_2 have equal possibilities with respect to solution of logical formulas from T.

Definition 6.6. The models (H_1, Ψ, f_1) and (H_2, Ψ, f_2) are called LG-isotypic, if for each point $\mu : W(X) \to H_1$ there is a point $\nu : W(X) \to H_2$ and, vice versa, for each $\nu : W(X) \to H_2$ we have $\mu : W(X) \to H_1$, such that $LKer(\mu) = LKer(\nu)$.

Remark 6.7. Observe that for each atomic formula $\varphi(w_1, \ldots, w_m)$ of the sort X the point μ satisfies the relation φ under the interpretation f_1 if and only if the point ν satisfies the same φ under the interpretation f_2. This follows from our definition of the logical kernel of a point. Indeed, consider the equality $LKer(\mu) \cap M_X = LKer(\nu) \cap M_X$ along with $LKer(\mu) = LKer(\nu)$. Let an atomic formula $\varphi(w_1, \ldots, w_m)$ belong to the left and right parts of the equality. Since it belongs to the left part, then the tuple $(w_1^\mu, \ldots, w_m^\mu)$ satisfies the relation φ under the interpretation f_1. On the other side, since this formula belongs to the right part, then the tuple $(w_1^\nu, \ldots, w_m^\nu)$ satisfies the same relation φ under the interpretation f_2.

Theorem 6.8. *The models (H_1, Ψ, f_1) and (H_2, Ψ, f_2) are LG-equivalent if and only if they are LG-isotypic.*

Proof. Let the models (H_1, Ψ, f_1) and (H_2, Ψ, f_2) be LG-equivalent. Take a point $\mu : W(X) \to H_1$. Then $\{\mu\}^L_{(f_1)} = LKer(\mu)$. One can show that $\{\mu\}^L_{(f_1)} = \{\nu\}^L_{(f_2)}$ for some point $\nu : W(X) \to H_2$. Then $LKer(\mu) = LKer(\nu)$. This implies isotypeness of the models. The opposite follows from the fact that every (f)-closed filter in $\Phi(X)$ is the intersection of logical kernels. □

Definition 6.9. The knowledge bases $KB_1 = KB(H_1, \Psi, F_1)$ and $KB_2 = KB(H_2, \Psi, F_2)$ are called isotypic if there exists a bijection $\alpha : F_1 \to F_2$ and for every $f \in F_1$ the models (H_1, Ψ, f) and (H_2, Ψ, f^α) are LG-isotypic.

Proposition 6.10. *If the models (H_1, Ψ, f_1) and (H_2, Ψ, f_2) are LG-isotypic, then they are elementarily equivalent.*

Proof. Let a formula u in $\Phi(X)$ and a point $\mu : W(X) \to H$ be given. By definition, $u \in LKer(\mu)$ if and only if $\mu \in Val^X_H(u)$. Furthermore, for any algebra H we have

$$Th^X(H) = \bigcap_{\mu : W(X) \to H} LKer(\mu).$$

So, any Boolean ultrafilter $LKer(\mu)$ contains the X-elementary theory of H. Since for each point $\mu : W(X) \to H_1$ there is a point $\nu : W(X) \to H_2$ and, vice versa, for each $\nu : W(X) \to H_2$ we have $\mu : W(X) \to H_1$, such that $LKer(\mu) = LKer(\nu)$, we have $Th(H_1) = Th(H_2)$.

□

Corollary 6.11. *Isotypic knowledge bases are elementarily equivalent.*

Proof. If the knowledge bases $KB_1 = KB(H_1, \Psi, F_1)$ and $KB_2 = KB(H_2, \Psi, F_2)$ are isotypic, then the models $\mathcal{M}_1 = (H_1, \Psi, f_1)$ and $\mathcal{M}_2 = (H_2, \Psi, f_2)$ are isotypic. By Proposition 6.10, $\mathcal{M}_1$ and $\mathcal{M}_2$ are elementarily equivalent. Hence, the knowledge bases $KB_1 = KB(H_1, \Psi, F_1)$ and $KB_2 = KB(H_2, \Psi, F_2)$ are elementarily equivalent. □

Theorem 6.12. *If the models (H_1, Ψ, f_1) and (H_2, Ψ, f_2) are LG-isotypic, then the categories of definable sets over the given models are isomorphic.*

Proof. We start with some general remarks. Take $s : W(X) \to W(Y)$ and, correspondingly, $s_* : \Phi(X) \to \Phi(Y)$. For $T \subset \Phi(Y)$ we set $s_*T = \{u \in \Phi(X) \,|s_*u \in T\}$. Dually, for $T \subset \Phi(X)$ we have $s^*T = \{s_*u \mid u \in T\}$. Further, s induces $\tilde{s} : Hom(W(Y), H) \to Hom(W(X), H)$. Take $B = s_*A = \tilde{s}^{-1}A$ for $A \subset Hom(W(X), H)$. For $B \subset Hom(W(Y), H)$ we have $s^*B = \{\tilde{s}(\mu) | \mu \in B\}$.

We have the properties:

1. If $T \subset \Phi(X)$, then $(s^*T)^L_H = s_*T^L_H$.
2. If $B \subset Hom(W(Y), H)$, then $(s^*B)^L_H = s_*B^L_H$.
3. If $A \subset Hom(W(X), H)$, then $s^*A^L_H \subset (s_*A)^L_H$.

We view these properties as rules of behavior of definable sets under the moves of affine spaces. The first rule implies that if A is an definable set, then so is s_*A. The second rule says that if T is an H-closed filter in $\Phi(Y)$, then so is s_*T in $\Phi(X)$.

Now we can prove that if (H_1, Ψ, f_1) and (H_2, Ψ, f_2) are LG-isotypic, then the categories $LK_\Theta(f_1)$ and $LK_\Theta(f_2)$ are isomorphic.

By Theorem 6.8 we can suppose that (H_1, Ψ, f_1) and (H_2, Ψ, f_2) are LG-equivalent. We shall define the isomorphism $\mathcal{F} : LK_\Theta(f_1) \to LK_\Theta(f_2)$. Let (X, A) be an object in $LK_\Theta(f_1)$. We set $\mathcal{F}(X, A) = (X, B)$, where $B = (A^L_{(f_1)})^L_{(f_2)}$. Here F determines a bijection on the objects of the category.

Take a morphism $[s]_{f_1} : (X, A_1) \to (X, A_2)$ in $LK_\Theta(f_1)$. We have $s : W(Y) \to W(X)$ and

$$\tilde{s} : Hom(W(X), H_1) \to Hom(W(Y), H_1).$$

If $\nu \in A_1$, then $\tilde{s}(\nu) \in A_2$. Let us check that for the same s we have $\tilde{s}(\mu) \in B_2$ if $\mu \in B_1$. Here $B_1 = ({A_1}^L_{(f_1)})^L_{(f_2)}$, $B_2 = ({A_2}^L_{(f_1)})^L_{(f_2)}$. Our aim is to define $[s]_{f_2} : (X, B_1) \to (X, B_2)$. The embedding $\tilde{s}(\nu) = \nu s \in A_2$ means that $\nu \in \tilde{s}^{-1}A_2 = s_*A_2$. This is equivalent to $A_1 \subset s_*A_2$. We have further

$$({A_1}^L_{(f_1)}) \supset (s_*A_2)^L_{(f_1)} \supset s^*{A_2}^L_{(f_1)}$$

and

$$({A_1}^L_{(f_1)})^L_{(f_2)} = B_1 \subset (s^*{A_2}^L_{(f_1)})^L_{(f_2)} = s_*({A_2}^L_{(f_1)})^L_{(f_2)} = s_*B_2.$$

Thus, $B_1 \subset s_*B_2$ and we have $\tilde{s}(\mu) \in B_2$ for every $\mu \in B_1$. Analogously one can check that if $\tilde{s}(\mu) \in B_2$ for every $\mu \in B_1$, then $\tilde{s}(\nu) \in A_2$ for every $\nu \in A_1$.

Let us show that for $s_1, s_2 : W(Y) \to W(X)$ the equality $\tilde{s}_1(\nu) = \tilde{s}_2(\nu)$ for every $\nu \in A_1 = A$ is equivalent to $\tilde{s}_1(\mu) = \tilde{s}_2(\mu)$ for every $\mu \in B_1 = B$.

Let $\tilde{s}_1(\nu) = \nu s_1 = \nu s_2 = \tilde{s}_2(\nu)$ be given. For every $w \in W(Y)$ we have $\nu s_1(w) = \nu s_2(w)$. Consider the equation $s_1 w \equiv s_2 w$. Then ν is a solution of this equation. This gives $A \subset Val^X_{(f_1)}(s_1 w \equiv s_2 w)$, and $s_1 w \equiv s_2 w \in A^L_{(f_1)}$. Since $A^L_{(f_1)} = B^L_{(f_2)}$, we have $s_1 w \equiv s_2 w \in B^L_{(f_2)}$ and $B \subset Val^X_{(f_2)}(s_1 w \equiv s_2 w)$. This means that for every $\mu \in B$ we have $\mu s_1 w = \mu s_2 w$. This is true for every $w \in W(Y)$, and, hence, $\mu s_1 = \mu s_2$, i.e., $\tilde{s}_1(\mu) = \tilde{s}_2(\mu)$. The converse statement is also true.

It is clear now that $\mathcal{F}([s]_{f_1}) = [s]_{f_2}$ is correctly defined since it does not depend on the choice of the representative. Thus, we get the isomorphism of categories: $\mathcal{F} : LK_\Theta(f_1) \to LK_\Theta(f_2)$.

The theorem is proved. □

Remark 6.13. The notion of isotypic knowledge bases is defined with respect to the case, when the set of relations Ψ is the same for both underlying multi-models. This assumption can be dropped and replaced by bijection between Ψ_1 and Ψ_2.

The following corollary of Theorem 6.12 is the main result of the paper.

Theorem 6.14. *The isotypic knowledge bases are informationally equivalent.*

Proof. Let the knowledge bases $KB_1 = KB(H_1, \Psi, F_1)$ and $KB_2 = KB(H_2, \Psi, F_2)$ be isotypic. Then, there is a bijection $\alpha : F_1 \to F_2$, and we can compare the isotypic models $\mathcal{M}_1 = (H_1, \Psi, f_1)$ and $\mathcal{M}_2 = (H_2, \Psi, f_2)$ with $f_2 = f_1^{\alpha}$. The isomorphic categories of definable sets $LK_{\Theta}(f_1)$ and $LK_{\Theta}(f_2)$ are the categories of knowledge content for KB_1 and KB_2, respectively. By Theorem 6.12 the functor $\mathcal{F}$ provides an isomorphism of these categories. Take $\mathcal{F}$ for the isomorphism γ from the definition of informationally equivalent knowledge bases.

The objects of the category $\widetilde{\Phi}_{(f_1)}$ have the form (X, T), where T is an (f_1)-closed filter. Since $\mathcal{M}_1$ and $\mathcal{M}_2$ are isotypic, every (f_1)-closed filter is (f_2)-closed, and vice versa. So, we have $\beta : \widetilde{\Phi}_{(f_1)} \to \widetilde{\Phi}_{(f_2)}$.

It remains to observe, that by the definition of the functors Ct, $\mathcal{F}$ and β we have

$$Ct_{f_1}\mathcal{F} = \beta Ct_{f_1^{\alpha}}.$$

□

We conclude with some explanatory words regarding Theorem 6.14. There is an essential difference between elementary equivalence and isotypeness of knowledge bases.

Take a knowledge base $KB(\mathcal{M})$, where $\mathcal{M} = (H, \Psi, f)$. Any point μ in the affine space $Hom(W(X), H)$ represents some knowledge from the knowledge area. This knowledge is described by the set of formulas u, such that $\mu \in Val^X_{(f)}(u)$, that is, by the formulas u valid on the point μ. This is exactly the logical kernel $LKer(\mu)$ of μ. Take another point ν. The description of knowledge represented by ν gives us another set of formulas $LKer(\nu)$. The intersection $LKer(\mu) \cap LKer(\nu)$ is the set of formulas which describes a common part of knowledge peculiar to μ and ν. Hence, the elementary theory of the model provides us with the description of the common knowledge which can be extracted from the model. It makes no distinction between the particular points, it characterizes the model as an entire entity. It is clear from positions of the common sense, that such description can be enough to characterize a knowledge base rigidly if and only if we have some additional information about the structure of a knowledge base. These speculations are confirmed by the meaning of the Theorem 6.4, which states that elementarily equivalent knowledge bases can be not informationally equivalent.

Theorem 6.14 says that the latter cannot happen in the case of isotypic knowledge bases and that the property of being isotypic characterizes a knowledge base rigidly. The whole point is that isotypeness of models compares the description of the knowledge peculiar to the points not uniformly, like in the case of elementary equivalence, but takes into account the logical individualities of the points. More precisely, isotypeness of models $\mathcal{M}_1 = (H_1, \Psi, f_1)$ and $\mathcal{M}_2 = (H_1, \Psi, f_2)$ claims that for every knowledge associated with the point μ in the affine space $Hom(W(X), H_1)$, there exists a point ν in $Hom(W(X), H_2)$ such that their logical descriptions coincide, that is $LKer(\mu) = LKer(\nu)$. This type of logical correspondence between knowledge bases turns out to be strong enough to characterize the operation of a knowledge base rigidly.

We finish the discussion with the following conjecture, which seems quite plausible:

Conjecture 6.15. Two finite knowledge bases $KB_1 = KB(H_1, \Psi, F_1)$ and $KB_2 = KB(H_2, \Psi, F_2)$ are informationally equivalent if and only if they are isotypic.

References

[1] A. V. Aho, Y. Sagiv, J. D. Ullman, *Equivalences Among Relational Expressions.* SIAM J. Comput. **8(2)** (1979), 218–246.

[2] P. Atzeni, G. Aussiello, C. Batini, M. Moscarini, *Inclusion and equivalence between relational database schemes.* TCS **19** (1982), 267–285.

[3] K.H. Baik, L. L. Miller, *Topological Approach for Testing Equivalence in Heterogenous Relational Databases.* The Computer Journal **33(1)** (1990), 2–10.

[4] F. Bancillon, *On the completeness of query language for relational databases.* Lecture Notes in CS **64** (1978), 112–123.

[5] G. Baumslag, A. Myasnikov, V. N. Remeslennikov, *Algebraic geometry over groups. I. Algebraic sets and ideal theory*, J. Algebra 219 (1999), 16–79.

[6] C. Beeri, A. Mendelzon, Y. Sagiv, J. Ullman, *Equivalence of relational database schemes.* ACM (1979), 319–329.

[7] E. M. Beniaminov, *Galois theory of complete relational subalgebras of algebras of relations, logical structures, symmetry.* NTI **2** (1980), 17–25.

[8] E. M. Beniaminov, *Algebraic methods in database theory and representation of knowledge* (Russian), Nauchnyj Mir, Moscow, 2003.

[9] J. J. Cannon, D. F. Holt, *Automorphism group computation and isomorphism testing in finite groups.* J. Symbolic Comput. **35(3)** (2003), 241–267.

[10] E. Yu. Daniyarova, A. G. Myasnikov, V. N. Remeslennikov, *Algebraic geometry over algebraic structures. V. The case of arbitrary signature.* Algebra and Logic **51(1)** (2012), 28–40.

[11] E. Yu. Daniyarova, A. G. Myasnikov, V. N. Remeslennikov, *Algebraic geometry over algebraic structures. II. Foundations*, Fundamental and Applied Mathematics, **17(1)** (2011/2012), 65–106.

[12] H. Garsia-Molina, J. D. Ullman, J. Widom, *Database systems. The complete book.* 2d Edition, Pearson Prentice Hall, 2009.

[13] P. Halmos, *Algebraic logic.* Chelsea Publishing Company, 1962.

[14] A. Heuer, *Equivalent schemes in semantic, nested relational, and relational database models.* Lecture Notes in Computer Science **364** (1989), 237–253.

[15] K. Hulek, *Elementary algebraic geometry.* AMS, 2000.

[16] M. Knjazhansky, T. Plotkin, *Knowledge bases over algebraic models: some notes about informational equivalence.* IJKM (International Journal of Knowledge Management) **8(1)** (2012), 22–39.

[17] M. Krasner, *Généralisation abstraite de la théorie de Galois.* Colloque Int. du CNRS (Algèbre et théorie des nombres) **24** (1949), 163–168.

[18] J. Los, *Quelques remarques, theor'emes et probl'emes sur les classes definissables dalg'ebres.* Mathematical Interpretations of Formal Systems, North-Holland (1955), 98–113.

[19] D. Marker, *Model Theory: An Introduction*, Springer Verlag, 2002.

[20] E. Mendelson, *Introduction to mathematical logic*, 2d Edition, Van Nostrand, 1979.

[21] A. Myasnikov, V. Remeslennikov, *Algebraic geometry over groups II, Logical foundations*, J. of Algebra, **234(1)** (2000), 225–276.

[22] E. A. O'Brien, *Isomorphism testing for p-groups.* J. Symbolic Comput. **16(3)** (1993), 305–320.

[23] B. Plotkin, *Universal Algebra, Algebraic Logic and Databases.* Kluwer, 1993.

[24] B. Plotkin, *Algebras with the same algebraic geometry*, Proceedings of the Steklov Institute of Mathematics, MIAN, **242** (2003), 176–207.

[25] B. Plotkin, *Algebraic logic and logical geometry in arbitrary varieties of algebras*, to appear.

[26] B. Plotkin, E. Aladova, E. Plotkin, *Algebraic logic and logically-geometric types in varieties of algebras.* Journal of Algebra and its Applications, **12(2)** (2013).

[27] B. Plotkin, T. Plotkin, *Geometrical aspect of databases and knowledge bases.* Algebra Universalis **46** (2001), 131–161.

[28] B. Plotkin, T. Plotkin, *Categories of elementary sets over algebras and categories of elementary algebraic knowledge.* LNCS, Springer-Verlag, **4800** (2008), 555–570.

[29] T. Plotkin, *Relational databases equivalence problem.* Advances of databases and information systems, Springer (1996), 391–404.

[30] T. Plotkin, M. Knjazhansky, *Symmetries of knowledge bases.* AMAI (Annals of Mathematics and Artificial Intelligence), Special Issue: Applications of Computer Algebra, Springer **64(4)** (2012), 369–383.

[31] J. Rissanen, *On the equivalence of database schemes* in: Proc. ACM. Symp. Princ. Of Database Systems **1** (1982), 22–26.

[32] C. M. Roney-Dougal, *Conjugacy of subgroups of the general linear group.* Experiment. Math. **13(2)** (2004), 151–163.

[33] J. D. Ullman, *Principles of Database and Knowledge-base systems.* Computer Science Press, 1988.

Elena Aladova
Bar Ilan University
Ramat Gan
Israel
e-mail: aladovael mail.ru

Eugene Plotkin
Bar Ilan University
Ramat Gan
Israel
e-mail: plotkin macs.biu.ac.il

Tatjana Plotkin
Bar Ilan University
Ramat Gan
Israel
e-mail: plot macs.biu.ac.il