

# A method for visualization of description logic formulas

NGUYEN NGOC THAN, ILDAR BAYMURATOV, NATALY ZHUKOVA

Department of Informatics and Applied Mathematics  
ITMO University  
Saint Petersburg,  
RUSSIA

**Abstract**— The visualization has proven to be very useful for exploring structures in different application domains. However, there is no any method for visualizing Description Logic formulas, which are widely used in different semantic and artificial intelligence techniques. This paper gives a method for visualization of Description Logic formulas by combining C.S. Pierce's existential graphs with KL-ONE knowledge representation system. In addition, we present a general view of Description Logic, existential graphs, KL-ONE and extended examples of visualization Description Logic formulas with the proposed method.

**Keywords**—Description Logics, existential graph, visualization, KL-ONE.

## 1 Introduction

The visualization has proven to be very useful for exploring structures in different application domains. However, there is no any method for visualizing Description Logic formulas, which are widely used in different semantic and artificial intelligence techniques. In this paper we provide a method for visualization Description Logic formulas. We start by presenting a short overview of Description Logics, its syntax and semantics, in Section 2 and continue with an intuitive introduction to C.S. Pierce's existential graphs as a method for visualizing logical formulas in Section 3. In Section 4 we adopt Pierce's method of visualization to Description Logics to visualize logical relations between concepts, some examples provided. Finally, in Section 5, we describe the developed method. The method is KL-ONE knowledge representation system, augmented with elements of Pearce's existential graphs to represent logical relations. We examine the proposed method with examples.

## 2 Description logics

Description logics (DLs) [1] [2] [3] [4] are a family of knowledge representation languages that can be used to represent the knowledge of an application domain in a structured and formally well-understood way. The name description logics is motivated by the fact that, on the one hand, the important notions of the domain are described by concept descriptions, i.e., expressions that are built from atomic concepts (unary predicates) and atomic roles (binary predicates) using the concept and role constructors provided by the particular DL; on the other

hand, DLs differ from their predecessors, such as semantic networks and frames, in that they are equipped with a formal, logic-based semantics.

Many DLs are more expressive than propositional logic but less expressive than first-order logic. In contrast to the latter, the core reasoning problems for DLs are (usually) decidable, and efficient decision procedures have been designed and implemented for these problems. There are general, spatial, temporal, spatiotemporal, and fuzzy descriptions logics, and each description logic features a different balance between DL expressivity and reasoning complexity by supporting different sets of mathematical constructors.

### 2.1 Syntax

The syntax of a member of the description logic family is characterized by its recursive definition, in which the constructors that can be used to form concept terms are stated. Some constructors are related to logical constructors in first-order logic (FOL) such as intersection or conjunction of concepts, union or disjunction of concepts, negation or complement of concepts, universal restriction and existential restriction. Other constructors have no corresponding construction in FOL including restrictions on roles for example, inverse, transitivity and functionality.

**Notation:** Let  $C$  and  $D$  be concepts,  $a$  and  $b$  be individuals, and  $R$  be a role. If  $a$  is  $R$ -related to  $b$ , then  $b$  is called an  $R$ -successor of  $a$ . Table 1 provides conventional notation for DL.

TABLE 1: CONVENTIONAL NOTATION

Symbol	Description	Example	Read
$\top$	$\top$ is a special concept with every individual as an instance	$\top$	Top
$\perp$	Empty concept	$\perp$	Bottom
$\sqcap$	intersection or conjunction of concepts	$C \sqcap D$	C and D
$\sqcup$	union or disjunction of concepts	$C \sqcup D$	C or D
$\neg$	negation or complement of concepts	$\neg C$	Not C
$\forall$	universal restriction	$\forall R.C$	all R-successors are in C
$\exists$	existential restriction	$\exists R.C$	an R-successor exists in C
$\sqsubseteq$	Concept inclusion	$C \sqsubseteq D$	all C are D
$\equiv$	Concept equivalence	$C \equiv D$	C is equivalent to D
$\doteq$	Concept definition	$C \doteq D$	C is defined to be equal to D
:	Concept assertion	$a : C$	a is a C
:	Role assertion	$(a, b) : R$	a is R-related to b

### The description logic ALC:

The prototypical DL Attributive Concept Language with Complements ALC was introduced by Manfred Schmidt-Schauß and Gert Smolka in 1991 [5], and it is the basis of many more expressive DLs. The following definitions follow the treatment in Baader et al.

Let  $N_C$ ,  $N_R$  and  $N_o$  be (respectively) sets of concept names (also known as atomic concepts), role names and individual names (also known as individuals, nominals or objects). Then the ordered triple  $(N_C, N_R, N_o)$  is the signature.

### Concept:

The set of ALC concepts is the smallest set such that:

- The following are concepts:
  - $\top$  (top is a concept)

- $\perp$  (bottom is a concept)
- Every  $A \in N_c$  (all atomic concepts are concepts)
- If  $C$  and  $D$  are concepts and  $R \in N_r$  then the following are concepts:
  - $C \sqcap D$  (the intersection of two concepts is a concept)
  - $C \sqcup D$  (the union of two concepts is a concept)
  - $\neg C$  (the complement of a concept is a concept)
  - $\forall R.C$  (the universal restriction of a concept by a role is a concept)

- $\exists R.C$  (the existential restriction of a concept by a role is a concept)

*Terminological axioms:*

A *general concept inclusion* (GCI) has the form  $C \sqsubseteq D$  where  $C$  and  $D$  are concepts. Write  $C \equiv D$  when  $C \sqsubseteq D$  and  $D \sqsubseteq C$ .

A TBox is any finite set of GCIs.

*Assertional axioms*

- A concept assertion is a statement of the form  $a : C$  where  $a \in N_0$  and  $C$  is a concept.
- A role assertion is statement of the form  $(a, b) : R$  where  $a, b \in N_0$  and  $R$  is a role.

An ABox is a finite set of assertional axioms.

## 2.2. Semantics

The semantics of description logics are defined by interpreting concepts as sets of individuals and roles as sets of ordered pairs of individuals. Those individuals are typically assumed from a given domain. The semantics of non-atomic concepts and roles is then defined in terms of atomic concepts and roles. This is done by using a recursive definition similar to the syntax.

The description logic ALC

The following definitions follow the treatment in Baader et al.

A terminological interpretation  $I = (\Delta^I, \cdot^I)$  over a signature  $(N_C, N_R, N_0)$  consists of

- a non-empty set  $\Delta^I$  called the domain
- an interpretation function  $\cdot^I$  that maps:
  - every individual  $a$  to an element  $a^I \in \Delta^I$
  - every concept to a subset of  $\Delta^I$
  - every role name to a subset of  $\Delta^I \times \Delta^I$

Such that

- $\top^I = \Delta^I$
- $\perp^I = \emptyset$
- $(C \sqcup D)^I = C^I \cup D^I$  (union means disjunction)
- $(C \sqcap D)^I = C^I \cap D^I$  (intersection means conjunction)
- $(\neg C)^I = \Delta^I \setminus C^I$  (complement means negation)

- $(\forall R.C)^I = \{ x \in \Delta^I \mid \text{for every } y, (x,y) \in R^I \text{ implies } y \in C^I \}$
- $(\exists R.C)^I = \{ x \in \Delta^I \mid \text{there exists } y, (x,y) \in R^I \text{ implies } y \in C^I \}$

Define  $I \models$  (read 'in  $I$  holds') as follows

*TBox* ( $T$ )

- $I \models C \sqsubseteq D$  if and only if  $C^I \subseteq D^I$
- $I \models T$  if and only if  $I \models \phi$  for every  $\phi \in T$

*ABox* ( $A$ )

- $I \models a : C$  if and only if  $a^I \in C^I$
- $I \models (a, b) : R$  if and only if  $(a^I, b^I) \in R^I$
- $I \models A$  if and only if  $I \models \phi$  for every  $\phi \in A$

## 3 Peirce's Graphs as a Method of Visualization

An existential graph is a type of diagrammatic or visual notation for logical expressions, proposed by Charles Sanders Peirce, who wrote on graphical logic as early as 1882, and continued to develop the method until his death in 1914.

Peirce's existential graphs (EGs) [6] [7] [8] [9] are the simplest, most elegant, and easiest-to-learn system of logic ever invented. Yet most logicians have never used them or even seen them. Part of the reason for their neglect is that the algebraic notation by Peirce (1880, 1885) with a change of symbols by Peano (1889) had already become the de facto standard for logic. Another reason is the complexity of Peirce's published explanations, which obscured the simplicity of the graphs with a mass of detail about important, but often distracting semiotic issues. In 1909, however, Peirce wrote Manuscript 514, which contains his clearest tutorial on existential graphs. He presented the syntax, rules of inference, and illustrative examples for first-order logic with equality.

First of all, Peirce defines "graph" as "the propositional expression in the System of Existential Graphs of any possible state of the universe". The existential graphs are then intended by Peirce to be systems of "propositions" or "assertions".

Peirce presents the graphs as three general systems [10] called, respectively, "alpha," "beta," and "gamma." This division corresponds fairly well to his division of

the "algebra of logic" into "non-relative logic," "first-intentional logic of relations," and "second-intentional logic of relations". Alpha nests in beta and gamma. Beta does not nest in gamma, quantified modal logic being more general than put forth by Peirce.

**3.1. Alpha Graphs**

For alpha, we work with a blank page (sheet of assertion), cuts and rules of transformation. Any graph may be enclosed by a simple closed curve called a cut. A cut can be empty. Two graphs on the blank page correspond to a Boolean conjunction, and an oval cut to a Boolean negation (example in Figure 1).

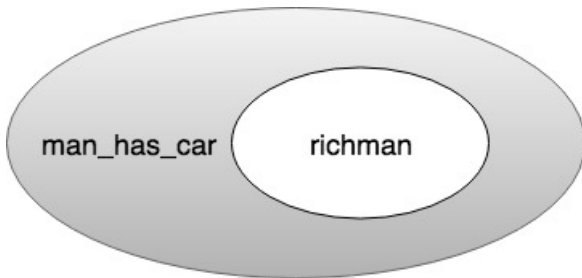


Fig 1: "if man has a car then man is rich"

**3.2. Beta Graphs**

Peirce notated predicates using intuitive English phrases; the standard notation of contemporary logic, capital Latin letters, may also be employed. A dot asserts the existence of some individual in the domain of discourse. Multiple instances of the same object are linked by a line, called the "line of identity". There are no literal variables or quantifiers in the sense of first-order logic. A line of identity connecting two or more predicates can be read as asserting that the predicates share a common variable.

For Beta graphs, the cuts are complemented by a line of identity, which is a diagrammatic analogue of equality, predication, existential quantification (example in Figure 2).

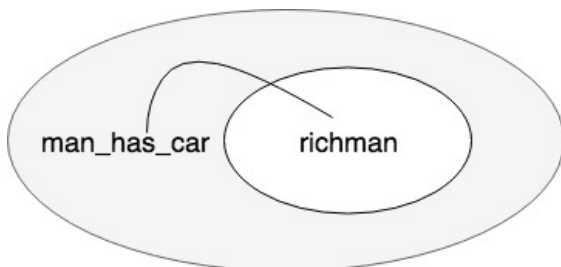


Fig 2: " All men who have car are rich "

**3.3. Gamma graphs**

For gamma, Peirce introduces a dotted oval, allowing the introduction of the modality of the possibility of "possibly not" (example in Figure 3).

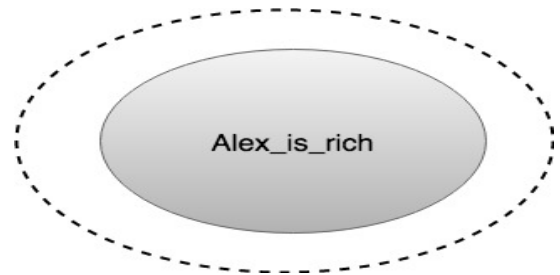


Fig 3: "It is possible that Alex is rich"

**3.4. Syntax**

The existential graphs consist of syntax: sheet of assertion (SA), single letters or phrases written anywhere on the SA, any graph may be enclosed by a simple closed curve called a cut. And in the beta system the following are added: "line of identity" and spots. In the gamma system is added to the syntax of alpha a second kind of simple closed curve, written using a dashed rather than a solid line. Peirce proposed rules for this second style of cut, which can be read as the primitive unary operator of modal logic.


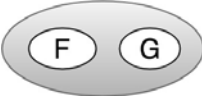

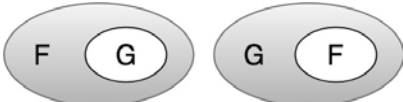
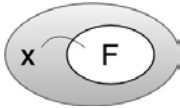

**3.5. Sematic**

The semantics of existential graphs are: The blank page denotes Truth; Letters, phrases, subgraphs, and entire graphs may be True or False; To enclose a subgraph with a cut is equivalent to logical negation or Boolean complementation. Hence an empty cut de-notes False; All subgraphs within a given cut are tacitly conjoined.

**3.6. Peirce's presentation of EG syntax for first-order logic**

The syntax of first-order logic [11] [12] is defined relative to a signature. A signature  $\sigma$  consists of a set of constant symbols, a set of function symbols and a set of predicate symbols. Each function and predicate symbol has an arity  $k > 0$ . We present syntax of first-order logic by Peirce's presentation of EG syntax on table 2.

TABLE 2: PEIRCE'S PRESENTATION OF EG SYNTAX FOR FIRST-ORDER LOGIC WITH EQUALITY

Symbol FOL	Description	EG
$F$	Atomic formula	$F$
$\neg F$	Negation	
$F \wedge G$	Conjunction	$FG$
$F \vee G$	Disjunction	
$F \sqsupset G$	Implication	
$F \sqsubset G$	Equivalence	
$\forall F$	Universal quantification	
$\exists F$	Existential quantification	

## 4 Visualization of Description Logic by Pearce's Graphs


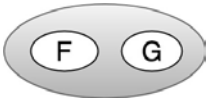
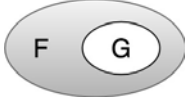

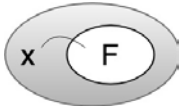

We propose a method for visualization description logic operations according to correspondence between FOL and DL constructions:

TABLE 3: CORRESPONDENCE BETWEEN FOL AND DL OPERATIONS

Description	FOL	DL
Atomic formula	$F$	$F$
Negation	$\neg F$	$\neg F$
Conjunction	$F \wedge G$	$F \sqcap G$
Disjunction	$F \vee G$	$F \sqcup G$
Implication	$F \sqsubseteq G$	$F \sqsubseteq G$
Equivalence	$F \equiv G$	$F \equiv G$
Universal quantification	$\forall F$	$\forall F.C$
Existential quantification	$\exists F$	$\exists F.C$

Summing up, we have following graphs for description logic operations:

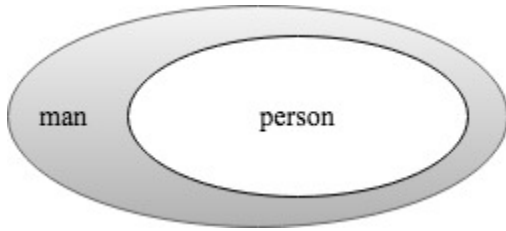
TABLE 4: PEIRCE'S PRESENTATION OF EG SYNTAX FOR DESCRIPTION LOGIC WITH EQUALITY

Symbol DL	Description	EG
$F$	Atomic formula	$F$
$\neg F$	Negation	
$F \sqcap G$	Conjunction	$FG$
$F \sqcup G$	Disjunction	
$F \sqsubseteq G$	Implication	
$F \sqLeftrightarrow G$	Equivalence	
$\forall F.C$	Universal quantification	
$\exists F.C$	Existential quantification	

We present many examples of visualization of description logic formulas by Pierce graphs:

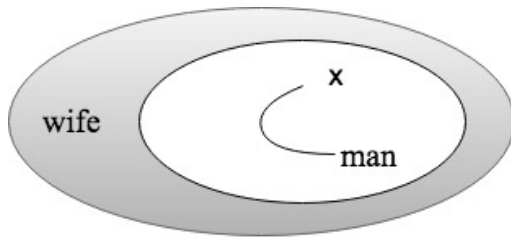
Example 1.  $C1 \sqsubseteq C2$

$\text{man} \sqsubseteq \text{person}$



Example 2.  $C1 \sqsubseteq \exists \text{op1.} C2$

$\text{wife} \sqsubseteq \exists \text{has husband. man}$



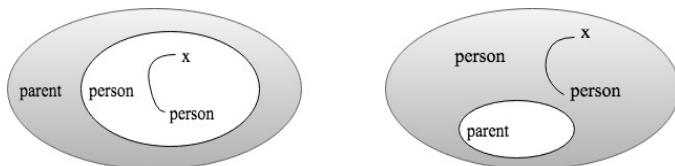
Example 3.  $C1 \sqsubseteq C2 \sqcap (\exists \text{op1.} C3)$

$\text{mother} \sqsubseteq \text{female} \sqcap (\exists \text{has child. person})$



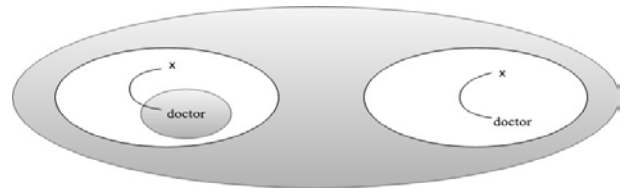
Example 4.  $C1 \sqsubseteq C2 \sqcap (\exists \text{op1.} C2)$

$\text{parent} \sqsubseteq \text{person} \sqcap (\exists \text{has child. person})$



Example 5.  $\forall \text{op1.} C1 \sqsubseteq \exists \text{op1.} C1$

$\forall \text{married. doctor} \sqsubseteq \exists \text{married. doctor}$



## 5 Visualization of Description Logic By Kl-One With Pearce's Graphs

In order to visualize description logic formulas, we propose to augment KL-ONE [13] knowledge representation system with elements of the Pierce's method to visualize logical relation in the way, described in the previous section.

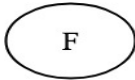
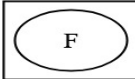
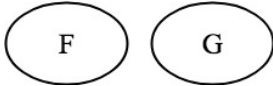
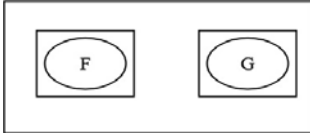
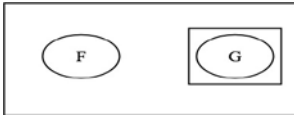
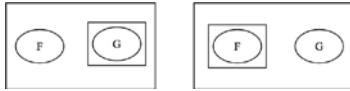
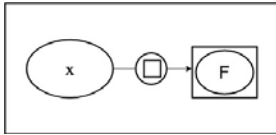
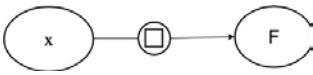
First of all, we are going to describe, what KL-ONE is. KL-ONE is the system for representing knowledge in Artificial Intelligence programs. It has been developed and refined over a long period and has been used in both basic research and implemented knowledge-based systems in a number of places in the AI community. KL-ONE offers a rigorous means of specifying terms (concepts) and basic relationships among them, such as subset/superset, disjointness, exhaustive cover, and relational structure. Concepts are denoted graphically as ovals. Concepts are structured objects whose structure is indicated by named relations (roles) between concepts. Roles are drawn as arcs containing a circle and square. The concepts at the end of the role arcs are said to be value restrictions. In addition, roles have maximum and minimum restrictions on the number of concepts that can be related by the role to the concept at the origin of the arc. Concepts can also have data attached to them, stored as a property list. Finally, the set of concepts is organized into an inheritance hierarchy, through subconcept relations drawn with double-line arrows from the subconcept to the superconcept.

However, there is no way to visualize axioms of DL in the KL-ONE system. We propose to use Pierce's method of visualizing logical operators in KL-ONE. In the context of Pierce's method, KL-ONE concepts are reconsidered as sentences, then a concept positioned on a sheet of assertion means true sentence, two concepts on a sheet of assertion means conjunction of sentences, cut concepts mean negated sentences and so on. The only problem is that both KL-ONE and Pierce's existential graphs use ovals, but in KL-ONE ovals mean concepts, while in Pierce's system they denote negation. Therefore, as any changings in KL-ONE are not welcomed, we propose to represent negations with rectangles.

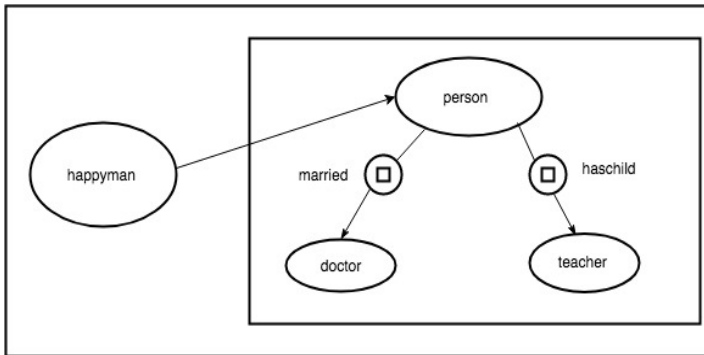
Summing up, we have the following system of visualization:



TABLE 5: VISUALIZATION OF DESCRIPTION LOGIC BY KL-ONE WITH PEARCE'S GRAPHS

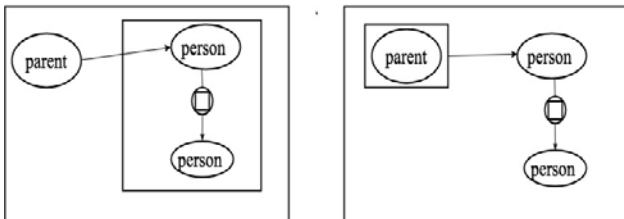
Symbol DL	Description	Visualization
$F$	Atomic formula	
$\neg F$	Negation	
$F \sqcap G$	Conjunction	
$F \sqcup G$	Disjunction	
$F \sqsubseteq G$	Implication	
$F \sqLeftrightarrow G$	Equivalence	
$\forall F.C$	Universal quantification	
$\exists F.C$	Existential quantification	

Example 6.  $\text{Happyman} \sqsubseteq \text{person} \sqcap (\exists \text{married.doctor}) \sqcap (\exists \text{haschild.teacher})$



Intuitively, this scheme shows that there is no situation where a man is happy, but he is not a person, which is married a doctor and has child a teacher. It correlates to the meaning of implication.

Example 7.  $\text{Parent} \sqsubseteq \text{person} \sqcap (\exists \text{haschild.person})$



This scheme shows that there is no situation where somebody is a parent but is not a person who has a child and there is no situation, where a person has a child, but is not a parent. It correlates to the meaning of equivalence.

## 6 Conclusion

In this paper we proposed a method for visualizing description logic formulas. The method is KL-ONE knowledge representation system, augmented with C.S. Peirce's existential graphs for representing logical relations. In order to do it, we represented basics of Description Logic, Peirce's existential graphs and KL-ONE knowledge representation system. Then we adopted Peirce's method of visualization to Description Logic and incorporated it into KL-ONE, slightly modified Peirce's notation. The developed method of

visualization is examined with examples. We believe that application of this method of visualization will be useful in many application domains, where complex structures are subjected to logical analysis, e.g. in systems of biology, electrical circuits or social networks.

## Acknowledgement

The authors would like to thank the anonymous reviewers for their helpful and constructive comments that greatly contributed to improving the final version of the manuscript.

## References

- [1] M. Krötzsch, F. Simancik, and I. Horrocks, "A Description Logic Primer", CoRR, abs/1201.4089, 2012.
- [2] F. Baader, I. Horrocks, U. Sattler Description Logics Handbook of Knowledge Representation, Elsevier, Amsterdam(2007), pp. 135-179.
- [3] Franz Baader. Description logics. In Sergio Tessaris, Enrico Franconi, Thomas Eiter, Claudio Gutierrez, Siegfried Handschuh, Marie-Christine Rousset, and Renate A. Schmidt, editors, Reasoning Web. Semantic Technologies for Information Systems – 5th International Summer School, 2009, volume 5689 of LNCS, pages 1–39. Springer, 2009. Available at <http://lat.inf.tu-dresden.de/research/papers.html>.
- [4] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, second edition, 2007.
- [5] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. Artificial Intelligence, 48(1):1–26, 1991.
- [6] Roberts, D., The Existential Graphs of Charles S. Peirce, The Hague, Paris, Mouton, 1973.
- [7] Peirce, C.S., Collected Papers of Charles Sanders Peirce. Eight volumes, Arthur W. Burks, Charles Hartshorne, and Paul Weiss (eds.). Cambridge, Harvard University Press, 1931–1958.
- [8] Peirce C.S. Manuscripts in the Houghton Library of Harvard University, as identified by Richard Robin // Annotated Catalogue of the Papers of Charles S. Peirce. Amherst, 1967.
- [9] Sowa, J. F. 2011. Peirce's tutorial on existential graphs. Semiotica, 1861-4, 345-394.
- [10] Zeman Jay J. The Graphical Logic of C.S. Peirce: dissertation. Chicago, 1964.
- [11] J. Barwise An introduction to first order logic Handbook of Mathematical Logic, Studies in Logic and the Foundations of Mathematics, 90, North Holland, Amsterdam (1977), pp. 5-46.
- [12] L.S. Cauman, First-Order Logic., 1998.
- [13] Brachman, R. J., and Schmolze, J. (1985). "An overview of the KL-ONE knowledge representation system." Cognitive Science, 9:171-216.