A Variant of Vocabulary and Syntax of Graphical Representation Method of First Order Predicate Logic Formulas

Ian ORLOVSKI, Andrei MIRON, Nicolae PELIN
orlovski@yandex.com, mnior@yandex.com, pelin_nicolae@yahoo.com
University of Applied Sciences of Moldova, Chişinău, Republic of Moldova

A method of logic formulas graphical representation is proposed. The principal idea consist in representing the basic elements of a predicate logic vocabulary as graph edges of different types, differing from previously known types, and directed vertices, connecting this edges, which in whole with the using rules define alternative vocabulary and syntax of proposed graphical language. This allows to essentially distinguish this approach from other well-known practices of logic formulas graphic representations based on the semantic networks. Proposed approach, from our point of view, increases expressiveness in a solving a number of specific problems.
Keywords: *logic, knowledge representation, graph, semantic networks.*

1 Introduction

For graphic representation of logic formulas are used a set of methods based on idea of a semantic networks, which simple variant is used for representation of propositional logic formulas [1, 2]. For increasing expressivity was proposed some advanced variants: extended semantic networks [2], conceptual graphs etc. Primary idea of existing representations results from the semantic networks definition — natural language description of the system translated into the set of the concepts and relations between these concepts. This, for example, allows to formalize natural language phrases in the machine-based form for translate knowledge between different program systems. This mode obviously declaring J. F. Sowa in standard ISO/IEC 24707:2007 for using conceptual graphs.

From our point of view, idea of well-known methods of logic formulas graphic representations, which based on natural language phrases reflection, is not completely correct and can lead to the representation oriented at the end to the natural language. As the goal we need to create a formal language (predicate logic) abstract representation based on graphs. Such method, from our point of view, will allow to create representation, which can be used in system modeling based on the predicate logic language, in the analysis of the developed models, possibly as an instrument of models modification and simplification. This method should be minimal and simple in comparison with known-approaches. In this article we make an attempt to develop graphical representation of first-order predicate logic formulas converted into the conjunctive normal form. In the process of develop was made, from our point of view, successful attempt to release from typical textual properties of logic traditional syntax such as: order number of arguments, parentheses for logic operations priority. In the article will be proposed for discussion vocabulary and syntax of suggested graphical knowledge representation language and its application directions.

From our evaluation of proposed method program realization essentially simplify notation and visual perception of predicate logic formulas. As an increasing of representation expressivity we will research possibilities of more expressive graphical representation based on the second-order predicate logic. Obtained results will extend development abilities in information systems research and modeling.

2 Situation of knowledge representation problem

Today exists a variety of methods and models for knowledge representation. After generalization of different approaches, as it was made in [3], we will come, from the one side,

to the direction of logic representation, and, from the other side, to the one of the structural representations. We will briefly refer to the basic advantages and disadvantages of the two most interesting approaches from opposite points of view: logic modeling and semantic networks modeling.
Logical model permit to get knowledge's formal description and to use this developed description for research with the help of existing mathematical theories. Using first-order predicate logic we obtain full formal system with wide set of theoretical methods of analysis, transformation and inference. Principal disadvantage of this representation – using complexity, including textual or verbal notation, for human, especially in the case of large and complex systems.
From the other side, model of semantic networks is one of the closest to the human thinking, which also have a graphical representation. An abstract graph structure for knowledge representing, from our point of view, is an only minor leaving from the human representation in the form of the semantic networks. Common disadvantage of the mentioned methods consist of practical absence of the formalization. Formalization can be added into the modeling system at the design phase based on formalism laid in the core of the system.
There are different approaches to represent logic using semantic network model, or inverse, such syntax definition on the semantic network (or, in general, on the graph) which allow representing logic formulas.
Our purpose to make a research and comparison of existing graphical representation methods of logic, to discover possible weakness of visualization and to suggest our own representation mode, as far as possible devoid of the discovered problems.

Implicit elements of traditional text representation of the first-order predicate logic formulas

In traditional symbolic notation exists some implicit elements. Though commas and parentheses are defined in vocabulary, their using explicit description usually are omitted as common rules. Addition to this exists implicit elements which are not explicitly defined in syntax – direction of reading and order number of the elements (resulting from the direction of an enumeration).
Commas are used for sharing terms in the list of arguments of a form or other term. Rules of reading arguments list (from left to right) are defining order number (position) or, in other words, identification number of the arguments. In this case parentheses a used to distinguish the object arguments from other elements in the logic formula.
Parentheses are also used for logic operations priority order definition. Decline textual notation, especially parentheses for priority order, we apply graphic variant of the reverse polish notation. About non-linear notation in detail was described in [4]

3 Proposed representation language theory

Proposed graphic representation vocabulary

Vocabulary V of the proposed representation, based on the ordered graph G, is represented by two sets:
V = {N, E}
The set N consist of next different node types: functional constants (F), predicate constants (P), logical operators (O), quantifiers (Q).
The set E consist of: elements connectivity edge (C), negation of the elements connectivity edge (N), term connectivity edge (T)
Below are represented graphical notations of the N and E sets elements:

1. set of different type nodes N:
1.1. functional constant □
1.2. predicate constant □
1.3. logical operators (labeled AND, OR, NOT) □
1.4. quantifiers
1.4.1. missing quantifier □
1.4.2. universal □□
1.4.3. existential □□
2. set of different type edges E:
1.5. element connectivity edge (without label)
1.6. term connectivity edge, for connection between terms and arguments or between

quantifiers and arguments (labeled by argument name for example x, y, z)

Please note:

- quantification node represents free (missing quantifier) or bounded (universal or existential) variable.
- functional and predicate constants are similarly defined in the vocabulary and its meaning distinguished in the syntax, such approach exists, for example, in Prolog language. Another example of such notice is described in [1]

Proposed graph representation syntax

For graph notation we will use Conjunctive Normal Form (CNF). This allow to avoid using conditional operator (implication), which is not commutative, in the nonlinear, reverse polish notation based, representation. Reading order role defined by graph edges direction.

Argument order number lacking in the graph results to using of argument name. The idea of using argument name instead of its order number was adopted from relational SQL language. We can assume that every argument from an object can be noted with the prefix denoting this object (predicate or functional constant). For example the predicate Parent(x, y) can be rewritten in the form of Parent(parentX, parentY). And in this case the logic formula: Parent(x,□y) Woman(x) can be rewritten as Parent(parentX, parentY) □ Woman(womanX) □ =(parentX, womanY). In the graph, arguments always are connected to the concrete objects, so the introduced prefix, that denotes link of the term and object in the text form, can be omitted.

In this form of notation we no more need the implicit elements such as order number of the argument that results from reading direction. In the proposed graph notation reading direction is explicitly represented by the edge direction as in the other forms of graphical representation.

We'll define allowed edge types between different node types, taking into account that edges are directed, so connection can be made only in one-way:

1. Functional constant F can be connected to the other functional constant F or to the predicate constant P by the term connectivity edge;
2. Predicate constant P can be connected to the logical operator by the element connectivity edge or by the outgoing element negation connectivity edge;
3. Logical operator can be connected to the other logical operator by the element connectivity edge or by the outgoing element negation connectivity edge;
4. Variable x (as well as other term, for example functional constant) can be connected to the functional constant F or to the predicate constant P by the term connectivity edge.

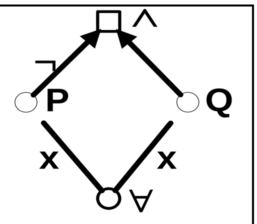
Any other connections are wrong and must be denied.

As the formal definition of the syntax we'll declare a triple R representing relation which have: start node Ns, edge E and end node Ne. Direction of this relation is defined as from the start node to the end node.

Syntax of the represented approach can be defined as the set V_R, where

V_R = {(F, T, F), (F, T, P), (P, C, O), (P, N, O), (O, C, O), (O, N, O), (Q, T, F), (Q, T, P)},

which correspondingly represents textual description given above.

An example of the graphical representation of the simple predicate logic formula□x ¬P(x)□ Q(x)	

Comparison characteristics

Semantic network represent a model for knowledge representation, but it can not be used as modeling language. Therefore we will compare proposed notation with a variants of semantic networks used for predicate logic formulas representation, i.e. extended semantic networks and conceptual graphs.

With the aim to simplify process of a comparison will define next possible comparison

characteristics:
1. well known logic form in the base of the model language
2. limitations of the notation due to selected logic form
3. additional limitations of the notation
4. simplicity and expressivity of notation (positive or negative properties)
5. additional properties

Comparison of the approaches with the defined characteristics:

Extended semantic networks
1. clausal form
2. quantification only by universal quantifier
3. only binary predicates
4. one type of nodes which are distinguished by the textual notation inside the node, exists two types of edges. Exists ability to mark out clauses

Conceptual graphs
1. -
2. -
3. Explicit missing of logic operator OR
4. exists two types of nodes – concepts and relations. Edges as direction of connection and as order number of argument in predicate (relation). Types of terms distinguished by textual notation inside the nodes

Proposed representation
1. conjunctive normal form
2. missing of conditional logic operator (implication)
3. missing of ability to represent single independent formula negation (for example: not(A))
4. 3 nodes types one of them syntactically distinguish into a predicate and a functional form; 3 edge types textual label one of them is used for denote argument
5. pure graphical notation without implicit rules and symbolic meaning

Conclusions

Language proposed in this article unambiguously gives ability of alternative representation of predicate logic formulas. This is undoubtedly useful for its understanding.

It is possible to automatize process of translation between predicate logic and proposed graph language. This is especially important for education systems, including distant learning throw the Internet.

Development and application of graph representation language will expand abilities of knowledge representation in artificial intelligence systems.

References

- Approche logique de l'intelligence artificielle, Bordas, Paris, 1988 (Russian translation an publication - Логический подход к искусственному интеллекту, М.: Мир, 1990)
- Logic and semantic networks, A. Deliyanni, R. A. Kowalski, 1977
- Representation and application of knowledge, H. Wano (Russian translation and publication - Представление и использование знаний.- М.: Мир, 1989)
- Intelligence artificielle, Jean-Louis Lauriere, Paris, 1987