SELF-MODIFICATED PREDICATE NETWORKS

Tatiana Kosovskaya

Abstract: *A model of self-modificated predicate network with cells implementing predicate formulas in the form of elementary conjunction is suggested. Unlike a classical neuron network the proposed model has two blocks: a training block and a recognition block. If a recognition block has a mistake then the control is transferred to a training block. Always after a training block run the configuration of a recognition block is changed. The base of the proposed predicate network is logic-objective approach to AI problems solving and level description of classes.*

Keywords: *artificial intelligence, pattern recognition, predicate calculus, level description of a class.*

ACM Classification Keywords: *I.2.4 Artificial Intelligence - Knowledge Representation Formalisms and Methods-- Predicate logic; I.5.1 Pattern Recognition Models – Deterministic; F.2.2 Nonnumerical Algorithms and Problems – Complexity of proof procedures.*

Introduction

An element of a classical neuron network is an adder of weighted inputs followed by a function mapping the result into the segment *[0, 1]*. The neuron network configuration is fixed and only the adder weights may be changed.

The proposed model of logic-predicate network has two blocks: a training block and a recognition block. An element of every block is an elementary conjunction of an atomic predicate formula or its negation. Configuration of the recognition block is formed after an implementation of the training block and may be changed with its help.

The training block is a "slowly running" block. At the same time the recognition block is a "quickly running" one.

The base of the proposed predicate network is a logic-objective approach to AI problems solving [Kosovskaya, 2007] and level description of classes [Kosovskaya, 2008]. The algorithm of level description of classes is published in [Kosovskaya, 2014; Kosovskaya, 2014a] and uses the notion of partial deduction introduced in [Kosovskaya, 2009].

Main notions of logic-objective approach to a recognition problem

Let an investigated object is represented as a set of its elements $\omega = \{\omega_1, ..., \omega_t\}$. The set of predicates $p_1, ..., p_n$ (every of which is defined on the elements of ω) characterizes properties of these elements and relations between them. Logical description $S(\omega)$ of an object ω is a collection of all true formulas in the form $p_i(\tau)$ or $\neg p_i(\tau)$ (where τ is an ordered subset of ω) describing properties of ω elements and relations between them.

Let the set Ω of all investigated objects is a union of classes $\Omega = \cup_{k=1}^{K} \Omega_k$. Logical description of the class Ω_k is such a formula $A_k(\mathbf{x})$ that if the formula $A_k(\omega)$ is true then $\omega \in \Omega_k$. The class description may be represented as a disjunction of elementary conjunctions of atomic formulas.

Here and below the notation $\mathbf{x}$ is used for an ordered list of the set x. To denote that there exists such a list $\mathbf{x}$ that all values for variables from the list $\mathbf{x}$ are distinct the notation $\exists \mathbf{x}_{\neq} A_k(\mathbf{x})$ is used.

The introduced descriptions allow solving many artificial intelligence problems [Kosovskaya, 2011]. These problems may be formulated as follows. **Identification problem:** to pick out all parts of the object ω which belong to the class Ω_k **Classification problem**: to find all such class numbers k that $\omega \in \Omega_k$. **Analysis problem**: to find and classify all parts τ of the object ω. The solution of these problems is reduced to the proof of logic sequent's $S(\omega) \Rightarrow \exists \mathbf{x}_{\neq} A_k(\mathbf{x})$, $S(\omega) \Rightarrow \vee_{k=1}^{K} A_k(\mathbf{x})$, $S(\omega) \Rightarrow \vee_{k=1}^{K} \exists \mathbf{x}_{\neq} A_k(\mathbf{x})$ respectively and determination of the values for x and k.

The proof of every of these sequent's is based on the proof of the sequent

$$S(\omega) \Rightarrow \exists \mathbf{x}_{\neq} A(\mathbf{x}), \tag{1}$$

where $A(\mathbf{x})$ is an elementary conjunction.

Upper bounds for number of steps for these problems solving are proved in [Kosovskaya, 2010]. These bounds have exponential under the length of the formula $A(\mathbf{x})$ form. More precisely, the power of the exponent is the number of arguments in $A(\mathbf{x})$ for an exhaustive algorithm and the number of atomic formulas for algorithms based on logical derivation in the first order predicate calculus.

It is proved that every of these problems is an NP-complete one. If the sign ∃ is changed by the sign. Then every of these problems is an NP-hard one.

Level description of classes

The notion of level description of classes was introduced in [Kosovskaya, 2008]. Such a description allows essentially to decrease the number of steps for an algorithm solving every of the above formulated problems. This notion is based on the extraction of "frequently" appeared sub-formulas $P_i^1(\mathbf{y}_i^1)$ ($l = 1, \ldots, n_1$) of $A_1(\mathbf{x}_1), \ldots, A_K(\mathbf{x}_K)$ with "small complexity" and changing them in these formulas by atomic formulas $p_i^1(y_i^1)$ defined by the equivalence $p_i^1(y_i^1) \Leftrightarrow P_i^1(\mathbf{y}_i^1)$. New predicates p_i^1 having new first-level arguments y_i^1 for lists $\mathbf{y}_i^1$ of initial variables are called first-level predicates. The formula $A_k^1(\mathbf{x}_k^1)$ is received from $A_k(\mathbf{x}_k)$ by means of a substitution of $p_i^1(y_i^1)$ instead of $P_i^1(\mathbf{y}_i^1)$.

Repeat the above described procedure with all formulas $A_k^1(\mathbf{x}_k^1)$. After L repetitions an L-level description in the following form is received.

$$A_k^L(\mathbf{x}_k^L)$$

$$p_1^1(y_1^1) \Leftrightarrow P_1^1(\mathbf{y}_1^1)$$

$$\cdots$$

$$p_{n1}^1(y_{n1}^1) \Leftrightarrow P_{n1}^1(\mathbf{y}_{n1}^1)$$

$$\cdots$$

$$p_i^l(y_i^l) \Leftrightarrow P_i^l(\mathbf{y}_i^l)$$

$$\cdots$$

$$p_{nL}^L(y_{nL}^L) \Leftrightarrow P_{nL}^L(\mathbf{y}_{nL}^L).$$

The solving of the problem in the form (1) with the use of the level description of classes is decomposed on the sequential ($l = 1, \ldots, L$) implementation of the following actions.

- For every i ($i = 1, \ldots, n_l$) check $S^{\,l-1}(\omega) \Rightarrow \exists \mathbf{y}_{i\neq}^l\ P_i^l(\mathbf{y}_i^l)$ and find all lists τ_i^l of previous levels constants for the values of the variable list $\mathbf{y}_i^l$ such that $S^{\,l-1}(\omega) \Rightarrow P_i^l(\tau_i^l)$;
- Introduce new l-level atomic formulas $p_i^l(y_i^l)$ defined by the equalities $p_i^l(y_i^l) \Leftrightarrow P_i^l(\mathbf{y}_i^l)$ with new l-level variables;
- Substitute $p_i^l(y_i^l)$ instead of $P_i^l(\mathbf{y}_i^l)$ into $A_k^{l-1}(\mathbf{y}_k^l)$ and obtain $A_k^{l-1}(\mathbf{y}_k^l)$;
- Add all constant atomic l-level formulas in the form $p_i^l(\tau_i^l)$ (τ_i^l were received at the first step) to $S^{l-1}(\omega)$ and obtain $S^l(\omega)$. Here τ_i^l are new l-level constants for the lists of (l – 1)-level constants.

At last check $S^L(\omega) \Rightarrow \exists \mathbf{y}_{k\neq}^L A_k^L(\mathbf{y}_k^L)$.

An approach to the extraction of common sub-formulas was described in [Kosovskaya, 2014; Kosovskaya, 2014a]. It is based on the notion of partial deduction.

Partial deduction

The notion of partial deduction was introduced by the author in [Kosovskaya, 2009] to recognize objects with incomplete information. During the process of partial deduction instead of the proof of (1) we search such a maximal (up to the names of variables) sub-formula $A'(\mathbf{x}')$ of the formula $A(\mathbf{x})$ that $S(\omega) \Rightarrow \exists \mathbf{x}'_{\neq} A'(\mathbf{x}')$.

Let a and a' be the numbers of atomic formulas in $A(\mathbf{x})$ and $A'(\mathbf{x}')$ respectively, m and m' be the numbers of objective variables in $A(\mathbf{x})$ and $A'(\mathbf{x}')$ respectively. Parameters q and r are defined as $q = a'/a$ and $r = m'/m$. In such a case sub-formula $A'(\mathbf{x}')$ is called a (q,r)-fragment of the formula $A(\mathbf{x})$.

Sub-formula $A'(\mathbf{x}')$ is called a maximal (up to the names of variables) sub-formula of the formula $A(\mathbf{x})$ if it is its (q,r)-fragment with the maximal value of the parameter q. Hence, for the (q,r)-fragment $A'(\mathbf{x}')$ the sequence

$$S(\omega) \Rightarrow \exists \mathbf{x}_{\neq} \ A'(\mathbf{x}')$$

is valid and for every other (q'',r'')-fragment $A''(\mathbf{x}'')$ of the formula $A(\mathbf{x})$ with $q'' < q$ this sequence is wrong.

The other definition of parameters q and r is possible. Let predicate symbols setting object characteristics have weights w_i $(i = 1, \dots, n)$ and objective variables in $A'(\mathbf{x}')$ and in $A(\mathbf{x})$ have weights v_j $(j = 1, \dots, m)$ and v'_j $(j = 1, \dots, m')$ respectively.

The value of parameter q_w is defined as quotient of the sum of weights for all predicate symbol occurrences in $A'(\mathbf{x}')$ to the sum of weights for all predicate symbols occurrences in $A(\mathbf{x})$. The value of parameter r_v is defined as quotient of the sum of weights for all objective variables in $A'(\mathbf{x}')$ to the sum of weights for all objective variables in $A(\mathbf{x})$. Parameter q (as well as parameter q_w) characterizes the in formativeness of (q,r)-fragment containing only r-th (r_v-th respectively) part of all variables.

To point that we check not the sequence $\Phi \Rightarrow \Psi$ for some formulas Φ *and* Ψ but their partial sequence the notation $A \Rightarrow_P B$ will be used.

Algorithm of level description construction

The notion of partial deduction allows to develop an algorithm for an extraction of a maximal (up to the names of variables) sub-formula of two elementary conjunctions [Kosovskaya, 2014; Kosovskaya, 2014a]. Let $A_i(\mathbf{x}_i)$ and $A_j(\mathbf{x}_j)$ be two elementary conjunctions of predicate formulas with the lists of variables $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively. Checking of partial deduction for $A_i(\mathbf{x}_i) \Rightarrow_P \exists \mathbf{x}_{j\neq}\ A_j(\mathbf{x}_j)$ (or for $A_j(\mathbf{x}_j) \Rightarrow_P \exists \mathbf{x}_{l\neq}\ A_i(\mathbf{x}_l)$) gives their maximal (up to the names of variables) common sub-formula $Q^1_{i,j}(\mathbf{x}_{i,j})$ such that $A_i(\mathbf{x}_i) \Rightarrow \exists \mathbf{x}_{i,j\neq}\ Q^1_{i,j}(\mathbf{x}_{i,j})$ and $A_j(\mathbf{x}_j) \Rightarrow \exists \mathbf{x}_{j,i\neq}\ Q^1_{j,i}(\mathbf{x}_{j,i})$. It may be proved that the formulas $Q^1_{i,j}(\mathbf{x}_{i,j})$ and $Q^1_{j,i}(\mathbf{x}_{j,i})$ coincide up to the names of variables.

The list of variables $\mathbf{x}_{i,j}$ may does not be contained in the list $\mathbf{x}_j$. And the list of variables $\mathbf{x}_{j,i}$ may does not be contained in the list $\mathbf{x}_i$. Nevertheless if we use an exhaustive algorithm or an algorithm based on logical derivation in the first order predicate calculus common unifiers (i.e. such substitutions of variables from the lists $\mathbf{x}_i$ (and $\mathbf{x}_j$ respectively) into the list of variables $\mathbf{x}_{i,j}$ (and $\mathbf{x}_i$ respectively) that after applying of these substitutions to $Q^1_{i,j}(\mathbf{x}_{i,j})$ (to $Q^1_{j,i}(\mathbf{x}_{j,i})$) it becomes a sub-formula of $A_i(\mathbf{x}_i)$ ($A_j(\mathbf{x}_j)$ respectively)) will be constructed. That's why the formulas $Q^1_{i,j}(\mathbf{x}_{i,j})$ and $Q^1_{j,i}(\mathbf{x}_{j,i})$ would not be distinguished.

Algorithm of level description

1. For every pair of elementary conjunctions in the class description by means of partial deduction checking for $A_i(\mathbf{x}_i) \Rightarrow_P \exists \mathbf{x}_{j\neq}\ A_j(\mathbf{x}_j)$ extract their maximal (up to the names of variables) common sub-formula $Q^1_{i,j}(\mathbf{x}^1_{i,j})$. At the same time find common unifiers for $A_i(\mathbf{x}_i)$ and $A_j(\mathbf{x}_j)$ with $Q^1_{i,j}(\mathbf{x}^1_{i,j})$.
2. Repeat the process of maximal (up to the names of variables) common sub-formulas extraction for every pair $Q^l_i(\mathbf{x}^l_i)$ and $Q^l_j(\mathbf{x}^l_j)$ ($l = 1, ..., L-1$), receive $Q^{l+1}_{i,j}(\mathbf{x}^{l+1}_{i,j})$ and common unifiers for $Q^{l+1}_{i,j}(\mathbf{x}^{l+1}_{i,j})$ with $Q^l_i(\mathbf{x}^l_i)$ and $Q^l_j(\mathbf{x}^l_j)$. The process will stop because the lengths of the extracted formulas decrease.
3. Among all extracted sub-formulas select minimal (according to the number of variables for the further exhaustive algorithm use and according to the number of atomic formulas for the further logic algorithm use). Denote the chosen sub-formulas by means of $P_i^1(\mathbf{y}_i^1)$ ($i = 1, \dots, n_1$) and introduce 1-st level predicates p_i^1 with 1-st level variables y_i^1 for the lists of initial variables defined by the equivalences $p_i^1(y_i^1) \Leftrightarrow P_i^1(\mathbf{y}_i^1)$.
4. Formulas $P_i^{l+1}(\mathbf{y}_i^{l+1})$ ($i = 1, \dots, n_{l+1}$, $l = 1, \dots, L-1$) are chosen among the previously extracted formulas with taking in account that all sub-formulas $P_i^l(\mathbf{y}_i^l)$ are replaced by new atomic formulas $p_i^l(y_i^l)$.

Note that during the training block run it is possible to assign weights to the level predicate and level variables. It may be done, for example, in dependence of the frequency of their appearances in the training set.

Formation of logic-predicate network

A training set of objects is given to form an initial variant of the network training block. Replace every constant ω_j in S(ω) by a variable x_j ($j = 1, \ldots, t$) and substitute the sign & between the atomic formulas. Initial description of classes consists of these elementary conjunctions.

Then the **Algorithm of level description** is implemented to the received set of elementary conjunctions. The training block is formed.

Formulas $P_i^l(\mathbf{y}_i^l)$ ($i = 1, \ldots, n_l$, $l = 1, \ldots, L$) obtained in the training block (together with unifiers) are the contents of the cells forming recognition block. This block runs as it was described in the section **Level description of classes**.

If after the recognition block run an object is not recognized nor has wrong identification then it is possible to train anew the network. The description of the "wrong" object must be added to the input set of the training block. The training block extracts common sub-formulas of this description and previously received formulas. Some sub-formulas in the level description would be varied. Then the recognition block is reconstructed.

The scheme of the network is presented on the Figure 1.

It is possible to modify the process of recognition. For example, it is possible to calculate a distance between the parts of a recognized object and sub-formulas defining a level predicate. Then the nearest sub-formula is taken for the further recognition of the object. The calculation of such a distance is presented in [Kosovskaya, 2012]. But in such a case the recognition block would run essentially slower. Moreover, if we use the procedure of "the nearest neighbor" then we can only suppose that some part (r-th part) of the object is contained in the object of the chosen class with some (q-th) degree of certainty.

A model example of predicate network construction

Given a training set of the class of "boxes" of contour images represented on the Figure 2.

Every object is a set of nodes in its contour image. Descriptions of objects are done in the terms of initial predicates V and L defining relations between the nodes. The descriptions of these predicates are presented on the Figure 3.

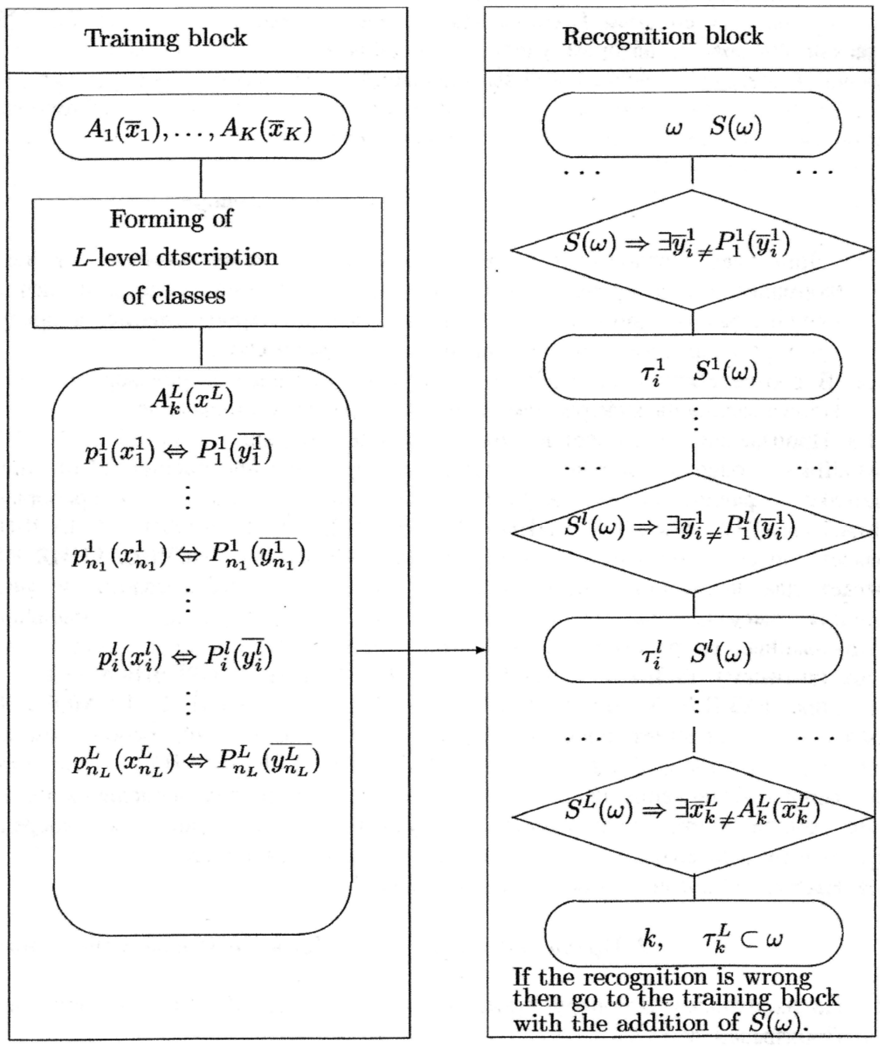
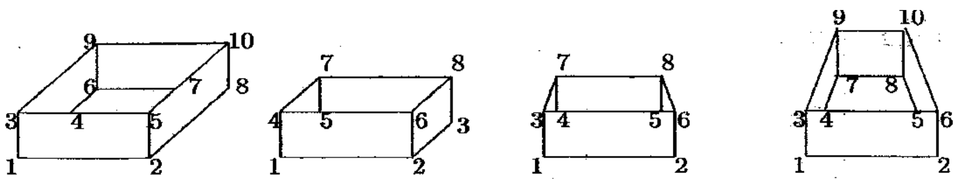
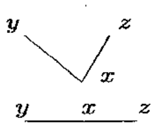


Figure 1. Scheme of the network



a b c d

Figure 2. Training set of a class



$V(x,y,z) \Leftrightarrow$ "$\angle yxz < \pi$"

L(x,y,z) $\Leftrightarrow$ *"x belongs the segment (y,z)"*

Figure 3. Initial predicates

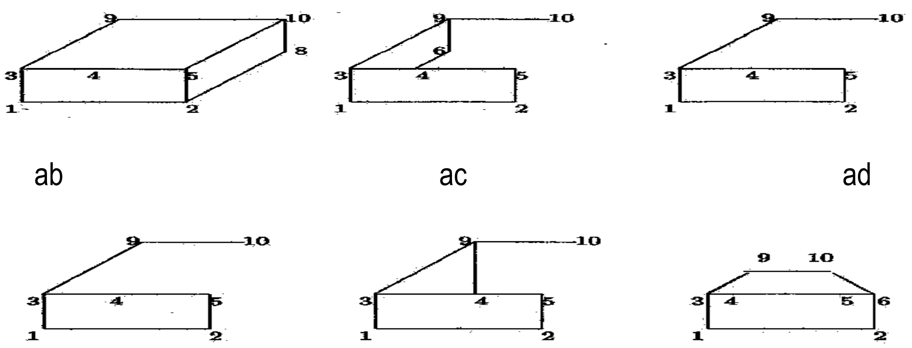
The description of class of "boxes" made according the given training set contains 4 elementary conjunctions. Every such conjunction consists of all atomic formulas which are valid for the corresponding image and in which the node name *i* is changed by the variable x_i.

For example, the elementary conjunction

$A_2(x_1, \dots, x_8) =$

$V(x_1,x_4,x_2)$ & $V(x_2,x_1,x_6)$ & $V(x_2,x_6,x_3)$ & $V(x_2,x_1,x_3)$ & $V(x_3,x_2,x_8)$ & $V(x_4,x_5,x_1)$ & $V(x_4,x_6,x_1)$ & $V(x_4,x_7,x_5)$ & $V(x_4,x_7,x_6)$ & $V(x_4,x_7,x_1)$ & $V(x_5,x_4,x_7)$ & $V(x_5,x_7,x_6)$ & $V(x_6,x_2,x_5)$ & $V(x_6,x_2,x_4)$ & $V(x_6,x_5,x_8)$ & $V(x_6,x_4,x_8)$ & $V(x_6,x_8,x_2)$ & $V(x_7,x_5,x_4)$ & $V(x_7,x_8,x_5)$ & $V(x_7,x_8,x_4)$ & $V(x_8,x_3,x_6)$ & $V(x_8,x_6,x_7)$ & $V(x_8,x_3,x_7)$ & $L(x_5,x_4,x_6)$

corresponds to the image *b* on the Figure 2 pairwise partial deduction of these elementary conjunctions allows to extract common sub-formulas corresponding to the images represented on the Figure 4.



ab ac ad

bc bd cd

Figure 4. Images corresponding to extraction of common sub-formulas

For example, the elementary conjunction

$Q^1_{2,4}(x_2,x_3,x_4,x_5,x_9,x_{10}) =$

$V(x_1,x_3,x_2)$ & $V(x_2,x_1,x_5)$ & $V(x_3,x_4,x_1)$ & $V(x_3,x_5,x_1)$ & $V(x_3,x_9,x_4)$ & $V(x_3,x_9,x_5)$ &

$V(x_3,x_9,x_1)$ & $V(x_5,x_2,x_4)$ & $V(x_5,x_2,x_3)$ & $V(x_9,x_{10},x_4)$ & $V(x_9,x_4,x_3)$ & $L(x_4,x_3,x_5)$

corresponds to the image *bd* on the Figure 4.

The following extraction by means of pairwise partial deduction between common sub-formulas corresponding images *ab, ac, ad, bc, bd, cd* gives a sub-formula corresponding to the image represented on Figure 5.

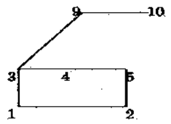


Figure 5. Image corresponding to the second extraction of common sub-formulas

Elementary conjunction

$Q^2(x_1,x_2,x_3,x_4,x_5,x_9,x_{10}) =$

$V(x_1,x_3,x_2)$ & $V(x_2,x_1,x_5)$ & $V(x_3,x_4,x_1)$ & $V(x_3,x_5,x_1)$ & $V(x_3,x_9,x_4)$ & $V(x_3,x_9,x_5)$ &

$V(x_3,x_9,x_1)$ & $V(x_5,x_2,x_4)$ & $V(x_5,x_2,x_3)$ & $V(x_9,x_{10},x_3)$ & $L(x_4,x_3,x_5)$

corresponds to the image on the Figure 5. The process of maximal common sub-formulas extraction stops.

The formula $Q^2(x_1,x_2,x_3,x_4,x_5,x_9,x_{10})$ is a minimal one and is taken for $P^1(x_1,x_2,x_3,x_4,x_5,x_9,x_{10})$. It defines a first-level predicate $p^1(x^1)$. Here x^1 is a new first-level variable for the list of initial variables $(x_1,x_2,x_3,x_4,x_5,x_9,x_{10})$. Taking in account common unificators (defined while extraction of common sub-formulas) variable x^1 will be used for the lists $(x_1,x_2,x_3,x_4,x_5,x_9,x_{10})$, $(x_1,x_2,x_4,x_5,x_6,x_7,x_8)$, $(x_1,x_2,x_3,x_4,x_6,x_7,x_8)$, $(x_1,x_2,x_3,x_4,x_6,x_9,x_{10})$ respectively in four formulas of the initial description.

After substitution $p^1(x^1)$ instead of every sub-formula in the form $P^1(\mathbf{y})$ into $A_i(\mathbf{x}_i)$ the two-level description is obtained. For example, the elementary conjunction corresponding to the image *b* on the Figure 2 has the form

$A_2{}^1(x^1, x_1, \ldots, x_8) =$

$p^1(x^1)$ & $V(x_2,x_6,x_3)$ & $V(x_2,x_1,x_3)$ & $V(x_3,x_2,x_8)$ & $V(x_5,x_4,x_7)$ & $V(x_5,x_7,x_6)$ & $V(x_6,x_5,x_8)$ & $V(x_6,x_4,x_8)$ &

$V(x_6,x_8,x_2)$ & $V(x_7,x_5,x_4)$ & $V(x_7,x_8,x_5)$ & $V(x_7,x_8,x_4)$ & $V(x_8,x_3,x_6)$ & $V(x_8,x_6,x_7)$ & $V(x_8,x_3,x_7)$.

Four second-level predicates are defined by elementary conjunctions corresponding to the images *ab, ac, bd, cd* on the Figure 3. This elementary conjunction contains the first-level predicate p^1 and the first-level variable x^1.

For example, the elementary conjunction corresponding to the image *b* on the Figure 4 has the form

$$P_1^2(x^1, x_1,x_2,x_3,x_4,x_5,x_8,x_9,x_{10}) =$$
$$p^1(x^1) \,\&\, V(x_2,x_5,x_8) \,\&V(x_2,x_1,x_8) \,\&\, V(x_5,x_3,x_{10}) \,\&\, V(x_5,x_4,x_{10}) \,\&\, V(x_5,x_{10},x_2) \,\&$$
$$V(x_8,x_2,x_{10})\&\, V(x_9,x_{10},x_3)\&\, V(x_{10},x_8,x_5)\&\, V(x_{10},x_8,x_9)\&\, V(x_{10},x_5,x_9).$$

It defines a second-level predicate p_1^2 and second-level variable x_1^2. The second-level variable x_1^2 is a variable for the list of initial variables and the first-level variable (x^1, $x_1,x_2,x_3,x_4,x_5,x_8,x_9,x_{10}$) which corresponds to the list of initial variables ($x_1,x_2,x_3,x_4,x_5,x_8,x_9,x_{10}$).

After substitution of $p_i^2(x^1)$ instead of every sub-formula in the form $P_i^2(\mathbf{y})$ $(I = 1, 2, 3, 4)$ into $A_k^1(\mathbf{x}_k)$ the three-level description is obtained. For example, the elementary conjunction corresponding to the image *a* on the Figure 2 with the use of the formula $P_1^2(x^1,x_1,x_2,x_3,x_4,x_5,x_8,x_9,x_{10})$ describing the image *ab* on the Figure 4 has the form

$$A_{1,1}^2(x_1^2,x_3,x_4,x_5,x_6,x_7,x_9,x_{10}) =$$
$$p_1^2(x_1^2) \,\&\, V(x_4,x_3,x_6) \,\&\, V(x_4,x_6,x_5) \,\&\, V(x_6,x_4,x_9) \,\&\, V(x_6,x_9,x_7) \,\&\, V(x_9,x_7,x_4) \,\&$$
$$V(x_7,x_5,x_6) \,\&\, V(x_7,x_6,x_{10}) \,\&\, V(x_9,x_6,x_3) \,\&\, V(x_9,x_{10},x_6) \,\&\, V(x_9,x_{10},x_3) \,\&\, L(x_7,x_5,x_{10}).$$

But the elementary conjunction corresponding to the image *a* on the Figure 2 with the use of the formula $P_1^2(x^1,x_1,x_2,x_3,x_4,x_5,x_8,x_9,x_{10})$ describing the image *ac* on the Figure 4 has the form

$$A_{1,2}^2(x_2^2,x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10},) =$$
$$p_2^2(x_2^2) \,\&\, V(x_2,x_5,x_8) \,\&\, V(x_2,x_1,x_8) \,\&\, V(x_5,x_3,x_7) \,\&\, V(x_5,x_3,x_{10}) \,\&\, V(x_5,x_4,x_7) \,\&\, V(x_5,x_4,x_{10}) \,\&$$
$$V(x_6,x_7,x_4) \,\&\, V(x_6,x_9,x_7) \,\&\, V(x_7,x_5,x_6) \,\&\, V(x_7,x_6,x_{10}) \,\&\, V(x_8,x_2,x_{10}) \,\&\, V(x_{10},x_5,x_9) \,\&$$
$$V(x_{10},x_7,x_9) \,\&\, V(x_{10},x_8,x_5) \,\&\, V(x_{10},x_8,x_7) \,\&\, V(x_{10},x_8,x_9) \,\&\, L(x_7,x_5,x_{10}).$$

Here the second-level variable x_2^2 is a variable for the list of first-level and initial variables (x^1, $x_1,x_2,x_3,x_4,x_5,x_6,x_9, x_{10}$) which corresponds to the list of initial variables ($x_1,x_2,x_3,x_4,x_5,x_6,x_9,x_{10}$).

The training block run stops.

Given a new image represented on the Figure 6 for recognition, the network would not recognize it because the first-level predicate is not valid.

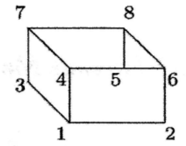


Figure 6. Control image

Add the description of this control image to the input data of the training block. The extraction of common sub-formulas for this description and the formula defining the first-level predicate gives a formula corresponding to the image represented on the Figure 7.

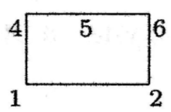


Figure 7. Image corresponding to the new first-level predicate

New second-level predicates correspond to three images represented on the Figure 8.

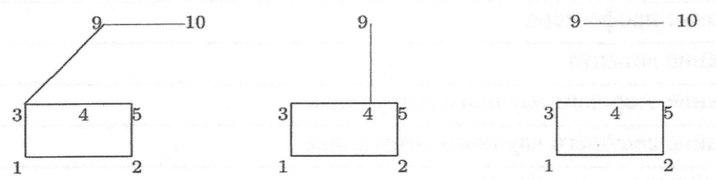


Figure 8. Images corresponding to three new second-level predicates.

The set of the third-level predicates coincides with the set of previous second-level predicates. So, the recognition block is constructed anew and represents four-level description of the class.

Conclusion

An approach to the construction of self-modificated predicate network is presented in the paper.

Now the main problem while constructing such a network is a detailing of unifiers storage and transfer. These unifiers allow to substitute new atomic formulas instead of the extracted (up to the names of variables) sub-formulas.

The open question is "what extracted formula must be changed by an atomic one if it may be done in different ways?" To answer this question complexity investigation must be done.

While extracting a sub-formula it may happen that it contains several variables of a lower (not initial) level. In such a case the sub-formula defines a relation between parts of an object. When we must regard these parts as informative pair or a new informative part?

Very interesting is the possibility of a weighted predicates and variables use. Probably these weights must vary in dependence of "valid" or "wrong" recognition as it is usual for traditional neuron networks.

Acknowledgements

The paper is published with financial support of RFBR grant 14-08-01276.

Bibliography

[Kosovskaya, 2007] T. M. Kosovskaya, "Proofs of the number of steps bounds for solving of some pattern recognition problems with logical description", In: Vestnik of St.Petersburg University. Ser. 1, 2007, No. 4, pp. 82 – 90 (In Russian)

[Kosovskaya, 2008] T. M. Kosovskaya, "Level descriptions of classes for decreasing step number of pattern recognition problem solving described by predicate calculus formulas", In: Vestnik of St.Petersburg University. Ser. 10, 2008, No. 1, pp. 64 – 72 (In Russian)

[Kosovskaya, 2009] T. M. Kosovskaya. "Partial deduction of a predicate formula as an instrument for recognition of an object with incomplete description", In: Vestnik of St. Petersburg University. Ser. 10, 2009, No. 1, pp. 77 – 87 (In Russian)

[Kosovskaya, 2010] T. M. Kosovskaya, "Some artificial intelligence problems permitting formalization by means of predicate calculus language and upper bounds of their solution steps" // SPIIRAS Proceedings, 2010, No 14.

[Kosovskaya, 2011] T. Kosovskaya, "Discrete Artificial Intelligence Problems and Number of Steps of their Solution" // In: International Journal "Information Theories and Applications", Vol. 18, Number 1, 2011, pp. 93 – 99

[Kosovskaya, 2012] Kosovskaya T., “Distance between objects described bt predicate formulas” // In: International Book Series. Informational Science and Computing. Book 25. Mathematics of Distances and Applications (Michel Deza, Michel Petitjean, Krasimir Markov(eds)), ITHEA-Publisher, Sofia, Bulgaria, 2012, pp. 153 – 159

[Kosovskaya, 2014] Kosovskaya T., “Construction of Class Level Description for Efficient Recognition of a Complex Object” // International Journal “Information Content and Processing”, Vol. 1, No 1, 2014, pp. 92 – 99

[Kosovskaya, 2014a] Kosovskaya T. M., “An approach to the construction of a level description of classes by means of a predicate calculus language” // SPIIRAS Proceedings, 2014, No 3(34), pp. 204 - 217 (In Russian)

Authors' Information

Tatiana Kosovskaya *– Dr., Professor of St. Petersburg State University, University av., 28, Stary Petergof, St .Petersburg, 198504, Russia; Senior researcher of St. Petersburg Institute in Informatics and Automation of Russian Academy of Science, 14 line, 39, St. Petersburg, 199178, Russia; e-mail: kosovtm@gmail.com*

Major Fields of Scientific Research: Logical approach to artificial intelligence problems, theory of complexity of algorithms