TECHNICAL TRANSACTIONS 3/2018
CZASOPISMO TECHNICZNE 3/2018

ELECTRICAL ENGINEERING

DOI: 10.4467/2353737XCT.18.041.8303
SUBMISSION OF THE FINAL VERSION: 12/2/2018

Krzysztof Wójcik (krzysztof.wojcik@mech.pk.edu.pl)
Krzysztof Wójcik, Production Engineering Institute, Faculty of Mechanical Engineering, Cracow University of Technology

Dariusz Pałka
Department of Applied Computer Science, Faculty of Electrical Engineering, Automatics, IT and Biomedical Engineering, AGH University of Science and Technology

Olaf Bar
Institute of Computer Science, Faculty of Mathematics, Physics and Technical Science, Pedagogical University of Cracow

Micro-ontology building – the main variants of the OTO method

Metoda "oto" tworzenia mikro-ontologii – podstawowe warianty algorytmu

Abstract

This article describes the main properties of an iterative method of simple knowledge structure creation. The method is based on an inductive learning scheme. The knowledge structure is built automatically and takes the form of a simplified ontology. Knowledge transformation plays a key role in the process of creating the knowledge structure. In order to regular describe many kinds of these transformations the article provides the relevant theoretical background. The task of finding the proper ontology (knowledge structure) is extremely complex. This paper highlights the necessity to investigate efficient search methods; additionally, the work draws attention to the advantages that arise from building the knowledge structure at the minimal possible size. The paper points to possible areas of the method application, especially in connection with problems of the automatic understanding of images and websites.

Keywords: pattern recognition, image understanding, machine learning

Streszczenie

Artykuł przedstawia podstawowe własności iteracyjnego procesu (nazywanego w pracy OTO) tworzenia struktury wiedzy. Budowana automatycznie struktura przyjmuje formę ontologii. Artykuł prezentuje podstawy teoretyczne opisywanego procesu. Kluczową rolę odgrywa w nim zestaw specyficznych algorytmów transformacji wiedzy. Opisywany proces jest ekstremalnie złożony obliczeniowo. Artykuł podkreśla konieczność opracowania bardziej efektywnych algorytmów numerycznych, uwypuklając jednocześnie korzyści z budowy ontologii w minimalnej, możliwej formie (mikro-ontologia). Proces budowy wiedzy przybliżono z pomocą odpowiednio dobranego przykładu. W pracy wskazano na możliwe obszary zastosowań metody, w szczególności dotyczące automatycznego rozumienia obrazów oraz rozumienia stron WWW.

Słowa kluczowe: rozpoznawanie obrazów, rozumienie obrazów, uczenie maszynowe

1. Introduction

The World Wide Web may be considered to be an environment in which the phenomena of an observation, understanding, or action may happen. As in the real world, the phenomena may be described using a suitable language (e.g. XML, Spanish, or graph language for image representation).

The phenomena may be also modelled by the use of an ontology. The creation of the ontology involves an operation of defining objects and concepts, and establishing relationships between them [9]. Despite the existenceof several tools which support this process [2], initial assumptions mainly depend on the intuition and practical knowledge of a researcher [1]. This statement remains valid regardless ofthe kind of ontology domain and the level of abstraction of its elements. For instance, this may relate to ontology for the analysis of texts from the World Wide Web as well as the ontology for the system which search weapons in the pictures obtained from social networking services.

To overcome the above difficulties, we propose a certain general scheme of the ontology creation. We assume that the scheme enables the knowledge to be built with minimal prior assumption (inborn knowledge); additionally, the generated system should be able to automatically update the knowledge. The proposed method constructs the knowledge structure during an iterative scheme called an OTO (observation-transformation-operation). This approach is one of many possible variants of the widely known methods of structured knowledge building (see Literature remarks). In the paper, a brief example of the method usage is outlined (see Section 3.4). This is involved with problems of building a very 'low-level' concept structure corresponding to an abstract concept of an equinumerosity of sets

The same problems also concern the automatic creation of basic concepts for website-understanding systems (text as well asimages).

The proposed approach can be regarded as a kind of an algebraization of the problem. Unfortunately, the considered method suffers from a lack of efficient methods for searching the proper knowledge structure within an extremely large search space

Let us articulate the main goal of the paper. This is to encourage researchers to investigate the efficient numerical methods for searching the optimal knowledge structure.

The authors are aware that a condition of this research is providing a expanded theoretical introduction which defines the key terms – this constitutes a major part of the paper and is an extension of works [12–14]. The presentation of a consistent theoretical background is the second main goal of the article.

As has been previously noted, the knowledge creation process is based on minimal assumptions and can produce a group of fundamental (very low level) concepts (e.g. 'equinumerosity of sets' see [15]). It is difficult to compare the proposed methodology to other known methods of knowledge building which utilise the basic human-defined concepts (these operate on the upper levels only). For this reason, the method comparison is omitted. Due to a lack of efficient optimisation methods (see above), we have also omitted a comparison of the computational efficiency of our approach and existing methodologies; however, some similarities between them should be highlighted.

1.1. Literature remarks

This section presents key points that relate the proposed approach to well-known techniques of artificial intelligence (AI) – these are grouped according to some selected sub-disciplines of AI.

1.1.1. Image Understanding Techniques (IU)

From the many variants of IU methods, we will focus on techniques which rely on the extraction of semantic content from the phenomenon. This is achieved through the automatic reasoning process, which generates and verifies hypotheses regarding the phenomena [10]. This idea appeared in the 1980s, at present, however, mainly syntactic variants of this process have been developed [10, 11]. In this case, the knowledge is generated as some structured grammar [6, 7]. It can be observed that there is a strong similarity between this kind of knowledge description and the representation used in the proposed methodology (Section 3).

1.1.2. Learning from observations, inductive learning methods

According to proposed scheme, knowledge acquisition should occur automatically. One of the possible solutions is to base this process on the observation of the relationships between objects in a given reality. The first works using this approach appeared in the 1980s [4]. They introduced the idea of 'conceptual clustering' which, in a different shape, may also be regarded as a basis for our methodology. The 1990s brought the development of inductive learning methods. There was invented a methodology of Inductive Logic Programming (ILP) [5], which applies, as a formal description, a predicate calculus. The presented approach (Section 2) utilises this formalism in an analogous manner.

1.1.3. Automatic ontology building, semantic web methods

The literature [1] presents the main ideas of ontology engineering. The 'ontology', which is the central term, may be simply defined as a 3-tuple: a set of concepts, a set of relationships (between the concepts) and a set of instances [1]. The definition closely corresponds to the term 'hierarchical structure of concepts' introduced in Section 2. In recent years (especially since the beginning of this century) we have noticed a rapid growth in the use of Semantic Web methods [1]. This is caused mainly by their usage for semantic analysis and automatic understanding of the websites. There are several projects that deal with this matter, the best known of which is the OWL (Web Ontology Language) [9]. OWL is a knowledge representation language which enables the description of ontology elements like features of the classes and features of relationships, etc. It also allows the definition of some operations on the classes. The analogous operations concerning the knowledge structure are widely disputed in Section3.

2. Inductive scheme of knowledge structure building

Our first goal is to propose a method of defining the knowledge about the visible phenomenon. The created knowledge structure strongly depends on previously established assumptions. To decrease this impact, we can reduce them to a minimal form (this corresponds to the operation called 'phenomenological reduction' [3]). This reduction leads to the statement that the phenomenon consists of elementary, atomic parts – these are called primitive objects or instances. The objects correspond to primitive 'concepts' which are also called 'types' or 'classes'. The concepts are some kind of generalisation of objects (i.e. there exists a predefined, fundamental relation between objects and concepts called a membership). Furthermore, we can assume that the objects may be connected by some primitive relationships. We will use the following notations: $\mathbf{X}_0$, $\mathbf{X}$ – sets of objects; $\mathbf{C}_0$, $\mathbf{C}$ – sets of concepts; $\mathbf{D}_0$, $\mathbf{D}$ – sets of relationships between objects, in which the primitive objects, concepts, and relationships are denoted by $\mathbf{X}_0$, $\mathbf{C}_0$, and $\mathbf{D}_0$, respectively. They contain the whole initial knowledge. We can assume that all processes of knowledge increasing relate to the extension of these sets.

Since the concepts are the generalisation of objects, an occurrence of the specific objects in the given reality should be a condition of the building of new concepts (this universal idea is utilised in various kinds of inductive learning methods and learning based on observation [4, 5]). The objects which will be used in this process should be in some way significant; we assume that this will happen if they are connected by the relationships. The frequent, multiple presence of such objects should strengthen their ability to create concepts according to the simple rule: *this phenomenon occurs many times, so it must be important*. We will now be more precise about this.

Let us assume that we can select some primitive objects (from $\mathbf{X}_0$) and recognise them as particular instances corresponding to certain concepts (from $\mathbf{C}_0$). Additionally, we assume that it is possible to check all prior defined relationships between all the objects. Let us consider one such relationship, which will be indicated by:

r_i, where: $r_i \in \mathbf{D}_0$, $i \in \mathbf{I}$, $\mathbf{I} = \{1, 2, ..., u\}$ is a set of indices of relations.

We can assume that r_i has n arguments; thus, it may be satisfied by some n-tuple (sequence of n elements) indicated by:

$t, t \in, X_0^n$ where: X_0^n – n-th Cartesian power of set $\mathbf{X}_0$.

Of course, this particular relationship can be satisfied by many other tuples, which may be denoted by:

t_{ik}, where: $i \in \mathbf{I}$, $k \in \mathbf{K}$, $\mathbf{K} = \{1, 2, ..., m\}$ is a set of indices of tuples which satisfy r_i relation (the first index of tuple *tik,* points to r_i relation).

On the basis of selected tuples, we propose the construction of a new concept definition. Let us try to construct a group (set) of such tuples. To each tuple, we will attach some information that identifies the relationship which it satisfies. Let us create the pair (t_{ik}, i), which contains the selected tuple and the index i that points to the relationship.

We will define the proposed group (denoted by **G**) as an ordered set of pairs:

$$\mathbf{G} = \{(t_{ik}, i) : i \in \mathbf{I}, k \in \mathbf{K}\} \tag{1}$$

G corresponds to one particular occurrence of some collection of objects which are connected by relationships. A set of all possible groups **G** which may be created on the basis of the given scene will be denoted by $\mathbf{G}^*$, i.e. $\mathbf{G} \in \mathbf{G}^*$. Let us transform the group **G** by simply replacing each object in each tuple with the label of the object type. As a result, we obtain a set:

$$\mathbf{S} = \{(vtype_i, i) : i \in \mathbf{I}\} \tag{2}$$

where: $vtype_i$ – vector of labels of argument types of the ith relation, $vtypei \in \mathbf{T}^{n_i}$ (n_i th Cartesian power of **T**), **T** – set of labels, n_i – number of arguments of ith relation.

S describes an abstract (with regard to types rather than specific objects) arrangement of relations in the group. The groups that have identical or similar (according to some distance function) arrangements will be regarded as similar. A number of such groups may appear in the sequence of the input phenomena; therefore, let us consider a set of similar groups:

$$\hat{\mathbf{G}} = \{\mathbf{G}_j \in \mathbf{G}^*, j = 1, 2, ...\} \tag{3}$$

Based on set $\hat{\mathbf{G}}$, we may determine one of the most characteristic groups denoted by $\mathbf{G}_p$. This group may be transformed to some $\mathbf{S}_p$ set which represents its arrangement of relations (see above). The $\mathbf{S}_p$ contains information about the relationships and their types of arguments; therefore, it may be called a 'pattern'. We can denote a subtask of the $\mathbf{S}_p$ pattern creation with **FG**.

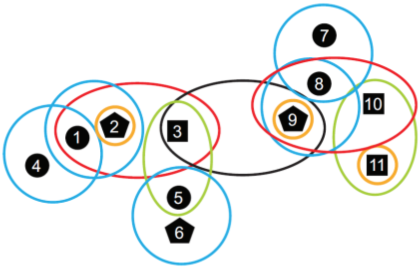

Fig. 1. Simple example of the group structure

Let us present a simple example of $\mathbf{S}_p$ pattern [13]. Figure 1 shows several indexed objects: $x_1, x_2, \ldots$ of three types (1 – circle, 2 – pentagon, 3 – square). A few relationships are also defined (r_1 – indicated by a red oval, r_2 – blue, r_3 – orange, r_4 – green, and r_5 – black oval). We can specify several tuples:

$$t_{31} = (x_2), t_{32} = (x_9), t_{33} = (x_{11}) \text{ (the } t_{31} \text{ denotes } 1^{st} \text{ tuple of relation } r_3)$$

$$t_{21} = (x_1, x_4),\ t_{22} = (x_1, x_2),\ t_{23} = (x_8, x_9),\ t_{24} = (x_8, x_7)$$

$$t_{41} = (x_3, x_5),\ t_{42} = (x_{10}, x_{11}),$$

$$t_{11} = (x_1, x_2, x_3),\ t_{12} = (x_8, x_9, x_{10}).$$

We can then point to two groups that are strongly connected by the relations:

$$\mathbf{G_1} = \{(t_{31},3), (t_{21},2), (t_{22},2), (t_{11},1)\},\ \mathbf{G_2} = \{(t_{32},3), (t_{24},2), (t_{23},2), (t_{12},1)\},$$

then: $\hat{\mathbf{G}} = \{\mathbf{G_1}, \mathbf{G_2}\}$ and as $\mathbf{G}_p$ we may chose $\mathbf{G_1}$, thus:

$$\mathbf{S}_p = \{((2), 3), ((1, 1), 2), ((1, 2), 2), ((1, 2, 3), 1)\}.$$

Returning to the main problem, let us assume that pattern $\mathbf{S}_p$ has been created. We will try to use it in the definition of a new concept. Nonetheless, such a definition must be completed with some additional elements. We may create a new concept (abstract knowledge); however, only the creation of more concrete items – objects belonging to a certain concept – make it possible to interpret a given phenomenon. The new object may be defined as a combination of several sub-objects (according to pattern $\mathbf{S}_p$). The question arises of whether such a new object may be applied as an argument of old relationships, or must the old relationships be redefined before that. Generally, this is dependent upon how long the new objects should inherit the behaviour of their parents.

Let us consider some relationships that assign certain values to objects. These values may be regarded as object attributes (properties) [9]. We encounter a similar problem: should the new objects have the new kinds of attributes? If so, we ought to provide methods to calculate them. For example, we may use combinations of some standard transformations, e.g. copying, calculation of a sum or an average of parent objects attributes. Finishing these remarks, let us denote processes of defining the new relationships and object attributes using **FR** and **FA**, respectively. Finally, we can define the concept as a composition of elements:

$$(\mathbf{S}_p, \mathbf{FR}, \mathbf{FA}) \tag{4}$$

The new objects, which have been created on the basis of the new concepts, may be utilised for building further concepts. This process may be carried out cyclically. This results in creating a 'hierarchical structure of concepts' and a 'hierarchical structure of objects'. The concept structure includes new general knowledge and can be treated as a model of reality. Thanks to its hierarchical form, such knowledge may be easily interpreted, verified and adapted to other systems. Using the concept structure as a set of patterns, and employing the set of primitive objects which describe the unknown phenomenon, we can produce new object structure – this structure may (or may not) contain objects of particular types. We can postulate that the presence of these objects can be considered as a kind of system response – it may be directly employed in tasks of objects classification, understanding, control, etc.

We should notice, however, two crucial matters concerning the whole approach. The first is associated with the problem of knowledge evaluation. Let us present here only general ideas:

Evaluation by the directly defined performance function.

The hierarchical knowledge in the form of the concept structure may be relatively easily interpreted. In many cases, this allows the identification of concepts that correspond to crucial states of the system. The presence of such concepts in the knowledge structure might be expressed by a particular mathematical formula. This statement may be the basis for the definition of the performance function.

Evaluation with the help of an arbiter, teacher (supervised learning).

The evaluation of the concept structure may be performed by examining the system responses to certain sets of objects. The objects and the proper system replays (determined by the teacher) create pairs which may be treated as elements of the learning sequence [8].

The second crucial problem is the way of finding a suitable knowledge structure; this is dealt with in the next section.

3. OTO multistage process of knowledge building

The process of knowledge construction starts from establishing and defining primitive objects, concepts and relationships ($\mathbf{X}_0$,$\mathbf{C}_0$, and $\mathbf{D}_0$). This initial phase may be performed in several ways. However, the main part of the concept creation involves multiple executions of the **FG**, **FR**, and **FA** tasks – there is a huge number of possible variants of these. Consequently, finding a suitable structure should be treated as the optimisation of a criterion function in an extreme large, multi-dimensional search space. Trying to solve the problem, we will make attempts to reduce the search space size. First, let us again consider the stages of the method; these can be briefly enumerated as follows:

1. calculation of all possible relationships between all objects from **X** set.
2. determining $\hat{\mathbf{G}}$ set.
3. deriving $\mathbf{S}_p$.
4. establishing new relations and new definitions of object feature.
5. creating new objects.

Let us group these issues into three major stages:

A. The observation (corresponding to 1st–3rd items)

In this phase, we identify the relationships between the elements (objects) of the reality. This process provides the answers to the following questions:

What is visible at present? (i.e. in this step).
What properties do the visible objects have?
What kinds of relationships exist between the objects?

We only take into account the observed relationships; therefore, we 'keep a connection' with the observed reality. This leads to decreasing the size of the search space (hypothesis space).

B. The transformation of knowledge (corresponding to the 4th item)

The aim of this process is to change knowledge in order to better describe the observed phenomena in the future (i.e. in the next steps of the process). This means that the transformed knowledge may take into consideration such relationships which are not directly observed. Thus, the main kinds of these transformations should lead to knowledge generalisation – this increases the size of the search space.

C. The operation (corresponding to the 5th item)

The essence of this process is upgrading the current representation of the reality. The representation is conveyed by the structure of the objects. The operation process may lead to the creation of the new objects; therefore, it may significantly increase the size of the search space. The reverse process is also possible – elimination of the object may cause a decrease of the search space size.

The presented scheme may be considered as a kind of framework. Let it be denoted by OTO (Observation-Transformation-Operation).

Let us try to estimate the order of the number of all solutions generated by OTO. We assume for simplicity, that we have m subtasks on each pass of the OTO loop. In each i th subtask, we may choose one of p_i possible ways (the edges of the decision tree). Therefore, the number of possible solutions might be approximated by:

$$h_{oto} = \left(\prod_{i=1}^{m} p_i \right)^2 \tag{5}$$

where: n – the number of passes of the main OTO loop.

The expression (5) is an imprecise estimation, m and pi usually vary with passes of the loop. Let us assume that the size of the knowledge structure may be expressed by the number of concepts which have been included in it. For simplicity, we do not take into account the number of relationships between the concept and the number of arguments in particular relationships.

Additionally, let us presume that the number of concepts produced in each OTO loop is limited to a certain, small value (normally less than 5). Therefore, the size of the structure is roughly proportional to the number of passes of the OTO loop (denoted by n). Formula (5) clearly shows a radical, exponential growth in the number of possible solutions with an increasing knowledge size (proportional to n). We may also interpret that relationship as a growth of the computational complexity with the problem size (expressed by the number of concepts); thus, computational complexity is $2^{O(n)}$. This is a key problem of the whole presented approach. In practical, even in very simple implementations this number exceeds millions.

The OTO process is executed in distinct phases. This allows the beneficial use of some specific problem-solving techniques [8]. We will depict only a general view of selected techniques (the authors' paper [12] provides more details).

The bundle search methods

At each stage of concept creation, we can calculate the value of some local criterion functions (objective function) which assess the ability of the constructed structure to create useful knowledge. It especially refers to the task of the proper choice of set $\hat{\mathbf{G}}$. The value of the criterion function may depend on the number of groups that belong to set $\hat{\mathbf{G}}$ (a typical approach is 'the more the merrier'). The criterion function might be used to rank all possible solutions. In further phases of the searching process, we may only take into account a limited number of potential best solutions (see [8]).

Monte Carlo methods.

Some kinds of Monte Carlo methods that combine random searching and bundle methods can be used. First, we rank the possible edges of the decision tree according to the local criterion function. Then, one edge is selected randomly from the top of the ranking list.

In order to decrease the computational complexity of the search process, we should focus especially on the opportunity to generate the structure at a minimal size. To achieve this, let us consider the possible variants of subtasks of the OTO in more detail. This study will follow according to the main stages – observation, transformation, and operation.

3.1. Variants of the observation process

In our earlier proposal, the observation refers to calculating all relationships between objects from reality. However, the question is what we want to consider as the observed reality. In a broader sense, it may be everything that brings in information, not only the particular objects (instance) of a concept but also the concepts as general beings. We may doubt that the concepts, which are abstract knowledge, may be a source of information about a real phenomenon. Let us remember, however, they have been produced according to some real objects.

The observation of the concepts may involve finding the similarity between them. This makes it possible to build a new concept after the identification of an analogy in the relation patterns of concepts created earlier. To sum up, with reference to the subject of the observation phase, we may list three, very general variants of the observation method.

Obser.1. (this abbreviation refers to a criterion of this simple classification; this notation will be continued in the next points).

1. Set of objects.
2. Set of concepts.
3. Set of relationships.

The next criterion refers to some constraints of the observation processes.

Obser.2.

1. All the objects are observed at a particular stage.
2. Only objects that are significant in some way areobserved. In this way, the observation process contains a kind of filtering.
3. One-off objectusage.
   Let us presume that an object denoted by $x2$ has been created based on object x_1. Let us then assume that we want to create another object x_3 from two objects $x1$ and x_2. This case is an example of many possible kinds of multiple inheritance (object x_3 contains redundant information). The logical solution is to ignore all the relationships that connect $x1$ and x_2 at the observation stage. This variant of the observation leads to a significant decrease in the size of the search space.
4. Central object method with binary relationships.
   Let us simplify the creation of the **G** sets. The first assumption is that a specific object (called the 'central object') exists, which is an argument of all relations in **G**. Secondly, let us only take binary relationships into account. As a result, the complexity of the observation process grows with the number of objects like: $O(n^2)$ (for each potential central object from **X** we must analyse relationships to all other objects from **X**). The presented approach is efficient, but the open problem is the possibility of substitution of n-ary relationships by combinations of the binary ones.

3.2. Variants of the transformation process

The overview of the variant of the $\mathbf{S}_p$ set transformation will start from modifications of relationships in the pattern. Let us point to main ideas.

Tran.1.

1. Replacing one, or more, indices that identify the relationships in the $\mathbf{S}_p$, with the other indices, see expression (2). We assume, that the new relationship has identical argument types as the previous relationship.
2. Creation of a new relationship through modification of the old relationship.
   The modification may consist in changing the relation parameters (for example a relation *to_be_big* obviously depends on the given threshold). It may also refer to combining the relationships with the other ones. In this way, the set of objects that satisfy the relation can be widened or narrowed (the disjunction and conjunction of the relations).
   We will now consider the other kinds of transformations that modify the number of satisfying objects. Let us take into account binary relationship r_0. We can imagine the new unary relation r_1 that is satisfied by the particular object $\boldsymbol{x}$, if the relationship r_0 is held between the $\boldsymbol{x}$ and only one other object from set **X**. Similarly, we may define the new relationship r_2 which will be satisfied by x if r_0 is satisfied by x and all the objects from **X**. In the same exact way, we may define r_3 if r_0 is satisfied by x and none of the objects from **X**. This kind of modification will be called an 'excluding transformation'. An obvious, but important group of relationship

modifications refers to changing their type of argument. Such transformations are very easy to achieve: suffice is to change one data (index of argument type) in set $\mathbf{S}_p$. This modification may have far-reaching consequences. Let us presume that set $\mathbf{S}_p$ defines a concept indexed by *j*. In $\mathbf{S}_p$, we are changing the type of index of a particular argument of a chosen relation into *k*. Therefore, the concept indexed by *k* is now used to define the concept indexed by *j*. Thus, we may consider this process as the operation of joining one item of knowledge to another.

Let us consider one important, special case of changing the argument type. Let us presume that, as above, the concept indexed by *j* is defined according to some pattern $\mathbf{S}_p$. Technically speaking, it is possible to change some index of the argument type of a selected relation in $\mathbf{S}_p$ also into *j*. In this way, we obtain a 'recursive' type which is defined by itself. The recursive definition cannot 'go' to infinity; there is a necessity to use an alternative definition of the considered concept (see the following points). The modification of the argument type, including creating the recursive type may be considered as a kind of structural transformation. To sum up, let us list some important variants of the relationship transformation.

a) the change of the relation parameters
b) the modifications that the number of objects that satisfy the input relation
c) the simple change of the type of argument to another type
d) the creation of the recursive type (special case of the previous point)
e) the enlargement of the set of possible argument types of certain relationship (in an extreme case the use of objects of any type is allowed)
f) the narrowness of the set of the possible argument type.

To finish the topic of relation transformation, let us remember the **FA** process, leading to the creation of new object features (also regarded as relations) – these may be calculated in different ways. Two main strategies are:

Tran.2.

1. Inheritance of the method of calculating object features from parent concepts.
2. Establishing the new method of the computation of features.

At this time, let us discuss a few basic transformations that change the structure of $\mathbf{S}_p$ sets (structure of concepts). Let us assume that we have created a copy of one particular pattern $\mathbf{S}_p$. Afterwards, this copy is transformed using some of the described methods. In this way, two different $\mathbf{S}_p$ patterns create alternative concept definitions. This leads to knowledge increasing, and may be regarded as an extension of the concept definition (4).

Let us reflect on a reverse operation – the deletion of some concepts. A simple variant of the operation involves removing a given concept and all the concepts which have been created from it. Simultaneously, we have to delete all objects (instances) of cancelled concepts. This reduction has a significant impact on the observation process. The decrease of the number of concepts and objects leads to a radical reduction in the number of generated tuples, as a consequence, this limits the computational time of the whole process. We can propose several ways of selecting the concept to be removed, e.g. deleting such concepts which were not used to produce the new concepts over a few runs of the OTO loop. To summarise this subsection, let us itemise the major variants of the described transformations.

Tran.3.

1. Permission (or not) for the following structural operations:
   a) the definition of alternative concepts,
   b) concept joining,
   c) concept deleting.

3.3. Variants of the operation process

The operation phase may concern two main processes – the creation and the removal of objects. The first may occur in the following main variants:

Oper.1.

1. Creation of all possible objects.
2. Creation of a reduced set of only the most dissimilar objects.

Other variants relate to the problem of recursive object creation (see above – creation of the recursive type).

Oper.2.

1. Permission (or not) for the creation of recursive objects. In order to avoid the creation of the infinity object structure, we should define specific stop criteria. For example, the object creating process may stop when:
   a) the currently created new object includes itself (among others) as an ancestor object
   b) the currently created new object is similar to the other objects which have been created in the previous passes of the loop (according to the distance function in the space of object attributes).

The second elementary operation refers to object removing.

Oper. 3.

1. Removal of the 'old' objects (which have been created in the earlier runs of the OTO loop).
2. Removal of the unused objects (which are not used in the creation of other objects over a given number of iterations of the OTO loop).

The presented description of the method variants does not pretend to be a strict classification – we have taken into consideration only the most important criteria.

4. A brief example of the application of the OTO scheme

As previously noted, the possibility to create an object of a certain type may be regarded as a system output. The newly constructed, potentially complex object is instantly, automatically classified (just as an object of the given type).

Paper [14] provides an example of the usage of the proposed approach for image recognition tasks. The generating system is able to identify specific spatial relationships between visible objects. A more complex classification is described in [15]. First, the system creates the abstract concept of an equinumerosity of sets. The system should build this concept with minimal prior assumptions (without using its well-known definition that utilises a bijection transformation). Let us briefly portray this problem.

Primitive objects will be considered; these are vectors having only one feature – *colour* (with two values: 0 and 1 or *blue* and *red*). Additionally, primitive binary relationships are defined: *equality* and *inequation* of the *colour* and unary relations: *having_blue_color* and *having_red_color*. The following variants of the concept-creating process are utilised (according to Section 3): **Obser.1.** 1 (1st variant); **Obser.2.** 2, 3, 4; **Tran.1.** 2b, 2c, 2d; **Tran.2.** 1; **Tran.3.** 1a; **Oper.1.** 2; **Oper.2.** 1a, 1b.

The search program has generated many appropriate solutions, we will depict only one:

typ 20 q1 pat:
typ 21 q1 pat: 13: 20 20
typ 22 q1 pat: 15: 21 21
typ 22 q1 pat: 15: 22 21
typ 24 q4 pat: neg 13: 22 20 15: 22 20 15: 22 21

This definition can be interpreted as follows:

typ 20 q1 pat:

typ 20 – a header of the definition of concept (type) no. 20, regarded as primitive objects, q1 – a performance value of this concept, (not significant), pat: – relation definitions (empty here);

typ 21 q1 pat: 13: 20 20

typ 21 – the definition of concept 21 (in further points, headers will be omitted), 13: 20 20 – relation 13 (*inequation*) held between two objects of type 20; the object of concept 21 is a pair of primitive objects (type 20) which have different colours, such a pair will be called a 'different pair'; the colour of this compound object is always 0 (*blue*) as a result of calculating the average colour of components in an integer domain; the first arguments of all binary relations are 'central objects' (Subsection 3.1),

typ 22 q1 pat: 15: 21 21

15: 21 21 – relation 15 (*equality*) held between two objects of type 21; the object of type 22 contains two objects of type 21 (different pairs); all objects of type 21 have *blue* colour, so relation 15 indicates here the existence of another object of type 21;

typ 22 q1 pat: 15: 22 21

15: 22 21 – relation 15 (*equality*) held between objects of type 22 and 21, this is a transformation of the previous concept, type 22 becomes the 'recursive type'; the object of extended concept 22 may be a set of different pairs (type 21);

typ 24 q4 pat: neg 13: 22 20 15: 22 20 15: 22 21

this is the 'excluding transformation' (denoted by 'neg') of relations:
13: 22 20 15: 22 20 15: 22 21;
object of type 24 may be interpreted as a set of different pairs (type 22) for which:
13:22 20 – another object (type 20) having a different colour does not exist,
15:22 20 – another object (type 20) having the equal colour does not exist,
15:22 21 – another pair does not exist (all pairs have *blue* colour).

The last conditions mean that an object of type 24 includes all of the different pairs, additionally, there exists no other object (type 20) that is not a component of them. As a consequence, the possibility of creating the object of type 24 indicates that two sets have the same cardinality. Figure 2 shows a tree structure of a chosen object of type 24.

The usage of the equinumerosity concept and performing of the operation of knowledge joining (see Subsection 3.2) enables the establishment of a percentage composition of some examined objects that are visible on the image [14]. The analogous usage of the method may relate to website interpretation. We know that the created object corresponds to the **G** group of relations which describe the observed situation (Section 2). We may say that the group contains the semantic content of the situation (phenomenon). We regard the extraction process of the semantic content of the given situation as a condition of its understanding (Section 1.1). As a consequence, the described method may be utilised in the automatic understanding of complex sentences or multipart pictures.

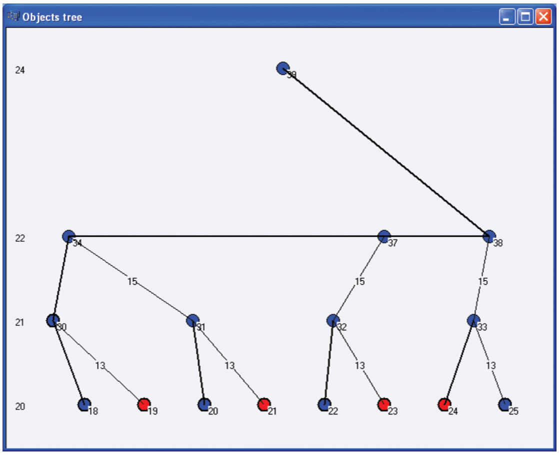

Fig. 2. An example of the tree structure of an object of type no. 24. All the objects on the same level are of an identical type (their numbers are written on the left). Bold line, leading to the particular object, indicate which one of its parent objects is the 'central object'. The other lines (edges) symbolise the binary relationships between the central object (mentioned above) and the other parent objects (the number of the relation is placed in the middle of the line)

5. Conclusions

The paper presents the multi-stage scheme of creating simple ontology. The creation process may start on the basis of minimal assumptions. The phases of the scheme, which correspond to knowledge generalisation and specification, allow convergence of the knowledge structure into the desired form. Consequently, the knowledge system has the ability to self-build and self-update. In this context, the presented method satisfies the requirements specified in Section 1. We can highlight some properties of the presented scheme, for example, the ability to construct knowledge without a model of given phenomena that extends the capability of classical knowledge systems (expert systems) [8].

The main drawback of the presented approach is the lack of efficient methods of searching the optimal knowledge structure. The invention of such methods remains the major challenge. In many parts of the paper, we accent the necessity of building the knowledge structure at the minimal possible size (micro-ontology). The article indicates several ways to achieve the knowledge simplification. The most important of these are:

- an ontology definition which uses relations between objects only;
- usage of binary relationships;
- applying the 'central object' method;
- using the operations of deletion of objects and concepts.

The usage of the OTO model may be very promising in areas where the knowledge has an evidently structural form, especially for the automatic understanding of the websites and the image recognition tasks.

References

[1] Davies J., Studer R., Warren P. (eds.), *Semantic Web Technologies Trends and Research in Ontology-based Systems*, John Wiley & Sons Ltd, 2006.

[2] Gennari J. et al., *The evolution of Protégé, An environment for knowledge-based systems development*, Int. Journal of Human-Computer Interaction, 58, 2003.

[3] Husserl E., *Logical Investigations*, Routledge, New York 2003 (first published in German as Logische Untersuchungen, M. Niemeyer, Hale 1900/1901).

[4] Michalski R.S., Carbonell J.G., Mitchell T.M. eds., *Machine Learning: An Artificial Intelligence Approach*, Springer Science & Business Media 2013.

[5] Muggleton S., *Scientific knowledge discovery using inductive logic programming*, Communica-tions of the ACM, Vol. 42, 1999.

[6] Piekarczyk M., Ogiela M.,R., *Matrix-based hierarchical graph matching in off-line handwritten signatures recognition*, Proc. of 2nd IAPR Asian Conference on Pattern Recognition, IEEE, 2013.

[7] Piekarczyk M., Ogiela M.R., *The Touchless Person Authentication Using Gesture types Emulation of Handwritten Signature Templates*, Proc. of 10th ICBWC, Coommunication and Applications BWCCA, Krakow 2015.

[8] Russell S., Norvig P., *Artificial Intelligence: A Modern Approach*, 3rd edn, Prentice Hall, En-glewood Cliffs 2010.

[9] Smith M.K., Welty C. (eds.), *OWL Web Ontology Language Guide*, http://www.w3.org/TR/owl-guide (access: 10.11.2016).

[10] Tadeusiewicz R., Ogiela M.,R., *Medical Image Understanding Technology*, Studies in Fuzziness and Soft Computing, Vol. 156, Springer-Verlag, Heidelberg 2004.

[11] Szczepaniak P.S., Tadeusiewicz R., *The Role of Artificial Intelligence, Knowledge and Wisdom in Automatic Image Understanding*, Journal of Applied Computer Science, Vol. 18, 2010.

[12] Wójcik K., *OTO Model of Building of Structural Knowledge – Areas of Usage and Problems*, Advances in Intelligent Systems and Computing, Image Processing and Communications Chal-lenges 4, Springer-Verlag Berlin, Heidelberg 2012.

[13] Wójcik K., *Indutive learning methods in the simple image understanding system*, ICCVG' 10 Proceedings of 10th International Conference on Computer Vision and Graphics, Part I, LNCS, Springer-Verlag Berlin, Heidelberg 2010.

[14] Wójcik K., *Hierarchical Knowledge Structure Applied to Image Analyzing System – Possibilities of Practical Usage*, ARES'2011 Proceedings of the IFIP WG 8.4/8.9 International Cross Domain Conference Viena, LNCS Springer-Verlag Berlin, Heidelberg 2011.

[15] Wójcik K., *Knowledge Transformations Applied in Image Classification Task*, Advances in Intelligent Systems and Computing, IPCC 5, Springer-Verlag Berlin, Heidelberg 2013.