Proceedings of the 9^{th} International Conference on Applied Informatics
Eger, Hungary, January 29–February 1, 2014. Vol. 1. pp. 67–74
doi: 10.14794/ICAI.9.2014.1.67

Role of negative properties in knowledge modeling[*]

László Kovács

University of Miskolc
kovacs@iit.uni-miskolc.hu

Abstract

An appropriate and effective knowledge domain model is the key element of any intelligent systems. The dominating knowledge models in information technology, like ontology or formal concept analysis [2] are strongly related with the current SE modeling tools from the viewpoint of property modeling. These models use positive and mainly binary valued properties. On the field of philosophical ontology, this can be a valid assumption, but IT ontologies are used sometimes for modeling epistemology. The epistemology model is the result of a cognitive process using deductive and inductive methods. In this model, the lack of some property can be an important characteristic of the objects. The paper presents a survey on interpretation of negative property [1] in philosophy and IT ontology. On both fields, very different approaches compete with each other [3]. In the analysis, we argue for the acceptance of negative properties in knowledge modeling. The main benefit of the dual representation of positive and negative properties is a finer description of the support values for facts during an inductive learning process. This extended structure provides more information compared even with the multi-valued logic or fuzzy representation models. The paper demonstrates the construction of knowledge graph using the dual property model.

Keywords: ontology, negative property, knowledge model

MSC: 68T30

1. Introduction

The area of ontology is the central element of current researches in knowledge engineering. The ontology is an umbrella term that includes methods to provide a high level, standardized management of shared conceptual models. According to

[*] This research was carried out with the support of the TAMOP 4.2.1.B-10/2/KONV-2010-0001 project supported by the European Union and by the European Social Fund.

the definition of Tom Gruber[1], an ontology is an explicit and formal specification of a conceptualization. A key characteristic of ontology is that it describes the objective world, it contains the truth on the world. The base components of ontology models are the concepts, the instances, the relations between them and the logical axioms. The methods of ontology like OWL or description logic are efficient tools to work with semantic databases. There are many application areas where ontology is applied to enhance the standard databases with semantic knowledgebase, like software development [5] or medical applications [7]. A special area of knowledge management focuses on the modeling of agent behaviors. The goal of this kind of investigation is to simulate and to understand the decision processes within a subjective agent. The development of personalized applications requires the modeling the subjective knowledge bases of subjective agents. This area is called epistemology [7]. One of the main differences between world ontology and subjective ontology that the subjective ontology is mainly the result of a learning process where some information is incomplete, partial.

The paper focuses on the management of incomplete information in subjective concept databases and it discusses the role of negative properties in both the concept specialization processes and in the management of incomplete information. The last section of the paper demonstrates the benefits of the proposed model in the domain of automated e-tutor framework.

2. Knowledge representation forms in ontology

The classic approach in knowledge representation is the application of binary valued logic. The *true* and *false* values give an efficient representation of the truth values. The crisp binary truth value is also a fundamental part of membership grade representation in the traditional conceptual modeling. Considering the classic ontology models [6], the relationship between concept a and an another concept or value (as instance concept) b is a strict relationship. The methodology of classic FCA (Formal Concept Analysis) uses the same approach [2]. The context of the FCA is based also on a strict relationship between objects and attributes. This context can be represented also with a binary values attribute-set: the attribute a at object o is 1, if $a(o)$ is true, otherwise the attribute a has a value 0. Thus every concept c can be uniquely represented with an attribute set, the intention part of the concept, where this set is a crisp set. The main benefit of FCA model is that it provides an efficient framework for concept generalization. The generalization g of a concept c has an intention set which is subset of the intention set for c. Thus only the existence of some common properties are the key factors in the concept generalization process. Considering, some others classic software engineering design models, like Entity Relationship or UML, although the attributes have a multi-valued domain, the existence of an attribute at an entity is a strict binary valued parameter.

The binary valued logic in modeling of the open objective world is accurate tool, as the law of excluded third is accepted as a general rule of our objective world.

Thus p or $\neg p$ is true for every proposition p. The inaccuracy of this binary valued logic was detected when the modeling was extended to subjective knowledge bases. The relative ontology is the conceptual model generated by a cognitive process. During the learning process, the input from the environment is used to update the conceptual model. The relative ontology depends on the training set of the agent.

Considering a training set O on concept c and a property p, then if for every transaction in $O : a(c)$ is met, then the relative ontology will contain $p(c)$ too. On the other hand if in all of the training transactions $\neg p(c)$ is met, then the model should contain the negation of p. It may happen that some transaction contains $p(c)$ while others contain $\neg p(c)$. A very different case is when the transaction does not contain information on $p(c)$. For these models, the simple binary valued attribute is not accurate enough and the logic model was extended to different directions.

The main approach is to use multi-valued logic. There are many different approaches to implement the multi-valued logic. The simplest case is three-valued case, the three truth values are true, false and unknown. This model is used for example in SQL standard too. Lukasiewicz, Tarski, Post, Gödel investigated the different extension of two-valued logic where some internal structure is assumed among the different truth values. A simplified model is among other the information system model [8] where the truth values are given with a simple enumeration.

3. Role of negative property in concept specialization

The usual approach is knowledge representation is to denote only the existence of some phenomena, i.e. usually the existence of some property. The not existence is considered as a derived property, a function of the existence property, similar to the other compound property expressions.

According to the statements of Armstrong [9], if p and q are existing properties then $p \vee q$ and $\neg p$ are derived and not real properties. He argues that there is no negative or disjunctive property. The central elements of his reasoning is that a property must have a causal power and the a mere lack or absence itself cannot cause any existence. In this sense, the negative property is not a property, as it means the absence of some property. Meixner [2] shows a good argumentation against this reasoning. He says, if p is a property and not p is not a property, what about the double negation, i.e. not not p, which should be equal to p again. If the negation is not a property, nor its negation can be a property. This a controversial inference, thus also the negative property should be a property too. Another pioneer work on considering negative properties relates to Russell [4], who introduced these concepts to solve the paradox of Demos.

From a more practical approach, the negation has an important role in every logic models. For example, in propositional calculus the formula

$$p \rightarrow q$$

can be expressed also with

$$\neg q \to \neg p.$$

In this sense, if this equivalent logic form is accepted as valid form, also the absence of some property can cause some consequences, thus it has a causal power. Taking a more formally approach based on the propositional logic, a negative property p^- has always a positive counterpart property p^+ and it can be defined here as

$$p^-(x) \leftrightarrow \neg p^+(x).$$

Also the argumentation of Meixner in [2] is based on this approach, it assumes a strict correlation between the positive and negative properties.

In the case of more complex logic model, this strict correlation can be released, the both properties can be independent from each others. Let us take a normalized multivalued logic with three logic values: (T,F,U). In this case, the definition of negative property can be refined into the following form:

$$p^-(x) = 1 - p^+(x) - p^u(x)$$

or

$$p^-(x) \leftrightarrow 1 - p^+(x) - p^u(x).$$

In this approach, the negative property has a larger independence from the positive counterpart, thus the explicit representation of p^- has a larger importance.

One of the applications of negative property in knowledge engineering technologies relates to the work of Kourie[10]. The knowledge database is constructed from a context similar to the context in Formal Concept Analysis. The context contains only those properties which are relevant in the problem domain. Kourie showed that if p is a relevant positive property than its negative counterpart is also a relevant property. To manage uncertainty, a three-valued logic is proposed in logic representation. The knowledge model consists of two base relations among the concepts: abstraction and its inverse, the refinement ($<$) relation. The set of properties that are true for concept C is denoted by $\mathit{true}(C)$. The Kourie assumption states the following for any two concepts C_1, C_2:

$$C_1 < C_2 \to \mathit{true}(C_1) \supseteq \mathit{true}(C_2), \forall p \in \mathit{true}(C_1) \backslash \mathit{true}(C_2) : \exists p' \in \mathit{true}(C_2) : p < p'$$

This means, that every sub-concept must have at least one such characteristic property which does not met at the super-concept, but there exists a generalized property that is true for the super-concept. To evaluate this approach, let us take the following simple concept hierarchy:

C_0: vehicles

C_1: vehicles that can fly

C_2: vehicles that can only fly

It is clear that regarding the concept content sets (every item that can only fly, can fly), the following relations are met:

$$C_2 < C_1 < C_0$$

Thus there should be a property of C_2 which is not met in C_1. But every property that is required to fly must be fulfilled in both C_2 and C_1. The difference between C_2 and C_1 is that the elements of C_2 have no such properties that are necessary to run on earth surface or on sea. In this way, the absence of some positive properties are the distinguishing properties. Thus it is reasonable to allow also negative properties as characteristic property.

These negative properties (like has_no_wheels, not_swim, etc) correspond to such properties which are met in the super-concepts. The super-concept may contain both the counterpart property or one of its generalization forms. Based on this reasoning, the following revised form of the Kourie assumption is proposed for the refinement relation:

$$C_1 < C_2 \to true(C_1) \supseteq true(C_2), \forall p \in true(C_1) \backslash true(C_2) : \\ \exists p' \in true(C_2) : p \to p' or \neg p \to p'$$

The original formula is extended with negative properties and the specialization relation is substituted with an implication relation. The implication between the corresponding properties provides a more general description, as the original specialization relation. In the given example, the characteristic property can be among others cannot_swim and its negation can imply the can_move property which is valid for C_1.

4. Information system with negative property

The proposed negative property can be used to enhance the expression power and accuracy of the implemented knowledge models. The standard model in knowledge engineering is the OWL ontology model. The term ontology in knowledge engineering is related to an open, shared and explicit conceptualization of the objective problem domain [1]. In our research, we focused not on the objective world, but on the subjective world of the human and machine agents. In this sense, the scope of the work relates to epistemology, but the tools and methods are borrowed from the ontology domain.

A standard ontology can be defined as a tuple

$$\langle C, R, I, M_I, M_R, A \rangle$$

where

- C: set of concepts
- R: set of roles
- I: set of instances
- $M_I : C \to \wp(I)$: instance assignment

- $M_R\colon I \times R \to \wp(I)$: role values
- A: axioms (integrity rules)

The model based on roles can be converted on a straightforward way into a predicate model. The roles can be represented by corresponding predicates on the domain of instances. The property predicates can be used to describe the ontology model. In this model

$$M_R\colon I \times R \to \wp(I)$$

is replaced with

$$P_R = \{r | r\colon I \times I \to \{\text{true}, \text{false}\}\} : \text{role predicates}$$

In the standard ontology model, the standard logic is used, i.e.

$$r^+(i,j) = \mathit{true} \leftrightarrow j \in M_R(i,r)$$

Beside the positive predicate, the corresponding negative predicate is defined with

$$r^-(i,j) = \mathit{true} \leftrightarrow j \notin M_R(i,r)$$

Using a three-valued logic, the formulas can be rewritten as

$$r^+(i,j) = l(j) \in M_R(i,r)$$

$$r^-(i,j) = l(j) \notin M_R(i,r)$$

where $l(e)$ denotes the logic value of event e.

5. Knowledge model for AQG

An important and very promising application area of knowledge engineering is the e-learning sector. The assessment module is used by students to write tests to check knowledge level. An important subtask of the assessment system is the generation of questions. Automatic question generation (AQG) can be used to provide a flexible user-adaptable set of test questions. Our prototype system implemented an important kind of test questions, namely the multi-choice selection questions

The key component in automatic question generation system is the semantic knowledge model of the target domain. George Miller at the Cognitive Science Laboratory of Princeton University [12] created a lexical semantic net called WordNet. Regarding the generation of candidate words, the model presented in [11] a uses simple semantic distance approach based on the semantic graph of the concepts. The distance is defined as the minimal number of links, edges on the paths connecting the two concept nodes. The system selects concepts near the targets, far from the target into the list of candidate words.

The baseline system for grammar exercises classified the candidate distracters on the basis of semantic similarity between verbs according to the distributional data. The distracters were compared for similarity employing the information radius measure. Furthermore, all candidate distracters had to match the target verb in terms of transitiveness, tense and person.

The model contains a set of characteristic functions, Every concept has a corresponding function which describes its relationships with the other concepts. The domain of a characteristic function is the whole set of concepts. The co-domain of the characteristic function is the strength, the truth value of the relationships. From this point of view, the characteristic functions are similar to fuzzy functions. where the value is calculated with [13]:

$$f_g^{'}(a) = \frac{\sum_i f_g(i) f_i(a)}{|C_i|}$$

and this value is always in the interval [0..1]. These functions can be used to describe the characteristics of the concepts. In the proposed approach, these functions will be used to measure the similarity and dissimilarity of the candidate words.

In our system, an information content distance was used which is based on the entropy of the differences. The concrete domain relates to a small subset of birds and animals, including 15 instance concepts. The set of attributes includes 12 base attributes, among others the can_fly, large, small, has_2_legs, has_4_legs and heavy properties. A subset of generated fuzzy membership functions is shown in Fig 1. With the calculation of the dist(can_fly,ci) distance values, a list of ci instance concepts in decision-uncertainty order can be generated. As an example, we take two calculated difficulty values: dist(can_fly, sparrow)=0.72 and dist(can_fly, penguin)=2.93. The yielded results are in accordance with our experiences, the calculated distance values can be used to estimate the difficulty of the decisions.

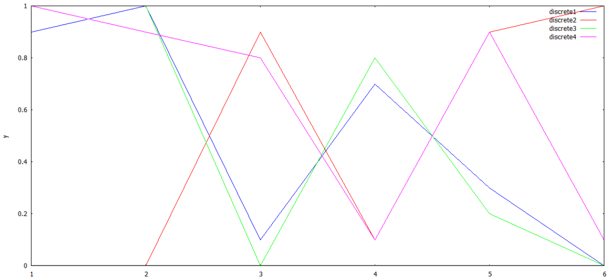

Figure 1: Fuzzy characteristic functions of the concepts

References

[1] GRUBER T, A translation approach to portable ontology specifications, Knowledge Acquisition, 5(2) (1993), 199–220.

[2] MEIXNER, U. On negative and disjunctive properties, Language, Truth and Ontology, Kluwer Publ. (1992)

[3] GANTER, B, WILLE, R, Formale Begriffsanalyse, Mathematische Grundlagen, Springer Publ., (1996)

[4] GRUNER, S, Abstraction, Refinement, Enrichment, Formal Aspects of Computing,, Shaker Publ., (2013)

[5] GUARINO, N., Formal Ontology and Information Systems, Proc. of FOIS'98, (1998), 3-15.

[6] MA, Y., NAZARENKO, A, AUDIBERT, L, Formal Description of Resources for Ontology-based Semantic Annotation, Proc. of LRE, (2010), 3765-3772

[7] BODENREIDER, O., SMITH, B., BURGUN, A., The Ontology-Epistemology Divide: A Case Study in Medical Terminology, Proc. of FOIS'2004, (2004), 185-195.

[8] LIU, Q., LIU., L., Rough Logic and its Reasoning, Transactions on Computational Science, II/2, (2008), 85-98.

[9] ARMSTRONG, D., Universals and Scientific Realism, Cambridge University Press, (1978)

[10] KOURIE, D., An Approach to Defining Abstractions, Refinements and Enrichments. Quaestiones Informaticae 6/4,pp.(1989), 174-178.

[11] BEDNARIK, L., KOVÁCS, L.: Implementation and assessment of the automatic question generation, Proc. of IEEE CogInfoCom (2012), 687-690.

[12] CONIAM, D.,A Preliminary Inquiry into Using Corpus Word Frequency Data in the Automatic Generation of English Language Cloze Tests, CALICO Journal 14, no.2-4, (1997), 15-33.

[13] KOVÁCS, L., SZEMÁN, G.: Complexity-based Generation of Multi-Choice Tests in AQG Systems, Proc. of CogInfoCom'2013, (2013), 399-402.