UDC 51.681.3

S. KRYVYI
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine,
e-mail: *sl.krivoi@gmail.com*.

H. HOHERCHAK
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine,
e-mail: *gogerchak.g@gmail.com*.

ANALYZING NATURAL-LANGUAGE KNOWLEDGE IN UNCERTAINTY ON THE BASIS OF DESCRIPTION LOGICS

Abstract. The article overviews the means for describing and formally analyzing natural-language text knowledge under uncertainty. We consider a family of classic attribute languages and logics based on them, their properties, problems, and solution tools. We also give an overview of propositional *n*-valued logics and fuzzy logics, their syntax, and semantics. Based on the considered logical constructions, we propose syntax and set-theoretic interpretation of *n*-valued description logic $ALCQ_n$ that provides means for describing concept intersection, union, complement, value restrictions, and qualitative and quantitative constraints. We consider the means for solving key problems of reasoning over such logics: executability, augmentation, equivalence, and disjunctivity. As an algorithm for calculating executability degree, we consider an extension of the tableau algorithm often used for first-order logic with solving simple numerical constraints. We prove that the algorithm is terminal, complete, and non-contradictory. We also provide several applications for the provided formal representation in natural language processing, including extending results of machine learning models, combining knowledge from multiple sources, and formally describing uncertain facts.

Keywords: knowledge bases, description logics, fuzzy logics, *n*-valued logics, natural language processing, knowledge extraction.

1. OVERVIEW OF DESCRIPTION LOGICS

1.1. Knowledge representation. Before proceeding to the description of the knowledge representation system, let us indicate some works related to the subject of this article. Note that there is a large number of works where logic is applied to the analysis of knowledge obtained from natural language texts. Thus, paper [1] presents the application of classical logics, and paper [2] presents fuzzy logics applications. Papers [3] and [4] use the methods of machine learning and computer linguistics. Unfortunately, the authors do not know of works in which multi-valued logics are applied to the analysis of knowledge obtained from natural language texts, and therefore the authors do not claim the primacy of such an application.

We introduce some key basics of the theory of knowledge bases and description logics that will be used below in this article, which is continued [5]. Concepts are a tool for recording knowledge about the subject area to which they apply. This knowledge is divided into general knowledge of concepts and their interconnections and knowledge of individual objects, their properties, belonging to concepts, and relations with other objects. According to this division, knowledge written using the language of description logic is divided into a set of terminal axioms named TBox and a set of facts about individuals named ABox.

Let $CN = \{A_1, A_2, \dots, A_n\}$ and $RN = \{R_1, \dots, R_m\}$ — finite non-empty sets of atomic concepts (concept names) and atomic roles (role names), respectively. Then the syntax of *AL*-language (attribute language) can be defined in the following way [6].

Definition 1. A set of concepts of an attribute language *AL* is defined by induction:

— symbols $\top$ (top, universal concept) and $\bot$ (bottom, empty concept) are concepts;

— each concept name $A_i \in CN$ is a concept;

— if A is a concept, then $\neg A$ (complement of A) is a concept;

— if C and D are concepts, then $C \sqcup D$ (intersection of concepts) is a concept;

— if C is a concept and R is an atomic role, then $\exists R \top$ (limited existence quantifier) and $\forall R.C$ (value limitation) are concepts;

— no other expressions are concepts.

Definition 2. A terminological axiom is an expression $C \sqsubseteq D$ (inclusion of a concept C into a concept D) or $C \equiv D$ (equivalence of concepts C and D), where C and D are arbitrary concepts. Terminology (TBox) is an arbitrary finite set of terminological axioms.

Let $IN = \{a_{1,} \ldots, a_k\}$ be a finite non-empty set of atomic individuals (individual names). Then we can also define knowledge about certain individuals and their relationships with facts of two types:

— $a : C$ states that individual a belongs to the concept C;

— aRb states that individuals a and b are related with a role R.

Definition 3. Assertion box (ABox) is an arbitrary finite set of facts.

Semantics of *AL*-based logic (or simply *AL*-logic) can be defined with set theory interpretation as follows [7].

Definition 4. Interpretation is a pair $I = (\Delta,^I)$ that consists of a non-empty set Δ named domain and interpretation function ^{I} that maps:

— to each atomic concept $A \in CN$ an arbitrary subset $A^I \subseteq \Delta$;

— to each atomic role $R \in RN$ an arbitrary subset $R^I \subseteq \Delta \times \Delta$.

Interpretation function is being spread on the whole set of *AL*-based logic concepts unambiguously:

$\top^I = \Delta$;

$\bot^I = \varnothing$;

$(\neg A)^I = \Delta \setminus A^I$;

$(C \sqcap D)^I = C^I \cap D^I$;

$(\exists R.\top)^I = \{a \in \Delta \mid \exists b \in \Delta : aR^I b\}$;

$(\forall R.C)^I = \{a \in \Delta \mid \forall b \in \Delta : aR^I b \to b \in C^I\}$.

In more powerful description logic types, terminology may also include axioms over roles that are defined respectively.

Definition 5. An axiom $C \sqsubseteq D$ $(C \equiv D)$ is true in interpretation I if $C^I \subseteq D^I$ $(C^I = D^I)$. In this case I is called a model of this axiom and write $I \models C \sqsubseteq D$. An interpretation I is called a model of terminology T $(I \models T)$ if it is a model for all axioms of the corresponding TBox. An interpretation I is called a model of assertion box A $(I \models A)$ if for each fact $a : C$ and aRb expressions $a^I \in A^I$ and $a^I R^I b^I$ are true respectively.

TBox (ABox) is called compatible or executable if it has a non-empty model. The concept C is satisfiable with respect to terminology T if there is a model I of terminology T such that $C^I \neq \varnothing$.

The terminology gives the opportunity to record general knowledge of concepts and roles. But it often insinuates the need to record knowledge about specific individuals: what class the individual belongs to, what relationships (roles) they relate to each other.

Below we will return to the concept of executability of terminology and concept and provide the tableau algorithm for this problem [8].

AL-logic described above can be extended by adding new concept constructors to its syntax definition.

ALCQ-logic can be built on *AL*-logic with addition of rules as follows:

— if C and D are concepts, then $C \sqcup D$ (union of concepts) is a concept;

— if C is a concept, then $\exists R.C$ (full existence quantifier) is a concept;

— if R is an atomic role and n is a natural number then $\leq nR$ (quantitate restriction) is a concept;

— if C is a concept, then $\neg C$ (complement of an arbitrary concept) is a concept;

— if C is a concept, R is an atomic role and n is a natural number then $\geq nR.C$ (qualitative restriction) is a concept.

Semantics of these kinds of concepts and roles are the following:

$(C \sqcup D)^I = C^I \cup D^I$;

$(\neg C)^I = \Delta \setminus C^I$;

$(\exists R.C)^I = \{a \in \Delta \mid \exists b(aR^I b \wedge b \in C^I)\}$;

$(\leq nR)^I = \{a \in \Delta \mid |\{b \mid aR^I b\}| \leq n\}$;

$(\leq nR.C)^I = \{e \in \Delta \mid |\{d \mid eR^I d \wedge d \in C^I\}| \leq n\}$.

Naturally, we can define the following notions:

$\geq nR.C = (\neg(\leq (n-1)R.C)$;

$= nR.C = (\leq nR.C) \sqcap (\geq nR.C)$;

$< nr.C = (\leq nR.C) \sqcap \neg(= nR.C)$;

$> nR.C = (\geq nR.C) \sqcap \neg(= nR.C)$.

Also, we can define $\exists R.C = (\geq 1R.C)$ and $\leq nR = \leq nR.\top$.

1.2. Key knowledge inference problems. Usage of knowledge bases is tightly connected with checking if its structure satisfies some specific conditions and properties. For example, it is useful to have an efficient way to check if the newly added concept has some meaning within existing knowledge, is equivalent to some already existing concept or is disjunct with it.

Let us define some key problems that are related to inference from knowledge systems:

— executability of a concept: concept C is executable in a terminology T if there exists model I of terminology T such that C^I is not empty;

— absorption of a concept: concept C is absorbed by concept D in a terminology T $(C \sqsubseteq_I D)$ if for each model I of terminology T $C^I \subseteq D^I$;

— equivalency of concepts: concepts C and D are equivalent in a terminology T $(C_T \equiv D)$ if for each model I of terminology T $C^I = D^I$;

— disjunctivity (mutual exclusion) of concepts: concepts C and D are disjunctive in a terminology T if for each model I of terminology T $C^I \cap D^I = \varnothing$.

Problems of absorption, equivalency and disjunctivity can be reduced to the problem of concept executability. With that said, we will mainly consider the latter problem within this article, without limiting the generality of considered approaches.

Various kinds of description logics provide methods for solving the executability problem by applying so-called tableau algorithm. Tableau algorithm for *ALCQ*-logic with examples and proof is considered in [6] and [7].

2. PROPOSITIONAL *n*-VALUED LOGIC

Syntax and semantics of *n*-valued propositional logic L_n differs from classic propositional logic in the amount of different logical values, that any expression can hold. So, L_n instead of two logical values 0 (false) and 1 (true) for L_2 utilizes n values of kind $i/(n-1)$, where $i \in \{0, 1, 2, \dots, n-1\}$. Interpretation of values 0 and 1 remains as false and true respectively, but new possible values describe the degree of truth between 0 and 1 [9, 10].

For the sake of conciseness, let us define LV_n to be the set of possible logical values for an n-valued logic:

$$LV_n = \left\{ \frac{i}{n-1} \,\middle|\, i \in \{0, 1, 2, \dots, n-1\} \right\}.$$

Instead of the single way of interpretation for L_2-logic simple operations, L_n-logic in general case may have various kinds depending on semantics of basic logical operations. Let us consider general rules of constructing such logics and several examples of L_n-logics as well.

Let $h(A)$ be the logical value of expression A. Then:

$h(\neg A) = 1 - h(A)$;

$h(A \wedge B) = t(h(A), h(B))$.

Here t is usually called a t-norm and satisfies the following properties:

$t(x, y) = t(y, x)$;

$t(x, t(y, z)) = t(t(x, y), z))$;

$x \le u \wedge y \le \upsilon \to t(x, y) \le t(u, \upsilon)$;

$t(x, 1) = x$.

Most L_n-logics use min–max algebra to form a t-norm function. For example, Kleene, Priest and Lukasiewicz logics use the following interpretation of conjunction:

$$h(A \wedge B) = \min\,(h(A), h(B)).$$

Disjunction can be defined by applying the De Morgan’s law as follows:

$$h(A \wedge B) = h(\neg(\neg A \wedge \neg B)) = \min\,(1 - h(A), 1 - h(B)) = \max\,(h(A), h(B)).$$

Implication in Kleene and Priest logics is defined similarly to the classic propositional logic:

$$h(A \to B) = h(\neg A \wedge B) = \max\,(1 - h(A), h(B)).$$

However, Lukasiewicz logic [10] treats this operation a bit differently:

$$h(A \to B) = \min\,(1, 1 - h(A) + h(B)).$$

3. DESCRIPTION OF *n*-GRADED SETS

Note, that n-graded and fuzzy logics usually are being used in a strong connection with somewhat related concept, namely n-graded and fuzzy set.

Definition 6. n-graded set is a pair $A = (U_A, \mu_A)$, where U_A is a set and $\mu_A : U_A \to LV_n$ is a membership function.

Like n-valued logics, n-graded sets have middle states between true and false — here it relates to the item being the element of the set. When $\mu_A(x) = 1$ element x is fully included into the set A, when $\mu_A(x) = 0$ it is not included, and in other cases it is partially included into the set A.

n-graded sets, like their classic equivalents, are empowered with a wide range of operations over them. Let us consider membership functions for the basic ones:

— complement: $\mu_{\neg A}(x) = 1 - \mu_A(x)$;

— intersection: $\mu_{A \cap B}(x) = t(\mu_A(x), \mu_B(x))$;

— union: $\mu_{A \cup B}(x) = s(\mu_A(x), \mu_B(x))$.

Above t is t-norm as defined in the previous section. Function s is called t-conorm and is defined in the following way: $s(x, y) = 1 - t(1 - x, 1 - y)$.

Usually, $t(x, y) = \min(x, y), s(x, y) = \max(x, y)$.

Definition 7. Two n-graded sets are disjoint iff $A \cap B = \varnothing$. Here, n-graded set A is included in an n-graded set B $A \subseteq B$ iff $\forall x \in U_A \cup U_B : \mu_A(x) \le \mu_B(x)$.

Value $\mu_{A \subseteq B} = \wedge_{x \in U_A \cup U_B} (\mu_A(x) \to \mu_B(x))$ is called a degree of inclusion of an n-graded set A in n-graded set B.

4. DESCRIPTION OF *n*-VALUED *ALCQ*-LOGIC

We will consider a combination of two types of logics that were presented above: *ALCQ*-logic and n-valued Lukasiewicz logic — to construct an apparatus for reasoning about knowledge in uncertainty. Let us refer to this as to $ALCQ_n$-logic.

Let the set of concepts and roles be defined as for the classic *ALCQ*-logic considered above.

Below we will continue referring to $h(A)$ as to the logical value of expression A in the sense of n-valued Lukasiewicz logic. It is worth mentioning, that in terms of *ALCQ*-logic we have 4 types of logical expression: concept inclusion ($C \sqsubseteq D$), concept equivalence ($C \equiv D$), belonging of the individual to the concept ($a : C$), and relation between two individuals within a role (aRb).

Definition 8. Terminology (TBox) of an n-valued knowledge base is a function h_t, that maps each terminological axiom to its logical value.

Example 1. Let $CN = \{animal, cat, dog, mouse, cute\}$, $RN = \{eat\}$.

We can define the following terminology in $ALCQ_3$-logic:

$$TBox = h_t = \{(cat \sqsubseteq animal, 1), (dog \sqsubseteq animal, 1), (mouse \sqsubseteq animal, 1),$$
$$(cat \sqsubseteq cute, 2/3), (mouse \sqsubseteq cute, 1/3),$$
$$(dog \sqcap \exists\, eat.cat \sqsubseteq cute, 1/3), (dog \sqcap \neg\exists\, eat.cat \sqsubseteq cute, 2/3)\}.$$

This terminology holds the information about several types of animals: dogs, cats, and mice, and their likeability. By the above-mentioned knowledge, cats are rather cute, mice are not so cute, and likeability of dogs depends on whether they eat cats or not: if yes, then they are not so cute, otherwise they are rather cute. In the n-valued description logic an interpretation of a concept is different from the classic one. Instead of classic set, it is interpreted as an n-graded set.

Definition 9. Interpretation is a pair $I = (\Delta, {}^I)$ that consists of a non-empty set Δ named domain and interpretation function ^{I} that maps:

— to each atomic concept $A \in CN$ an arbitrary n-graded set with membership function $h_A^I : A \to L_n$;

— to each atomic role $R \in RN$ an arbitrary n-graded set of pairs with membership function $h_R^I : A \times A \to L_n$.

In simple words:

— in $ALCQ_n$ for each concept we provide a measure of each domain item belonging to the set of items of that kind; for each role we provide a measure of each pair of domain items to relate within that role.

Given that, we can provide the following interpretation for the $ALCQ_n$ concept operations:

$h_\top^I(a) = \mu_{\top^I}(a) = 1$;

$h_\bot^I(a) = \mu_{\bot^I}(a) = 0$;

$h_{\neg C}^I(a) = \mu_{\neg C^I}(a) = 1 - h_C^I(a)$;

$h^I_{C\sqcap D}(a)=\mu_{C^I\cap D^I}(a)=\min(h^I_C(a),h^I_D(a))$;
$h^I_{(C\sqcup D)}(a)=\mu_{C^I\cup D^I}(a)=\max(h^I_C(a),h^I_D(a))$;
$h^I_{\forall R.C}(a)=\mu_{(\forall R.C)^I}(a)=\min(1,\min_{b\in\Delta}(1-h^I_R(a,b)+h^I_C(b))$;
$h^I_{\exists R.C}(a)=\mu_{(\exists R.C)^I}(a)=h(\exists\, b(aR^I b\wedge b\in C^I))=\max_{b\in\Delta}(\min(h^I_R(a,b),h^I_C(b))$;
$h^I_{\leq nR.C}(a)=\mu_{(\leq nR.C)^I}(a)=1-\max_{\{b_1,\ldots,b_{n+1}\}\sqsubseteq\Delta}\min_{1\leq i\leq n+1}(h^I_R(a,b_i),h^I_C(b_i))$;
$h^I_{\geq nR.C}(a)=h^I_{\neg(\leq(n-1)R.C)}(a)=\max_{\{b_1,\ldots,b_n\}\sqsubseteq\Delta}\min_{1\leq i\leq n}(h^I_R(a,b_i),h^I_C(b_i))$.

Definition 10. An axiom $C\sqsubseteq D$ ($C\equiv D$) has logical value v in interpretation I if $\mu_{C^I\subseteq D^I}=v$ ($\min(\mu_{C^I\subseteq D^I},\mu_{D^I\subseteq C^I})=v$). In this case I is called a model of this axiom and write $I\models C\sqsubseteq D$. An interpretation I is called a model of terminology T ($I\models T$) if it is a model for all axioms of the corresponding TBox. An interpretation I is called a model of assertion box A ($I\models A$) if for each factual expression $a:C$ and aRb $h_F(a:C)=h^I_C(a)$, and $h_F(aRb)=h^I_R(a,b)$.

5. REASONING OVER $ALCQ_n$-LOGIC

We considered above the key reasoning problems for classic *ALCQ*-logic terminologies, namely executability, absorption, equivalency and disjunctivity.

Their definition remains the same except for classic set is being replaced by *n*-graded set. However, sometimes problems that are aimed at acquiring the degree of the corresponding properties are more informative:

— degree of executability: $h_e(C)=1-h(C\sqsubseteq\bot)$;
— degree of absorption: $h_{abs}(C,D)=h(C\sqsubseteq D)$;
— degree of equivalency: $h_{equiv}(C,D)=h(C\equiv D)$;
— degree of disjunctivity: $h_{disj}(C,D)=h(C\sqcap D\sqsubseteq\bot)$;

Theorem 1. The following statements are true:
$h_e(C)=1-h_{abs}(C,\bot)$;
$h_{equiv}(C,D)=\min(h_{abs}(C,D),h_{abs}(D,C))$;
$h_{disj}(C,D)=h_{abs}(C\sqcup D,\bot)$;
$h_{abs}(C,D)=1-h_{exec}(C\sqcap\neg D)$;
$h_{equiv}(C,D)=1-\max(h_{exec}(C\sqcap\neg D),h_{exec}(D\sqcap\neg C))$;
$h_{disj}(C,D)=1-h_{exec}(C\sqcap D)$;
$C\sqsubseteq_T D\Leftrightarrow h_{abs}(C,D)=1$;
$C\equiv_T D\Leftrightarrow h_{equiv}(C,D)=1$;
C-executable $\Leftrightarrow h_{exec}(C)=1$;
C,D-disjunctive $\Leftrightarrow h_{disj}(C,D)=1$.

Proof is an obvious conclusion of set theory properties and definitions given above.

Theorem 1 shows that all the main problems of reasoning over terminology can be reduced to the problem of evaluating the executability degree. Below we will concentrate on the means of solving this problem.

We will refer below to v-executability as to the case when $h_e(C)\geq v$.

5.1. Tableau algorithm. Let us consider $ALCQ_n$-based knowledge base consisting of terminology T and facts A. This produces a set of constraints of the following form:

$$S=\{x:C\geq v\mid(x:C,v)\in A\}\cup\{xRy\geq v\mid(xRy,v)\in A\}\cup$$
$$\cup\{\top\sqsubseteq\neg C\sqcup D\geq v\mid(C\sqsubseteq D,v)\in T\}\cup$$
$$\cup\{\top\sqsubseteq(\neg(C\sqcup D)\sqcap(\neg D\sqcup C)\geq v)\mid(C\equiv D,v)\in T\}.$$

Let us formulate an algorithm to solve such a set of constraints. Based on the above-described semantics of the $ALCQ_n$ description logic formulas, we can provide a set of rules that constructs derived constraints and simplifies the overall structure of defined knowledge.

These rules have the similar nature as the rules of derivation for the classic first order logic, so this algorithm is often being referred to as a tableau algorithm.

These rules transform the initial set of constraints S to a system of independent irreducible constraint sets $\{S_i\}$.

Constraint set S_i contains a contradiction if and only if it contains:

— two constraints of the following kind: $x:C \geq v, x:C \leq u$, where $v > u$;

— two constraints of the following kind: $xRy \geq v, xRy \leq u$, where $v > u$;

— constraint $x:\bot \geq v$, where $v > 0$;

— constraint $x:\top \leq v$, where $v < 1$.

Rules of tableau algorithm for $ALCQ_n$ can be defined as follows:

$$\neg_{\geq}: \frac{x:\neg C \geq v}{x:C \leq 1-v}, \; \neg_{\leq}: \frac{x:\neg C \leq v}{x:C \geq 1-v},$$

$$\sqcap_{\geq}: \frac{x:C \sqcap D \geq v}{x:C \geq v, x:D \geq v}, \; \sqcap_{\leq}: \frac{x:C \sqcap D \leq v}{x:C \leq v \,|\, x:D \leq v},$$

$$\sqcup_{\geq}: \frac{x:C \sqcup D \geq v}{x:C \geq v \,|\, x:D \geq v}, \; \sqcup_{\leq}: \frac{x:C \sqcup D \leq v}{x:C \leq v \,|\, x:D \leq v},$$

$$\forall_{\geq}: \frac{x:\forall R.C \geq v, xRy \geq u, u > 1-v}{x:\forall R.C \geq v, xRy \geq u, \; y:C \geq v+u-1}, \; \forall_{\leq}: \frac{x:\forall R.C \leq v}{y:C = u, xRy \geq 1+u-v},$$

$$\geq n_{\geq}: \frac{x:\geq nR.C \geq v}{xRy_i \geq v, \; y_i:C \geq v, \; y_i \neq y_j},$$

$$\geq n_{\leq}: \frac{x:\geq nR.C \leq v, \{y_1, \ldots, y_n\} \subseteq IN, \; y_i \neq y_j \in A, xRy_i \geq u_i, \min u_i > v}{x:\geq nR.C \leq v, xRy_i \geq u_i, \; y_i:C \leq v},$$

$$\leq n_{\leq}: \frac{x:\leq nR.C \geq v}{x:\geq (n+1)R.C \leq 1-v}, \; \leq n_{\geq}: \frac{x:\leq nR.C \leq v}{x:\geq (n+1)R.C \geq 1-v},$$

$$T_{\geq}: \frac{y \in IN, \top \sqsubseteq E \geq v}{x:E \geq v}.$$

We should note that all rules but $\forall_{\leq}$ and $\geq n_{\leq}$ remove the constraint that match the premise from the constraint set. Also, none of the rules are applied twice to the same set of constraints considered as a premise.

Here and below, we will refer to a set of constraints that has a solution as executable. We will also refer to a set of constraints, for which none rules are applicable, as final.

As for the classic $ALCQ$-logic, $T_{\leq}$-rule in combination with $\geq n_{\leq}$-rule may produce infinite cycles of producing individuals of the same types. To resolve that we need to improve an idea of blocking between individuals for using in $ALCQ_n$-logic. For classic logic, the child individual was considered blocked if a set of concepts that he belongs to is a subset of the corresponding set for the parent at a certain point of time in the processing of the algorithm.

Definition 11. Individual x is blocking the individual y iff x or its child has produced y in result of $\geq n_{\leq}$-rule and $\forall x, y, C(x:C \geq v \in S \rightarrow y:C \geq u \in S) \wedge$ $\wedge (x:C \leq u \in S \rightarrow y:C \leq v \in S)$, where $u \geq v$. We will show later that this condition applied to the tableau algorithm described above makes it finite.

Let us consider an example of solving the constraint set with the help of this algorithm.

Example 2. Let us consider the following terminology over $ALCQ_3$-logic:

$$\text{TBox} = h_t = \{(cat \sqsubseteq eat.mouse, 2/3), (cat \cap \exists eat.cat \sqsubseteq \bot 1)\} .$$

We can construct the following set of constraints from this terminology:

$$S = \{\top \sqsubseteq \neg cat \sqcup \geq 1\, eat.mouse \geq 2/3, \top \sqsubseteq \neg(cat \sqcap \geq 1\, eat.cat) \geq 1\} .$$

Let us check whether our current terminology allows an individual to be both cat and mouse simultaneously. For this, let us add new condition to the constraint set:

$$S' = S \cup \{x : cat \sqcap mouse \geq \upsilon\} .$$

Let us construct a derivation tree following the rules of the tableau algorithm:

1. $(S')\, x : cat \sqcap mouse \geq \upsilon$.
2. $(T_{\geq})x : \neg cat \sqcup \geq 1\, eat.mouse \geq 2/3$.
3. $(T_{\geq})x : \neg(cat \sqcap \geq 1\, eat.cat) \geq 1$.
4. $(\sqcap_{\geq}, 1)x : cat \geq \upsilon, x : mouse \geq \upsilon$.
5. $(\neg_{\geq}, 3)x : cat \sqcap x : \geq 1\, eat.cat \geq 0$.
6. $(\sqcap_{\leq}, 5)S_1 | S_2$,

$$S_1 = \{x : \neg cat \sqcup \geq 1\, eat.mouse \geq 2/3, x : cat \geq \upsilon, x : mouse \geq \upsilon, x : cat \leq 0\} ,$$

S_1 has solution iff $\upsilon = 0$, which means that the initial fact is impossible in this branch,

$$S_2 = \{x : \neg cat \sqcup 1\, eat.mouse \geq 2/3, x : cat \geq \upsilon, x : mouse \geq \upsilon, x : \geq 1\, eat.cat \leq 0\} .$$

7. $(S_2)x : \geq 1\, eat.cat \leq 0$.
8. $(\sqcup_{\geq}, 2)S_{21} | S_{22}$,

$$S_{21} = \{x : \neg cat \geq 2/3, x : cat \geq \upsilon, x : mouse \geq \upsilon, x : \geq 1\, eat.cat \leq 0\} .$$

9. $(S_{21})x : \neg cat \geq 2/3$.
10. $(\neg_{\geq}, 9)x : cat \leq 1/3$.

No rules can be applied anymore. S_{21} has solution iff $\upsilon \leq 1/3$,

$$S_{22} = \{x : \geq 1\, eat.mouse \geq 2/3, x : cat \geq \upsilon, x : mouse \geq \upsilon, x : \geq 1\, eat.cat \leq 0\} .$$

8′. $(S_{21})x : \geq 1\, eat.mouse \geq 2/3$.
9′. $(\geq n_{\geq}, 9)x$ **eat** $y \geq 2/3$, $y : mouse \geq 2/3$.
10′. $(T_{\geq})\, y : \neg cat \sqcup \geq 1\, eat.mouse \geq 2/3$.
11. $(T_{\geq})\, y : \neg(cat \sqcap \geq 1\, eat.cat) \geq 1$.
12. $(\geq n_{\leq}, 7, 10)\, y : cat \leq 0$.
13. $(\neg_{\geq}, 12)\, y : cat \sqcap x : \geq 1\, eat.cat \leq 0$ (is absorbed by 13).
14. $(\sqcup_{\geq}, 11)S_{221} | S_{222}$,

$$S_{221} = \{x \textbf{ eat } y \geq 2/3,\ y : mouse \geq 2/3, x : cat \geq \upsilon, x : mouse \geq \upsilon,$$
$$x : \geq 1\, eat.cat \leq 0,\ y : \neg cat \geq 2/3,\ y : cat \leq 0\}.$$

15. $(S_{221})\, y : \neg cat \geq 2/3$.
16. $(\neg_{\geq}, 16)\, y : cat \leq 1/3$.

S_{221} has solution for any υ, which means that the initial fact is possible with any degree and we can stop the process of the algorithm.

As a result of the algorithm, we also got the set of constraints under which the initial premise is possible:

$$S = \{x \textbf{ eat } y \geq 2/3,\ y : mouse \geq 2/3, x : mouse \geq \upsilon, x : \geq 1\, eat.cat \leq 0,\ y : cat \leq 0\} .$$

In human language this means that our selected individual cat x be cat and mouse if it eats something likely to be a mouse but not a cat and eats no cats. So, for more derivation power we need to explicitly mention in the terminology that no individuals can be cat and mouse at the same time by adding an axiom $(cat \sqcap mouse \sqsubseteq \bot, 1)$.

Lemma 1. There does not exist an infinite sequence $S_0, S_1, \ldots$ in which each next set of constraints S_{i+1} is derived from S_i by some rule of the above-mentioned algorithm.

Proof. Let us consider a tree with sets of constraints as leaves, S_0 as a root and with edges defining that one set was derived from another by rules of an algorithm. Then the power of each vertice is limited by $\geq n_{\leq}$-rule.

All rules but $\forall_{\geq}$ and $\geq_{\leq}$ remove the constraint that match the premise from the constraint set. Both these rules can be applied no more than twice on the individuals that are ancestors of the same individual before it reaches blocked state that will prevent it to get new such concept constraints. Thus, the length of each sequence in such tree from the root to the leave will be limited. ■

Lemma 2. The following statements are true.

Let S' be derived from S by applying one of the rules

$$\neg_{\geq},\ \neg_{\leq},\ \sqcap_{\geq},\ \sqcup_{\leq},\ \forall_{\geq},\ \forall_{\leq},\ \geq n_{\geq},\ \leq n_{\geq},\ \leq n_{\leq},\ T_{\geq}.$$

Then S is executable with regards to T iff S' is also executable with regards to T.

Let S', S'' be derived from S by applying one of the rules $\sqcap_{\leq}$, $\sqcup_{\geq}$. Then S is executable with regards to T iff either S' or S'' is also executable with respect to T.

Let $\{S^i\}$ be derived from S by applying $\geq n_{\leq}$-rule. Then S is executable with regards to T iff either of S^i is also executable with regards to T.

Proof. Let us consider each individual rule.

— Let S' be derived from S by applying $\neg_{\geq}$-rule. If S is executable, then exists its model I. By the premise of the rule: $h_{\neg C}(x^I) \geq v$. Thus, $h_C^I(x^I) = 1 - h_C(x^I) \leq 1 - v$. Thus, I is a model of S'. The opposite is obvious.

— Let S' be derived from S by applying $\neg_{\leq}$-rule. If S is executable, then exists its model I. By the premise of the rule: $h_{\neg C}(x^I) \leq v$. Thus, $h_C^I(x^I) = 1 - h_C(x^I) \geq$ $\geq 1 - v$. Thus, I is a model of S'. The opposite is obvious.

— Let S' be derived from S by applying $\sqcap_{\geq}$-rule. If S is executable, then exists its model I. By the premise of the rule: $h_{C \sqcap D}(x^I) \geq v$. By interpretation, $h_{C \sqcap D}(x^I) = \min(h_C(x^I), h_D(x^I))$. Thus, $h_C^I(x^I) \geq v$ and $h_D^I(x^I) \geq v$. Thus I is a model of S'. The opposite is obvious.

— Let S', S'' be derived from S by applying $\sqcap_{\leq}$-rule. If S is executable, then exists its model I. By the premise of the rule: $h_{C \sqcap D}(x^I) \leq v$. By interpretation, $h_{C \sqcap D}(x^I) = \min(h_C(x^I), h_D(x^I))$. Thus, $h_C^I(x^I) \leq v$ or $h_D^I(x^I) \leq v$. Thus I is a model of S' or S'' respectively. The opposite is obvious.

— Let S', S'' be derived from S by applying $\sqcup_{\geq}$-rule. If S is executable, then exists its model I. By the premise of the rule: $h_{C \sqcup D}(x^I) \geq v$. By interpretation, $h_{C \sqcup D}(x^I) = \max(h_C(x^I), h_D(x^I))$. Thus, $h_C^I(x^I) \geq v$ or $h_D^I(x^I) \geq v$. Thus, I is a model of S' or S'' respectively. The opposite is obvious.

— Let S' be derived from S by applying $\sqcup_{\leq}$-rule. If S is executable, then exists its model I. By the premise of the rule: $h_{C \sqcup D}(x^I) \leq v$. By interpretation, $h_{C \sqcup D}(x^I) = \max(h_C(x^I), h_D(x^I))$. Thus, $h_C^I(x^I) \leq v$ and $h_D^I(x^I) \leq v$. Thus, I is a model of S'. The opposite is obvious.

— Let S' be derived from S by applying $\forall_{\geq}$-rule. If S is executable, then exists its model I. By the premise of the rule: $h^I_{\forall R.C}(x^I) \geq \upsilon$ and $h^I_R(x^I, y^I) \geq u$. By interpretation, $h_{\forall R.C}(x^I) = \min(1, \min_{y^I \in \Delta} 1 - h^I_R(x^I, y^I) + h^I_C(y^I)$. Thus,

$$\min_{y^I \in \Delta} 1 - h^I_R(x^I, y^I) + h^I_C(y^I) \geq \upsilon \to 1 - h^I_R(x^I, y^I) + h^I_C(y^I) \geq \upsilon,$$

$$h^I_C(y^I) \geq \upsilon - 1 + h^I_R(x^I, y^I) \geq \upsilon + u - 1.$$

Thus, I is a model of S'. The opposite is obvious because the rule does not remove any constraints.

— Let S' be derived from S by applying $\forall_{\leq}$-rule. If S is executable, then exists its model I. By the premise of the rule: $h^I_{\forall R.C}(x^I) \leq \upsilon$. By interpretation, $h_{\forall R.C}(x^I) = \min(1, \min_{y^I \in \Delta} 1 - h^I_R(x^I, y^I) + h^I_C(y^I))$. Thus, exists y^I such that $1 - h^I_R(x^I, y^I) + h^I_C(y^I) \leq \upsilon$. Let $h^I_C(y^I) = u$. Then $h^I_R(x^I, y^I) \geq 1 + u - \upsilon$. Thus, I is a model of S'. The opposite is obvious.

— Let S' be derived from S by applying $\geq n_{\geq}$-rule. If S is executable, then exists its model I. By the premise of the rule: $h^I_{\geq nR.C}(x^I) \geq \upsilon$. By interpretation, $h_{\geq nR.C}(x^I) =$ $= \max_{\{y^I_1, \ldots, y^I_n\} \subseteq \Delta} \min_{1 \leq i \leq n}(h^I_R(x^I, y^I_i), h^I_C(y^I_i))$. Thus, exists $\{y^I_1, \ldots, y^I_n\} \subseteq \Delta$ such that: $\min_{1 \leq i \leq n}(h^I_R(x^I, y^I_i), h^I_C(y^I_i)) \geq \upsilon$. This means that $h^I_R(x^I, y^I_i) \geq \upsilon$ and $h^I_C(y^I_i) \geq \upsilon$. Thus, I is a model of S'. The opposite is obvious.

— Let S^i be derived from S by applying $\geq n_{\leq}$-rule. If S is executable, then exists its model I. By the premise of the rule: $h^I_{\geq nR.C}(x^I) \leq \upsilon$, $xRy_i \geq u_i$, $\min_i u_i > \upsilon$. By interpretation, $h_{\geq nR.C}(x^I) = \max_{\{y^I_1, \ldots, y^I_n\} \subseteq \Delta} \min_{1 \leq i \leq n}(h^I_R(x^I, y^I_i), h^I_C(y^I_i))$. Thus, for each $\{y^I_1, \ldots, y^I_n\} \subseteq \Delta$ $\min_{1 \leq i \leq n}(h^I_R(x^I, y^I_i), h^I_C(y^I_i)) \leq \upsilon$. As $xRy_i \geq u_i > \upsilon$, then $\min_{1 \leq i \leq n}(h^I_C(y^I_i)) \leq \upsilon$. This means that exists i such that $h^I_C(y^I_i) \leq \upsilon$ and I is a model of corresponding S^i. The opposite is obvious because the rule does not remove any constraints.

— Let S' be derived from S by applying $\leq n_{\geq}$-rule. If S is executable, then exists its model I. By the premise of the rule: $h^I_{\leq nR.C}(x^I) \geq \upsilon$. Thus, $h^I_{\leq nR.C}(x^I) =$ $= 1 - h^I_{\geq(n+1)R.C}(x^I) \leq 1 - \upsilon$. Thus, I is a model of S'. The opposite is obvious.

— Let S' be derived from S by applying $\leq n_{\leq}$-rule. If S is executable, then exists its model I. By the premise of the rule: $h^I_{\leq nR.C}(x^I) \leq \upsilon$. Thus, $h^I_{\leq nR.C}(x^I) =$ $= 1 - h^I_{\geq(n+1)R.C}(x^I) \geq 1 - \upsilon$. Thus, I is a model of S'. The opposite is obvious.

— Let S' be derived from S by applying $T_{\geq}$-rule. If S is executable, then exists its model I. By the premise of the rule: $h^I_E(x^I) \geq \upsilon$. Thus, I is a model of S'. The opposite is obvious because the rule does not remove any constraints. ■

Lemma 3. Non-contradictory final set of constraints is executable.

Proof. Let S be final non-contradictory set of constraints derived from S_0. Let $h(x:C)$ and $h(xRy)$ be the minimum degree of $x:C$ and xRy, respectively, that complies with all constraints of S. Let I be as follows:

$$\Delta^I = \{x : x \text{ — } active\},$$

$$h_C^I(x^I) = \begin{cases} \{h(x:C), \ x \text{ — } active, \\ h(z:C), \quad z \ blocks \ x \wedge z \text{ — } active, \end{cases}$$

$$h_R^I(x^I, y^I) = \begin{cases} h(xRy), \ x, y \text{ — } active, \\ h(zRy), \ z \ blocks \ x \wedge z, y \text{ — } active, \\ h(xRw), \ w \ blocks \ y \wedge w, x \text{ — } active, \\ h(zRw), \ z \ blocks \ x \wedge w \ blocks \ y \wedge z, w \text{ — } active. \end{cases}$$

T-rule cannot be applied, so $h_E^I(x^I) \geq v$ or exists active z that blocks x. In the second case $h_E^I(z^I) \geq v$ and by construction $h_E^I(x^I) = h_E^I(z^I) \geq v$. I is a model of T.

Let $xRy \geq v \in S$. Then by construction $h_R^I(x^I, y^I) \geq v$. We need to prove that $\forall x^I \in \Delta^I : x : C \geq v \in S \to h_C^I(x^I) \geq v$ by induction of construction of C. As S is final, it can only consist of the following constraints:

$$x : A \leq \geq v; \ xRy \leq \geq v; \ x : \leq nR.C \leq v; \ x : \forall R.C \geq v; \ \top \sqsubseteq E \geq v.$$

All other constraints in other case can be reduced by applying corresponding rules of the algorithm.

Base of induction is obvious from construction of I.

Induction step:

— Let $x : \forall R.C \geq v \in S$. As far as $\forall_{\geq}$-rule is inapplicable:

$$\forall y \ (xRy \geq u \in S \wedge u \leq 1 - v) \vee$$
$$\vee \ (xRy \geq u \in S \wedge u > 1 - v \wedge y : C \geq v + u - 1 \in S)).$$

Let x be active. Let us take $y^I \in \Delta^I$ such that $h_R^I(x^I, y^I) > 1 - v$.

Let y also be active. By construction, $h(xRy) > 1 - v$. Thus, there exists $u > 1 - v$ such that $xRy \geq u \in S$. Thus, $y : C \geq v + u - 1 \in S$ and $h_C^I(y^I) \geq v + u - 1$.

If $h_R^I(x^I, y^I) > 1 - v$ then $1 - h_R^I(x^I, y^I) + h_C^I(y^I) \geq -h_R^I(x^I, y^I) + v + u \geq v$.

If $h_R^I(x^I, y^I) \leq 1 - v$ then $1 - h_R^I(x^I, y^I) + h_C^I(y^I) \geq v + h_C^I(y^I) \geq v$.

Thus,

$$h_{\forall R.C}(x^I) = \min(1, \min_{y^I \in \Delta} 1 - h_R^I(x^I, y^I) + h_C^I(y^I)) \leq n.$$

Let y be blocked by active w. By construction, $h(xRw) > 1 - v$. Thus, there exists $u > 1 - v$ such that $xRw \geq u \in S$. Thus, $w : C \geq v + u - 1 \in S$ and $h_C^I(y^I) = h_C^I(w^I) \geq$ $\geq v + u - 1$. And the rest of proof is by analogy.

Let x be blocked by active z. Let us take $y^I \in \Delta^I$ such that $h_R^I(x^I, y^I) > 1 - v$.

Let y be active. By construction, $h(zRy) > 1 - v$. Thus, there exists $u > 1 - v$ such that $zRy \geq u \in S$. By definition of blocking $x : \forall R.C \geq v \to z : \forall R.C \geq v$. Thus, $y : C \geq v + u - 1 \in S$ and $h_C^I(y^I) = h_C^I(w^I) \geq v + u - 1$. And the rest of proof is by analogy.

Let y be blocked by active w. By construction, $h(zRw) > 1 - v$. Thus, there exists $u > 1 - v$ such that $zRw \geq u \in S$. By definition of blocking $x : \forall R.C \geq v \to$ $\to z : \forall R.C \geq v$. Thus, $w : C \geq v + u - 1 \in S$ and $h_C^I(y^I) = h_C^I(w^I) \geq v + u - 1$. And the rest of proof is by analogy.

Let $x : \geq nR.C \leq v \in S$.

Let exist n such $y_i^I : h_C^I(y_i^I) > \upsilon$ such that $h_R^I(x^I, y_i^I) > \upsilon$.

Let x, y_i be active. By construction, $h(xRy_i) > \upsilon$ and $h(y_i : C) > \upsilon$, which means that either premise of $\geq n_{\leq}$-rule is true or S has contradiction. Contradiction.

Let x be blocked by active z and y_i be active. By construction, $h(zRy_i) > \upsilon$ and $h(y_i : C) > \upsilon$. By definition of blocking $x : \geq nR.C \leq \upsilon \to z : \geq nR.C \leq \upsilon$. This means that either premise of $\geq n_{\leq}$-rule is true or S has contradiction. Contradiction.

Let x be blocked by active z and y_i be blocked by active w_i. By construction, $h(zRw_i) > \upsilon$ and $h(w_i : C) > \upsilon$. By definition of blocking $x : \geq nR.C \leq \upsilon \to$ $\to z : \geq nR.C \leq \upsilon$. This means that either premise of $\geq n_{\leq}$-rule is true or S has contradiction. Contradiction.

This means that either $h_C^I(y_i^I) \leq \upsilon$ or $h_R^I(x^I, y_i^I) \leq \upsilon$ for each such y_i^I. Hense,

$$h_{\leq R.C}(x^I) = \max_{\{y_1^I, \ldots, y_n^I\} \in \Delta} \min_{1 \leq i \leq n} (h_R^I(x^I, y_i^I), h_C^I(y_i^I) \leq \upsilon. \blacksquare$$

Theorem 2. Tableau algorithm solves the υ-executability problem for $ALCQ_n$.

Proof. From lemma 1 we can conclude that search tree has no infinite sequences and because degree of its branching is limited the search tree is finite. Thus, for any input data tableau-algorithm returns an answer in a finite time.

If S_0 is executable, then by lemma 2 at least one of final S is executable. By design of algorithm response in this case will be 1 (“yes”).

Let the response of the algorithm be 1 (“yes”). Then among its final sets of constraints exists S such that it is non-contradictive and final. By lemma 3 S is executable. Then by lemma 2 S_0 is executable. ■

6. NATURAL LANGUAGE PROCESSING APPLICATIONS

There are multiple approaches to the natural language knowledge analysis, that include usage of machine learning approaches, hand-written rules, or both. We will concentrate on the possible applications of the above-described formal system in the general process of knowledge extraction utilizing machine learning models or other probabilistic techniques.

We will consider the application of the *n*-valued and fuzzy description logics in cases of extending results of machine learning models, combining results from multiple sources, and formally describing uncertain facts.

6.1. Knowledge extraction in natural language processing. The task of Relation Extraction is to identify the relations between two entities in a sentence (often referred to as subject and object). Successful solution of this problem serves as a basis for many problems of natural language texts processing: questions answering, population of knowledge bases with natural language knowledge, etc. An example of the results of solving this problem is given in Fig. 1.

Due to the complexity of forming metrics to compare the results of relationship detection in the general case, the problem of detecting canonical relationships (mostly referred in the literature as a relation extraction) with the metric F1 is singled out, and the rest is related to the problem of open information extraction — constructing large knowledge bases based on information from the natural language texts.

At the time of writing, this problem does not have clearly defined and generally accepted standards of outcome, i.e., what relationships should be obtained and how they should be designed. In view of this, there is also no standard for evaluating models and bodies of acceptable size for quality training of ML-models, as is the case for many of the tasks described above in the field of natural language text processing.

Entity	←subject	Relation	object→	Entity
I		was visiting		Central Park

Fig. 1. Sample result of relation extraction

Open Information Extraction (OpenIE) is a problem for extraction of knowledge, represented by the natural language text, in the form of verbal phrases and their arguments. In contrast to Relation Extraction, OpenIE considers n-ary relations and some machine learning models also consider temporal and other features of the knowledge items.

In general, the proposed models for solving these problems are divided into two subtypes:

— machine learning systems (e.g., Neural Open Information Extraction and OpenIE-5.0);

— rule-based systems (e.g., Graphene).

6.2. Knowledge extraction result extension. Knowledge extraction systems that are based on machine learning approaches mostly produce as the result deterministic knowledge. This is dictated by their usage in several natural language processing tasks and their aiming at limited capacity of the evaluation metrics, that are based on the deterministic information.

By nature of neural networks, they still often operate with vague results within the system: machine learning model produces several potential results with confidence scores for each of them, or even is aimed at building the mapping of each result candidate to its confidence score. These preliminary results are then converted to deterministic ones by simple picking of the most confident option, thresholding or by applying more complex selection algorithms.

Being based on the corpus the model was trained and evaluated on makes the model highly dependent on the knowledge and techniques used in corpus and is quite biased towards it. This causes machine learning based system to give a higher score for the incorrect options and results not only in placing the incorrect information into result, but also ignoring the potential possibility of less confident results to be the correct ones.

For example, machine learning based system for Open Information Extraction OpenIE4 is known to often provide non-accurate confidence scores, especially for the sentences with pronouns, which is caused by drawbacks of initial corpus it was trained on. Thus, current state-of-the-art model SpanBERT uses all OpenIE4 extractions for its model training with an adapted loss function to gain better recall for such cases [11, 12].

Based on the above-described problems, more vague and versatile result of the machine learning model processing before applying determination algorithms can be used to apply further analysis of the result, its validity and compliance with some well-known knowledge items, already constructed knowledge etc. Analysis of such constructed knowledge can be performed utilizing the above-described mechanisms of n-ary and fuzzy description logics that provide an ability to check constructed terminology and facts for executability, i.e., validity, non-contradiction, and compliance with other knowledge.

Example 3. Let us consider the following natural language sentence:

A cafeteria is located on the sixth floor, a chapel on the 14th floor, and a study hall on the 15th floor.

Let the result of an OpenIE processing of the mentioned sentence provide the following n-ary relations: (a study hall; located; on the sixth floor), (a study hall; located; on the 14th floor), (a study hall; located; on the 15th floor)

We can conclude that the system parsed the grammatical relations within a sentence incorrectly, which produced the incorrect knowledge pieces to be produced. We can convert this to the *ALCQ*-logic as follows:

$$IN = \{a, b, c, d\},$$

$$CN = \{(Study\,Hall, 6th\ Floor, 14th\ Floor, 15th\ Floor, hall, floor)\},$$

$$RN = \{locate\},$$

$$TBox = \{(Study\,Hall \sqsubseteq hall), (6th\,Floor \sqsubseteq floor), (14th\,Floor \sqsubseteq$$

$$\sqsubseteq floor), (15th\,Floor \sqsubseteq floor)\},$$

$$Abox = \{(a \ \mathbf{locate}\ b,\ a\ \mathbf{locate}\ c,\ a\ \mathbf{locate}\ d,\ a : Study\,Hall,$$

$$b : 6th\,Floor,\ c : 14th\,Floor,\ d : 15th\,Floor)\}.$$

Let us populate the terminology with a well-known fact, that we could have derived from some aside knowledge source, that there is exactly one floor some hall can be located on:

$$TBox' = TBox \cup \{hall \sqsubseteq {=1}\ locate.\, floor\}.$$

Assuming b, c, and d are pairwise different individuals, the tableau-algorithm for *ALCQ* will result in such ABox to be non-executable. However, this does not mean that the initial sentence contains a contradiction.

Let us consider the result that model provided before its determinization:

(*a study hall; located; on the sixth floor; score* = *0.51), (a study hall; located; on the 14th floor; score* = *0.6), (a study hall; located; on the 15th floor; score* = *0.8), (a chapel; located; on the 14th floor; score* = *0.36), (a cafeteria; located; on the 14th floor; score* = *0.34), (a cafeteria; located; on the sixth floor; score* = *0.2*).

We can convert this to the $ALCQ_{\infty}$-logic as follows:

$$IN = \{a, b, c, d, e, f\},$$

$$CN = \{Study\,Hall, chapel, cafeteria, 6th\,Floor, 14th\,Floor, 15th\,Floor, hall, floor\},$$

$$RN = \{locate\},$$

$$TBox = \{(Study\,Hall \sqsubseteq hall), (chapel \sqsubseteq hall), (cafeteria \sqsubseteq hall),$$

$$(6th\,Floor \sqsubseteq floor), (14th\,Floor \sqsubseteq floor), (15th\,Floor \sqsubseteq floor),$$

$$(hall \sqsubseteq {=1}\ locate.\, floor, v)\}.$$

We can write slightly modified set of constraints as follows:

$$S = \{(a\ \mathbf{locate}\ b \geq v_1), (a\ \mathbf{locate}\ c \geq v_2), (a\ \mathbf{locate}\ d \geq v_3),$$
$$(e\ \mathbf{locate}\ c \geq v_4), (f\ \mathbf{locate}\ b \geq v_5), (f\ \mathbf{locate}\ c \geq v_6),$$
$$(a : StudyHall \geq 1), (b : 6th\,Floor \geq 1), (c : 14th\,Floor \geq 1),$$
$$(d : 15th\,Floor \geq 1), (e : chapel \geq 1), (f : cafeteria \geq 1)\}.$$

If we run the tableau-algorithm over such constraint set, we get such constraints over used degree variables:

$$((v_1 = 0 \wedge v_2 = 0) \wedge (v_2 = 0 \wedge v_3 = 0) \wedge (v_1 = 0 \wedge v_2 = 0)) \wedge (v_5 = 0 \wedge v_6 = 0).$$

Aiming at getting the largest overall confidence holding these constraints, we can maximize the following expression:

$$0.51v_1 + 0.6v_2 + 0.8v_3 + 0.36v_4 + 0.31v_5 + 0.2v_6,$$
$$v_1 = 0,\ v_2 = 0,\ v_3 = 1,\ v_4 = 1,\ v_5 = 1,\ v_6 = 0.$$

This allows us to exclude facts a **locate** b, a **locate** c, f **locate** c from consideration and resolve the contradiction they produce.

7. COMBINING KNOWLEDGE FROM VARIOUS SOURCES

The same approach can be used for combining knowledge acquired from various knowledge extraction algorithms or systems. Let (T_i, A_i) be classic deterministic knowledge bases. Then, they can be combined into an $ALCQ_n$-based knowledge base (T, A) as follows:

$$T = \left\{ \left(t, \frac{\Sigma_i h_i(t)}{n} \right) \middle| \, t \in \bigcup_i T_i \right\} ; \; A = \left\{ \left(f, \frac{\Sigma_i h_i(t)}{n} \right) \middle| \, f \in \bigcup_i A_i \right\} .$$

If the knowledge bases (T_i, A_i) do not contain a contradiction, but (T, A) does, we can eliminate it by solving an optimization problem defined in the previous section.

This technique provides an ability to combine ready knowledge bases. However, rule-based knowledge extraction systems can also get a significant gain from this approach.

Many rule-based tools utilize the output of several natural language processing tasks, including verbal event extraction, role labeling, boxing, modality detection, tense representation, named entities detection, dependency parsing, coreference resolution and machine translation. However, all these problems, often being solved by the means of machine learning, have their own issues, weak sides, and errors. This makes such a pipeline to combine all the errors from all its components and limits overall abilities. Such systems can get a significant gain from getting several models solving each task to come into play.

Instead of building several isolated knowledge extraction systems that are hard to operate with, we can construct vague preliminary structures that can be further used to extract the knowledge.

7.1. Operating with vague concepts. In contrast to the above-considered applications of vague description logics, there are also more direct use cases. There are various kinds of natural language expressions that have no certain meaning and are fuzzy.

Example 4. Let us consider the following text fragment:

"*Paul is a young boy from Australia. He smokes and wears expensive dress*."

Concepts "young" and "expensive" are uncertain and cannot be defined by a specific range of years that Paul might have lived for or a specific price for that dress. However, we can make some partial conclusions from Paul being an individual of that concept. For example, if person is young, then probably he or she is not likely to be 18:

$$(young \sqcap person \sqsubseteq 18^+ \; years, 1/3) .$$

Also, in Australia it is prohibited to smoke when you are not an adult. So, if we will consider a law-oriented knowledge base, then we can write

$$(person \sqcap \exists smoke. \top \sqsubseteq 18^+ \; years, 1) .$$

This allows us to say that with degree of 1 / 3 there is a contradiction, or a lawless action described in the fragment.

REFERENCES

1. Bos J. Applying automated deduction to natural language understanding. *Jour. of Applied Logic*. 2009. Vol. 7, N 1. P. 100–112. https://doi.org/10.1016/j.jal.2007.07.008.
2. Licata G. Fuzzy logic, knowledge and natural language. In: Fuzzy Inference System — Theory and Applications. Azeem M.F. (Ed.). IntechOpen, 2012. P. 3–18. https://doi.org/10.5772/36498.

3. Joshi M., Chen D., Liu Y., Weld D.S., Zettlemoyer L., Levy O. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics.* 2020. Vol. 8. P. 64–77. URL: https://transacl.org/index.php/tacl/article/view/1853.
4. Cambria E, White B. Jumping NLP curves: a review of natural language processing research. *IEEE Computational Intelligence Magazine.* 2014. Vol. 9, Iss. 2. P. 48–57. https://doi.org/10.1109/MCI.2014.2307227.
5. Kryvyi S., Hoherchak H. Analyzing natural language knowledge in uncertainty. *Proc. 2022 IEEE International Conference on Advanced Trends in Information Theory (ATIT 2022)* (15–17 December 2022, Kyiv, Ukraine). Kyiv, 2022. P. 268–272.
6. Hoherchak H., Darchuk N., Kryvyi S. Representation, analysis, and extraction of knowledge from unstructured natural language texts. *Cybernetics and Systems Analysis*. 2021. Vol. 57, N 3. P. 481–500. https://doi.org/10.1007/s10559-021-00373-7.
7. Hoherchak H. Knowledge based and description logics applications to natural language texts analysis. *Proc. 12th International Scientific and Practical Conference of Programming (UkrPROG 2020)* (September 15–16 2020, Kyiv, Ukraine). Kyiv, 2020. P. 259–269. URL: https://ceur-ws.org/Vol-2866/ceur_260-269hoherchak.pdf.
8. Baader F., Horrocks I., Latz C., Sattler U. An introduction in description logic. Cambridge University Press, 2017. 265 p. https://doi.org/10.1017/9781139025355.
9. Wajsberg M. Axiomatization of the three-valued propositional calculus. In: Polish Logic: 1920 – 1939. McCall S. (Ed.). New York: Oxford University Press, 1967. P. 264–284.
10. Aguzzoli S, Ciabattoni A. Finitness in infinite-valued Łukasiewich logic. *Journal of Logic, Language and Information*. 2000. Vol. 9, Iss. 1. P. 5–29. https://doi.org/10.1023/A:1008311022292.
11. Zhan J., Zhao H. Span model for open information extraction on accurate corpus. *Proc. the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)* (7–12 February 2020, New York, USA). New York, 2020. Vol. 34, N. 5. P. 9523–9530. https://doi.org/10.1609/aaai.v34i05.6497.
12. Pogorilyy S., Kramov A. Coreference resolution method using a convolutional neural network. *Proc. 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT 2019)* (18–20 December 2019, Kyiv, Ukraine). Kyiv, 2019. P. 397–401. https://doi.org/10.1109/ATIT49449.2019.9030596.

С.Л. Кривий, Г.І. Гогерчак

АНАЛІЗ ПРИРОДНОМОВНИХ ЗНАНЬ В УМОВАХ НЕВИЗНАЧЕНОСТІ НА ОСНОВІ ДЕСКРИПТИВНИХ ЛОГІК

Анотація. Представлено огляд засобів опису та формального аналізу знань, добутих з природномовного тексту з можливими невизначеностями. Розглянуто сім'ю класичних атрибутивних мов і логік, які на них ґрунтуються, властивості цих логік, проблеми і способи їхнього розв'язання. Представлено огляд пропозиційних *n*-значних логік і нечітких логік, їхнього синтаксису і семантики. На основі розглянутих конструкцій цих логік запропоновано синтаксис і теоретикомножинну інтерпретацію дескриптивної *n*-значної логіки $ALCQ_n$, яка описує властивості концептів за допомогою операцій перетину, об'єднання, доповнення та обмежених кванторів. Розглянуто засоби розв'язання ключових проблем для таких логік: виконáнність, розширення, еквівалентність та диз'юнктивність. Як алгоритм для обчислення ступеня виконуваності запропоновано застосувати розширений алгоритм семантичного табло, який використовують у логіці предикатів першого порядку для розв'язання простих числових обмежень. Доведено, що пропонований алгоритм є термінальним, повним і несуперечним. Наведено приклади застосування для формального представлення й оброблення природномовного тексту, які містять деякі результати моделей машинного навчання, комбінування знань з багатьох джерел і формальний опис сумнівних фактів.

Ключові слова: база знань, дескриптивні логіки, нечіткі логіки, *n*-значні логіки, оброблення природномовних текстів, добування знань.

Надійшла до редакції 10.03.2023