[Future Computing and Informatics Journal](https://digitalcommons.aaru.edu.jo/fcij)

Volume 7
Issue 1 (2022) *Issue 1*

Article 5

2022

Fuzzy Reasoning Procedure for Ontologies based on Rough Membership Approximation

Armand Florentin Donfack Kana
Ahmadu Bello University, donfackkana@gmail.com

Babatunde Opeoluwa Akinkunmi
University of Ibadan, bo.akinkunmi@ui.edu.ng

Recommended Citation

Donfack Kana, Armand Florentin and Akinkunmi, Babatunde Opeoluwa (2022) "Fuzzy Reasoning Procedure for Ontologies based on Rough Membership Approximation," *Future Computing and Informatics Journal*: Vol. 7: Iss. 1, Article 5.
DOI: https://doi.org/10.54623/fue.fcij.7.1.5
Available at: https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/5

Future Computing and Informatics Journal
Homepage: https://digitalcommons.aaru.edu.jo/fcij/
doi: http://Doi.org/10.54623/fue.fcij.7.1.5

Fuzzy Reasoning Procedure for Ontologies based on Rough Membership Approximation

Armand Florentin Donfack Kana [1, a], Babatunde Opeoluwa Akinkunmi [2, b]

[1] Ahmadu Bello University, Nigeria, [2] University of Ibadan, Nigeria

[a] donfackkana@gmail.com, [b] bo.akinkunmi@ui.edu.ng

ABSTRACT- One of the major challenges in modeling a real-world domain is how to effectively represent uncertain and incomplete knowledge of that domain. Several techniques for representing uncertainty in ontologies have been proposed with some of the techniques lacking provision for vague inference. The classical tableaux-based algorithm does not provide the flexibility for reasoning over such vague ontologies. However, several extensions of the tableaux-based algorithm have been proposed to cope with fuzzy reasoning. Similarly, several alternative reasoning methods for incomplete, inconsistent, and uncertain ontologies have been proposed. One of the major limitations of most of those techniques is that they require reengineering existing ontologies to cope with uncertainty. This paper proposes a satisfiability algorithm for vague ontologies that uses a rough set to approximate the concepts and roles. The proposed technique takes advantage of the existing ontology knowledge base to achieve vague reasoning without the need of reengineering the ontology. The results show that the proposed technique conforms to the tableaux-based algorithm while providing a way of reasoning over the uncertain aspects of ontologies.

Keywords: Uncertainty, Ontology, Satisfiability, Rough set, Description Logics

1. Introduction

One of the key features of ontologies especially the Description Logics(DLs) ontologies is their inference capability. The inference is important to derive implicit knowledge from explicitly represented knowledge. This is essential to determine the consistency of the ontology as well as determining the satisfiability of defined concepts and roles. Classical reasoning algorithms assume that ontologies are built for a crisp domain using a crisp logic. As a result, the resulting ontology gives an a priori model of the domain, which must accept as true by all users thereby neglecting or wrongly capturing the vague aspects of the domain. There are several situations where it is preferable to store a piece of information in an uncertain form as it appears rather than approximating it. Several techniques for modeling the uncertain aspects of the world have been proposed. They include the Dempster-Shafer theory[1], Human-Inspired Model[2], fuzzy logic[3], probabilistic[4][5], rough set[6, 7]. Most of these proposed methods led to the extension of the language to support the representation of

Published by Arab Journals Platform, 2022

uncertainty. Fuzzy ontology is widely seen as a solution to the problem of uncertainty in the ontology. In fuzzy ontologies, the uncertain aspects of an application domain are represented by using fuzzy concepts, fuzzy relationships, fuzzy datatypes, and axioms that only hold to some degree of truth[3]. A survey of Fuzzy logic extension of DLs was presented in [8]. The state of the art of fuzzy extensions to allow fuzziness in ontologies, web languages, and tools as well as several very current examples of fuzzy ontologies in real-world applications is also presented in[9]. A review of type-2 fuzzy Ontology was presented in[10]. Some of these techniques are domain or application-specific[11,12,13].

Despite the merit of these approaches, most of them do not provide a clear way on how inference on vague knowledge should be performed or do require reengineering existing ontology. This paper presents a rough satisfiability algorithm for $\mathcal{ALC}$ based on rough membership approximation using a rough set. The main advantage over existing techniques is that vagueness can be handled without the need of reengineering existing ontology. The proposed approach takes advantage of existing knowledge of ontology to achieve vague reasoning. The paper is organized as follows: Related works are reviewed in section 2, Section 3 reviews the description logic and rough set. Section 4 presents the technique for reasoning Over vague ontologies using rough set and section 6 experiments it through examples. Finally, section 6 concludes the paper.

2. Related work

Several techniques aiming at dealing with uncertainty in ontologies have been proposed in recent years. The major differences between them are on the selected ontological language, the fuzzy knowledge supported, and the fuzzy reasoning approach. A survey of automata-based techniques for uncertain reasoning in fuzzy DLs that emphasizes on the main constructors used was performed in [14]. A logical entailment between the domain-specific ontology and entities using fuzzy rule was proposed in[13] to give a better retrieval rate in fuzzy ontology for images. A minimalistic reasoning algorithm to solve imprecise instance retrieval in fuzzy ontologies with application to querying Building Information Models was proposed in[15]. The author proposed a novel lossless reduction of fuzzy to crisp reasoning tasks, which can be processed by any Description Logics reasoner. A fuzzy logic reasoner fuzzyDL was proposed in[16] to support fuzzy reasoning in an expressive is a DL reasoner using a combination of a tableaux algorithm and Mixed-integer linear programming (MILP). A tableau algorithm for computing the inconsistency degree of a knowledge-base in possibilistic DL was proposed in [17]. the proposed procedure was designed for ALC extended with inverse roles and transitive roles. A non-monotonic probabilistic reasoner named Pronto[5] was developed to reason about uncertainty in OWL ontologies. Pronto is built on top of OWL reasoner to provide routines for higher-level probabilistic reasoning procedures while maintaining existing OWL reasoning services. A probabilistic inference named BUNDLE was proposed in [4]. Bundle is able to exploit pellet as well as other non probabilistic OWL reasoners perform probabilistic reasoning.

3. Preliminaries

3.1 Description logics

Description logics[18] are knowledge representation formalisms used to model an application domain in a structured and formally well-understood way.

Elementary descriptions are *atomic concepts* and *atomic roles* from which, *c*omplex descriptions can be built from them inductively with *concept constructors* and *role constructors.* most recent DLs Concept descriptions in $\mathcal{ALC}$ are formed according to the following syntax rule:

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/5
DOI: https://doi.org/10.54623/fue.fcij.7.1.5

C,D →A | (atomic concept)
⊤ | (universal concept)
⊥ | (bottom concept)
¬ A |(atomic negation)
C ⊓ D |(intersection)
$\forall R.$ ⊤ |(value restriction)
$\exists R.$⊤ |(limited existential quantification).

ALC knowledge base consists of a set of terminological axioms (TBox) and a set of assertional axioms (ABox). Expressive DLs also allow role axioms (RBox).

- TBox axioms capture relationships between concepts. Figure1 shows a snippet of a Tbox of a family domain .

Woman ≡ Person ⊓ Female
Man ≡ Person ⊓ ¬ Woman
Father ≡ Man ⊓ ∃hasChild.Person
Mother ≡ Woman ⊓ ∃hasChild.Person
Parent ≡ Father ⊔ Mother
MotherWithoutDaughter ≡ Mother ⊓ ∀hasChild. ¬ Woman
HappyFather ≡ Man ⊓∃hasChild.Person ⊓∀careFor.Healthy
HealthyPerson ≡ Person ⊓ Healthy
Healthy ≡ MentallyStable ⊓ EmotionallyStable ⊓ MedicallySound
Husband≡ Man⊓ isMalePartnerIn. (Marriage⊓ hasFemalePartner. Woman)

Figure 1 TBox

- ABox axioms capture knowledge about named individuals.

For example, the concept assertion Father(edmund_bright_1813) asserts that edmund_bright_1813 is an instance Father. The role assertion isFatherOf (edmund_bright_1813, john_ bright _1842)asserts that edmund_bright_1813 is the father of john_ bright _1842. Figure 2 shows a sample Abox.

- RBox axioms capture interdependencies between the roles. For example, the role inclusion *isFatherOf ⊑ isParentOf*

One of the key operation of ontology is instantiation. That is the act of deciding if an arbitrarily chosen individual *x* is an instance of a concept *C* denoted by *C(x)* or if a given pair of individuals *x* and *y* are instances of a binary relation *R* denoted by *R(x, y)*. Instantiation is used in the process of populating the ontology's knowledge base and queries answering.

Through their inference capability, DLs can infer additional knowledge from the knowledge explicitly stated in an ontology.

An ontology is said to be satisfiable if an interpretation exists that satisfies all its axioms. otherwise, it is said to be unsatisfiable. An interpretation $I = (\Delta^I , .^I)$ consists of a set Δ^I called the domain of *I*, and an interpretation function $.^I$ that maps each atomic concept A to a set $A^I \subseteq \Delta^I$, every role R to a binary relation R^I, subset of $\Delta^I \times \Delta^I$ and each individual name *a* to an element $a^I \in \Delta^I$.

Published by Arab Journals Platform, 2022

isFatherOf (edmund_bright_1813, john_ bright _1842) isFatherOf (edmund_bright_1813, mary_ bright _1845)
hasSister (edmund_bright_1813, caroline_bright_1822) hasSister (edmund_bright_1813, eliza_bright_1825) hasBrother (edmund_bright_1813, james_bright_1809) hasBrother (edmund_bright_1813, william_bright_1827) isMotherOf (sarah_webb, william_bright_1827) isMotherOf (sarah_webb, james_bright_1809)
CareFor(edmund_bright_1813, john_bright_1842) CareFor(edmund_bright_1813, mary_bright_1845)
CareFor(edmund_bright_1813,caroline_bright_1822) CareFor(edmund_bright_1813,james_bright_1809)
Healthy(*john_ bright _1842*) EmotionallyStable (edmund_bright_1813) MentallyStable(edmund_bright_1813)
Healthy(mary_bright_1845) EmotionallyStable (caroline_bright_1822) MentallyStable(eliza_bright_1825)
Healthy(caroline_bright_1822) EmotionallyStable (william_bright_1827) MentallyStable(*john_ bright _1842*)
Healthy(william_bright_1827) EmotionallyStable (james_bright_1809) MentallyStable(william_bright_1827)
MedicallySound (james_bright_1809) EmotionallyStable (eliza_bright_1825) MedicallySound (sarah_webb)
MentallyStable(mary_bright_1845) EmotionallyStable(sarah_webb) MedicallySound (caroline_bright_1822)
MedicallySound (mary_bright_1845) MedicallySound (*john_ bright _1842*)

Figure 2 ABox about family relationships

3.2 Rough set

Rough set[19] is based on the assumption that the attributes of an object can be used to describe the information associated with the object. Objects that cannot be distinguished based on the selected set of attributes are called indiscernible. The indiscernibility relation expresses the inability to distinguish some objects based on available knowledge about them. Therefore, by dealing with objects as clusters, meaningful knowledge about them can be obtained than dealing with a single object.

A rough set is defined by two sets, the lower approximation and the upper approximation. Given a set X and an indiscernibility relation R which is assumed to be an equivalence relation, the *lower approximation* of X with respect to R is the set of all objects that can be for sure classified as *X*. The upper approximation of X with respect to R is the set of all objects which can be *possibly* classified as *X*. Formally, rough set is defined as follow:

Definition 1: Suppose we are given a set of objects *U* called the universe and an indiscernibility relation $R \subseteq U \times U$. Let *X* be a subset of *U*.

- *R-lower approximation* of *X* is defined by $R_*(x) = \bigcup_{x \in U} \{R(x) : R(x) \subseteq X\}$ (1)
- *R-upper approximation* of *X* is defined by $R^*(x) = \bigcup_{x \in U} \{R(x) : R(x) \cap X \neq \emptyset\}$ (2)
- *R-boundary region* of *X* is defined by $RN_R(X) = R^*(X) - R_*(X)$ (3)

A Set is crip or precise if it has an empty boundary region otherwise, the set is rough.

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/5
DOI: https://doi.org/10.54623/fue.fcij.7.1.5

4. Rough reasoning Over vague ontologies

An ontology is made up of the set of concepts $C = \{ c_{1,} c_2, c_{3,} \dots, c_m\}$ where c_i is an atomic or complex concept defined in the Tbox, the set of roles $R = \{ r_{1,} r_{2,}, r_{3,}, \dots, r_n\}$, and a set of individuals $I = \{ i_{1,} i_{2,}, i_{3,}, \dots, i_p\}$ as defined in the Abox. [6] showed that membership of vague concepts and roles over the set of individuals I can be approximated. The authors viewed a concept as attributed valued and constructed a decision table based on concepts' attributes to approximate their rough membership using the indiscernibility relation.

Definition 2: Let $c \in C$ be a concept of an ontology *O* and let *B* be the sets of attributes and concepts appearing on the expanded definition of *c*. Two individuals *x*, $y \in I$ are said to be indiscernible by the concept *c* if and only if *a(x)=a(y)* for every *a*∈*B.*

In other words, two individuals *x*, $y \in I$ cannot be distinguished with respect to a concept *c* if their instantiation over the sets of attributes and concepts the expanded definition of *c* are the same.

Every concept c induces a unique indiscernibility relation. The partition of *I* induced by c is denoted by *I/c* and the equivalence class in the partition containing $i \in I$, denoted by $[i]_c$. the c-upper, c-lower and the c-boundary region approximations of *X* ⊆*I* can be defined respectively as follows:

$$c_*(X) = \{i \in I \text{ such that } [i]_c \subseteq X\} \quad (4)$$

$$c^*(X) = \{i \in I \text{ such that } [i]_c \cap X \neq \emptyset\} \quad (5)$$

The *c-boundary region* $= c^*(X) - c_*(X)$ (6)

In the rest of this paper, we shall use $f^*(c)$ and $f_*(c)$ instead of $c^*(X)$ and $c_*(X)$ respectively to resents the upper and lower approximation of *c* over the set of instances I.

Definition 3: A concept c is said to be vague if its *c-boundary region* is not empty

Definition 4: An individual i, is said to be an absolute instance of $c \in C$, if and only if $i \in f_*(c)$.

Definition 5: An individual i is said to be a rough instance of $c \in C$, if and only if $i \in (f^*(c) - f_*(c))$.

Definition 6: A concept $c \in C$ is satisfiable if $f_*(c) \neq \emptyset$. if $f_*(c) = \emptyset$ and $f^*(c) \neq \emptyset$, then *c* is roughly satisfiable.

4.1. Ontologies satisfiability approximation algorithm

Reasoning capability helps in deciding the consistency of a knowledge base. This is important since the knowledge base describes the real state of the ontologies and consequently, definitions should not contradict each other. Many reasoning problems can be reduced to checking the satisfiability of the knowledge base. Accordingly

A concept $c \in C$ is called satisfiable with respect to a given knowledge base if there exists an interpretation model I that maps *c* to a non-empty set. that is $c^I \neq \emptyset$

The widely used algorithm to decide the satisfiability of ontologies is the tableaux-based algorithm. It aims at constructing a model that satisfies all axioms of the given knowledge base. Tableaux reasoning approach for several expressive description logics have been described extensively in [20, 21] Several OWL reasoners use tableaux procedures and have proved to be efficient[22]. They include Sequoia reasoner [23], FaCT++[24], or RacerPro[25]. Alternative reasoning techniques include the resolution methods[26], consequence-based approaches [27], hypertableaux [28] which is a refinement of the tableaux technique and is used as the core reasoner of OWL2 DL.

We now present a reasoning algorithm to decide the satisfiability of concepts based on membership approximation methods. This algorithm constructs a model that defines a set of possible domains of approximation dom (R), a set of all approximation relations (R) with associated range $ran(R)$, and establishes if an approximation of the defined relations holds between the domains and the ranges. Mathematically, the domain and the range of a

Published by Arab Journals Platform, 2022

relation R are respectively defined by dom $(R) = \{(a \in A \mid (\exists\ (b \in B)\ ((a,b) \in R)\}$ and
$ran\,(R) = \{b \in B \mid (\exists\ a \in A)\ ((a,b) \in R)\}$.
In the rest of this paper, we assume all concepts definition are in Negation Normal Form (a formula is in Negation Normal Form(NNF) if the negation operator is only applied to variables and the only other allowed Boolean operators are conjunction and disjunction).

Definition 7: Let *O* be an ontology and let $c \in C$ be a concept in NNF. $\delta(X, Y)$ is the satisfiability approximation model for c with respect to ontology *O* if and only if $X \subseteq \wp(I)$ is the set of domains of approximation and *Y* is the set of pair (*r*,*E*) such that E is the range of approximation and $r: d \longrightarrow E$ is a relation associating individuals from the domain $d \in X$ to elements of E where X, Y are generated from the expansion rules applied to *c*.

The set of possible domain of approximation $X = \{d_1, d_2, \ldots, d_n\}$ is defined by $d_i = f(c_1) \cap f(c_2) \cap \ldots \cap f(c_m)$ for some $c_i \in C$ where c_i's are introduced by the expansion rules defined in figure 2 and $f(c_i)$ is the approximation of c_i.
The set of pair (*r*, *E*) of possible approximation range $E = \{e_1, e_2, \ldots, e_n\}$ defined by $e_i = f(c_1) \cap f(c_2) \cap \ldots \cap f(c_m)$ for some $c_i \in C$ also introduced by the expansion rules and $r \in R$ is a relation such that (*r*,*E*) is a pair of *Y* iff $r \in R$ and is defined from X to E.
for simplicity purposes, in the remaining of this paper, we shall use $c_i \in d_i$ and $c_i \in e_i$ rather than $f(c_i) \in d_i$ and $f(c_i) \in e_i$ to denote the fact that the elements of d_i and e_i are generated from the approximation of c_i. Furthermore, the following properties must be established.

(i) if $C \in d_i$, then $\neg C \notin d_i$
(ii) if $C \in e_i$, then $\neg C \notin e_i$
(iii) if $C \in e_i$, is introduced by $\forall$ S.C, then the cardinality of the relation S:di→ ei must be ≥0 and the relation S:di→ ¬ ei must be is empty, for some $d_i \in X$
(iv) if $C \in e_i$, is introduced by $\exists$ S.C then the cardinality of the relation S:di→ ei must be > 0, for some $d_i \in X$
(v) if $C \in e_i$, is introduced by ≤nS.C, then the cardinality of the relation S:di→ ei must be ≤ n, for some $d_i \in X$
(vi) if $C \in e_i$, is introduced by ≥nS.C , then the cardinality of the relation S:di→ ei must be ≥ n, for some $d_i \in X$

lema1: a concept $c \in C$ is satisfiable with respect to ontology O iff *c* has a clash-free approximation model such that the following conditions hold:
- $\exists d \in X\ such\ that\ f(d) \neq \emptyset\ and$
- $\forall (r, E) \in Y,\ r\colon d \rightarrow e_i$ is not empty and satisfy the properties on cardinality stated above for some $e_i \in$ E

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/5
DOI: https://doi.org/10.54623/fue.fcij.7.1.5

$\sqcap$**- Rule:** if $C_1 \sqcap C_2 \in d_i$, and there is no $d \in X$ such that $d=d_i \cup \{C_1, C_2\}$ then $d_i \leftarrow d_i \cup \{C_1, C_2\}$

$\sqcup$ **- Rule:** if $C_1 \sqcup C_2 \in d_i$, and there is no d' and d'' such that $d' = d_i \cup \{C_1\}$ and $d'' = d_i \cup \{C_2\}$ then

create d' such that: $d' \leftarrow d_i \cup \{C_2\}$ and set $d_i \leftarrow d_i \cup \{C_1\}$

$\exists$**- Rule:** if $\exists S.C \in d_i$

If there is a pair $(r,E) \in Y$ such that r=S then

If there is no $d \in E$ such that $c \in d$ then,

create a new set $d \in E$ such that d={c} and set α= -

If there is no pair $(r,E) \in R$ such that r=s then

create a new pair (S,E) such that $Y=Y \cup (S,E)$ where E={{c}

$\forall$ **- Rule:** if $\forall S.C \in d_i$

If there is a pair $(r,E) \in Y$ such that r=S then

If there is no $d \in E$ such that $c \in d$ then,

create a new set $d \in E$ such that d={c} and set α= +

If there is $d \in E$ such that $c \in d$ then, set α= +

If there is no pair $(r,E) \in Y$ such that r=s then

create a new pair (S,E) such that $Y=Y \cup (S,E)$ where E ={{c}} and set α= +

$\geq$ **– rule:** if $(\geq n\ S.C) \in d_i$,

If there is a pair $(r,E) \in R$ such that r=S then

If there is no $d \in E$ such that $c \in d$ then,

create a new set $d \in E$ such that d={c} and set α=- ,β=n , γ=∞

If there is $d \in E$ such that $c \in d$ then, set β=n, γ=∞

If there is no pair $(r,E) \in Y$ such that r=s then

create a new pair (S,E) such that $Y=Y \cup (S,E)$ where E is E={{c}}and set α=-,β=n, γ=∞

$\leq$ **– rule:** if $(\leq n\ S.C) \in d_i$,

If there is a pair $(r,E) \in Y$ such that r=S then

If there is no $d \in E$ such that $c \in d$ then,

create a new set $d \in E$ such that d={c} and set α=- ,β=0 , γ=n

If there is $d \in E$ such that $c \in d$ then, set β=0, γ=n

If there is no pair $(r,E) \in R$ such that r=s then

create a new pair (S,E) such that $Y=Y \cup (S,E)$ where E={{c}} and set α=- ,β=0 , γ=n

Figure 3 Expansion Rule

The cardinality property ensures that the universal and existential qualifications as well as the cardinality restriction of the relations are satisfied. Let $\Delta\ (\alpha,\beta,\gamma)$ defines the characteristics of the relation *r* with respect to *E* such that α represents the qualification, β the minimum cardinality and γ the maximum cardinality of the relation *r* with respect to E. α takes the value of "-" if the relation is introduced by the existential qualification or in the absence of qualification and takes the value of "+" if the relation is introduced by the universal qualification.

The cardinality properties constraints is defined as follow:

- If $\alpha = -$, then cardinality of r : $d_i \in X \rightarrow e_i \in E$ >0
- If $\alpha = +$, then the cardinality of r: $d_i \in X \rightarrow e_i \in E$ ≥ 0 and r: $d_i \in X \rightarrow \neg e_i \in E$ is empty
- $\beta \leq \gamma$

The approximation model is said to be fully expanded if no further expansion is possible.

The expansion contains a clash if (c and $\neg$ c) $\in$ d_i or e_i. If the expansion contains some inconsistencies with respect to the cardinality properties, then the expansion model has a clash. Clashes either in the domain and range of approximation or in cardinality properties invalidate only the affected sets or relations. They are said to be blocked. Blocked sets can no longer be expanded since they contain a contradiction. The approximation model is said to be fully blocked when all possible sets or relations of approximation are blocked.

The algorithm starts by initializing the D with a single set containing c that is X={{c}} and Y=∅. for each $d_i \in X$ if d_i contains a non-leave concept, substitute it with its definition and apply the expansion rule on it until no further expansion is possible or d_i is blocked. For each $e_i \in E$ if e_i contains a non-leave concept, substitute it with it definition and apply the expansion rule on it until no further expansion is possible or e_i is blocked.

4.2 Interpretation of Satisfiability of Vague Concepts.

Unlike the two-state tableaux-based algorithm in which, concepts are either satisfiable or not satisfiable, the algorithm $\delta(X, Y)$ has three states of decision for vague concepts. Concepts are absolutely satisfiable, roughly satisfiable or not satisfiable.

Definition 8: A vague concept definition is said to be absolutely satisfiable or just satisfiable for short iff

$\exists i \in f_*(d_i)\ such\ that\ \forall(r, e_i), \exists j \in f_*(e_i) \wedge j \in h_*(r(i))$ for some $i, j \in I$.

where $d_i \in X$ is the domain of approximation, $h_* \subseteq f_* \times f_*$ is the lower approximation of the relation *r*(*i*), $e_i \in E$ is the range of approximation and *I* is the set of individuals.

Furthermore,

(a) If *r* is introduced by $(\forall r)$ then, $\forall(j) \in h^*(r_s(i))$, $j \in f_*(e_i)$.

(b) If *r* is introduced by $(\exists r)$ then, $\exists(j) \in h_*(r_s(i))$, $such\ that\ j \in f_*(e_i)$.

(c) If *r* is introduced by $(\geq n.r)$ then, $|h_*(r_s(i))| \geq n$ where $|h|$ denotes the cardinality of *h*.

(d) If *r* is introduced by $(\leq n.r)$ then, $|h^*(r_s(i))| \leq n$.

Definition 9: A vague concept definition is said to be roughly satisfiable iff

$\exists i \in f^*(d_i)\ such\ that\ \forall(r, e_i), \exists j \in f^*(e_i)\ and\ (i, j) \in h^*(r_s(i))$ for some $i, j \in I$

where $d_i \in X$ is the domain of approximation, $h^* \subseteq f^* \times f^*$ is the upper approximation of the relation *r(i)*, $e_i \in E$ is the range of approximation and *I* is the set of individuals.

Furthermore,

(a) If *r* is introduced by $(\forall r)$ then, $\forall(j) \in h_*(r_s(i))$, $j \in f^*(e_i)$

(b) If *r* is introduced by $(\exists r)$ then, $\exists(j) \in h^*(r_s(i))$, $such\ that\ j \in f^*(e_i)$

(c) If *r* is introduced by $(\geq n.r)$ then, $|h^*(r_s(i))| \geq n$

(d) If *r* is introduced by $(\leq n.r)$ then, $|h_*(r_s(i))| \leq n$

Definition 10: A vague concept is said to be not satisfiable if it is neither absolutely satisfiable nor roughly satisfiable.

It is necessary to note that, for crisp concepts, their satisfiability will always be evaluated to either satisfiable or not satisfiable since the upper and the lower approximations of the domains, the ranges and the relations are the same. With this, the classical tableaux-based algorithm decision is therefore recovered for crisp concepts.

5. Results and Discussion

In this section, we will demonstrate how the proposed reasoning procedure can be applied on vague ontology and compare the results with the tableaux-based algorithm. Tableaux procedure aims at constructing a model that satisfies all axioms of the given knowledge base. It is implemented as a finite tree which is expanded as concepts are expanded using the expansions rule [29]. Nodes of the tree are labeled with concept name and edges are labeled with role occurring between concepts. Like the algorithm presented in this work, the tableaux-based algorithm assumes the concepts' definition to be in NNF. For the sake of simplicity, we restrict the approximation throughout this section to the data available in Tbox and Abox defined in section 2.1. which were extracted and adapted from a large OWL ontology defined in [30]

Example1: approximation of the vague concept using the proposed algorithm.

$HappyFather \sqsubseteq Man \sqcap \exists hasChild.Person \sqcap \forall careFor.Healthy$

By expanding this definition of *HappyFather*, we obtain

$$HappyFather \sqsubseteq (\text{Person} \sqcap \neg \text{Woman}) \sqcap \exists hasChild.Person \sqcap \forall careFor.Healthy$$

$$\sqsubseteq (\text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female}) \sqcap \exists hasChild.Person \sqcap \forall careFor.Healthy$$

$$\sqsubseteq (\text{Person} \sqcap (\neg \text{Person} \sqcup \neg \text{Female}) \sqcap \exists hasChild.Person \sqcap \forall careFor.Healthy$$

$$HappyFather \sqsubseteq \begin{cases} \text{Person} \sqcap \neg \text{Person} \sqcap \exists hasChild.Person \sqcap \forall careFor.Healthy \ (i) \\ \text{Person} \sqcap \neg \text{Female} \sqcap \exists hasChild.Person \sqcap \forall careFor.Healthy \ (ii) \end{cases}$$

Note that(i) is blocked due to the contradiction Person $\sqcap \neg$ Person . Table 1 is obtained by applying the expansion rules defined in figure 2. This is also shown pictorially in figure 3. The existence of the following relations needs to be established and shows that each of them satisfies its cardinality properties as specified in Figure 3 by using the Abox knowledge.

$$\{ f(Person) \cap f(\neg female) \} \xrightarrow{haschild} \{ f(Person) \}$$

$$\{ f(Person) \cap f(\neg female) \} \xrightarrow{CareFor} \{ f(Healthy) \}$$

In the expansion of Table 1, *person* is the most general concept and can be treated similarly as crisp attribute, *hasChild* is a crisp relation. However *careFor* is a vague relation and healthy is a vague attribute.

Table 1. Expansion of *HappyFather*

	Domain	Relation	Range	Description
1	{{ HappyFather }}	-	-	-
2	{{Man, haschild.Person, ∀careFor.Healthy}}	-	-	-
3	{{Person, ¬ Female, haschild.Person, careFor.Healthy }}	-	-	-
4	{ {Person, ¬Female , ∀careFor.Healthy }}	haschild	{{Person}}	α =+, β=1,γ=∞
5	{ {Person, ¬Female }}	careFor	{{Healthy}}	α =+, β=1,γ=∞

Published by Arab Journals Platform, 2022

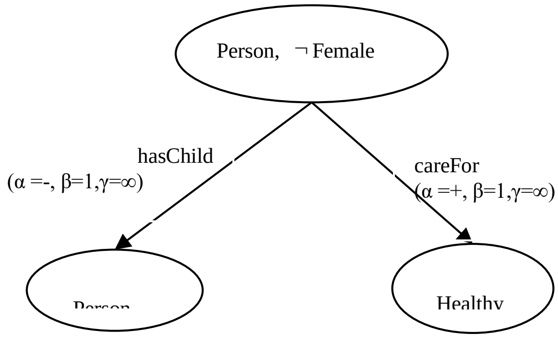

Figure 3. Representation of the Expanded Model of HappyFather

Based on Abox of figure 2, the following knowledge can be derived.
f(Person)={ sarah_webb*, edmund_bright_1813, john_ bright _1842, mary_ bright _1845,* caroline_bright_1822*, eliza_bright_1825, james_bright_1809, william_bright_1827}*
f(female)={ sarah_webb*, mary_ bright _1845, eliza_bright_1825,* caroline_bright_1822 *}*

The satisfiability of *{ f(Person)∩ f(¬ female) }* $\xrightarrow{haschild}$ *{ f(Person) }* can now be evaluated.

f(¬ Female)= f(Person)- f(Female)={ edmund_bright_1813, john_ bright _1842, james_bright_1809, william_bright_1827}

f(Person)∩f(¬ Female)={ edmund_bright_1813, john_ bright _1842, james_bright_1809, william_bright_1827}

The partitions created by the relation $hasChild$ from the set *f(Person)∩f(¬ Female)* to the set *f(person)* based on the knowledge in the Abox can now be defined. Consider the individual *edmund_bright_1813 then,*
$hasChild$ *(edmund_bright_1813)={ john_ bright _1842, mary_bright_1845)}*
Since the cardinality of $hasChild_s$*(edmund_bright_1813)>0,* the relation
hasChild { f(Person)∩ f(¬ female) }→ { f(Person) } is satisfied by *edmund_bright_1813.*

The satisfiability of *{ f(Person)∩ f(¬ female)}* $\xrightarrow{CareFor}$ *{ f(Healthy)}* can now be evaluated. Based on the knowledge in the Abox, the decision table of Table 2 can be constructed.

Table 2. Decision Table of *healthy*

	Attributes			
f(person)	MentallyStable	EmotionallyStable	MedicallySound	**Healthy**
edmund_bright_1813	1	1	0	0
john_ bright _1842	1	0	1	1
james_bright_1809	0	1	1	0
caroline_bright_1822	0	1	1	1
mary_bright_1845	1	0	1	1
eliza_bright_1825	1	1	0	0
sarah_webb	0	1	1	0
william_bright_1827	1	1	0	1

From the table, the set of partitions of *f(person)* with respect to *Healthy* is defined as follows:

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/5
DOI: https://doi.org/10.54623/fue.fcij.7.1.5

[MentallyStable:*1*; EmotionallyStable:*0*; MedicallySound:*1*]= {*mary_ bright _1845, john_ bright _1842*}
[MentallyStable:0; EmotionallyStable:*1*; MedicallySound:*1*] = {*james_bright_1809*, sarah_webb, *caroline_bright_1822*}
[MentallyStable:*1*; EmotionallyStable:*1*; MedicallySound:0] = {*william_bright_1827, eliza_bright_1825, edmund_bright_1813*}
The rough approximation of Healthy are therefore as follow:
$f_*(Healthy)$={{ *mary_ bright _1845, john_ bright _1842* }
$f^*(Healthy)$ ={{ *mary_ bright _1845, john_ bright _1842*}, {*james_bright_1809, caroline_bright_1822*, sarah_webb }, {*william_bright_1827, eliza_bright_1825, edmund_bright_1813*}}
Since $f_*(Healthy) \neq f^*(Healthy) \neq \emptyset$, *Healthy* is vague and satisfied.
Therefore, the approximation of the relation *CareFor* over the Cartesian product $f(Person) \times f(Healthy)$ can be defined in such a way that,
$(x, y) \in CareFor \Rightarrow x \in f(Person) \wedge y \in f(Healthy)$
Assuming an individual *edmund_bright_1813* then, the partition defined by $CareFor$ is as follow:
$CareFor$ *(edmund_bright_1813)={ john_bright_1842, mary_bright_1845,caroline_bright_1822, james_bright_1809}*
The granule of knowledge about *edmund_bright_1813* are the following.
$hasChild$ *(edmund_bright_1813)={ john_bright_1842, mary_bright_1845 }*
$hasSister$ *(edmund_bright_1813)={ caroline_bright_1822,eliza_bright_1825 }*
$hasBrother$ *(edmund_bright_1813)={james_bright_1809,william_bright_1827) }*
These granules of knowledge can be regarded as the partitions created based on the knowledge on *edmund_bright_1813.*
By comparing the $CareFor\ (edmund\_bright\_1813)$ with others granules of knowledge stated above, it appears that, at the upper approximation, edmund_bright_1813 cares for anybody related to him by blood. But the lower approximation shows that, he cares for all his children. Thus,
$CareFor^*(edmund\_bright\_1813)$={$hasChild$ *(edmund_bright_1813), *$hasSister$ *(edmund_bright_1813),* $hasBrother$ *(edmund_bright_1813)*}
$CareFor_*(edmund\_bright\_1813)$={$hasChild$ *(edmund_bright_1813)}*
The cardinality of $CareFor_*(edmund\_bright\_1813)$ with respect to the range of approximation *f(Healthy)* is greater than 0 and none of the $CareFor_*$*(edmund_bright_181)* belongs to *f(¬Healthy).* Consequently,
$\{ f(Person) \cap f(\neg female) \} \xrightarrow{CareFor} \{ f(Healthy) \}$ is roughly satisfied. Thus, *edmund_bright_1813* is a rough instance of *HappyFather.* Therefore, *HappyFather* is roughly satisfied.
Example 2**:** Satisfiability model of the vague concept *HappyFather* using tableaux-based algorithm.
HappyFather ⊑ *Man* ⊓∃*hasChild.Person* ⊓ ∀*careFor.Healthy*
Step1: Expansion of *HappyFather* based on the concept definition in the Tbox.
HappyFather ⊑ *Man* ⊓∃*hasChild.Person* ⊓ ∀*careFor.Healthy*
⊑ *(Person* ⊓ ¬*Female)* ⊓∃*hasChild.Person* ⊓ ∀*careFor.Healthy*
Step 2: Construction of a tree model for concept *HappyFather*
Figure 4 represents the tree model of *HappyFather*.
From the knowledge in the Abox, the following can be established
Healthy ^{I} *={ john_ bright _1842, mary_ bright _1845, caroline_bright_1822, william_bright_1827}*
$CareFor$ *(edmund_bright_1813)={ john_bright_1842, mary_bright_1845, caroline_bright_1822, james_bright_1809 }*

Published by Arab Journals Platform, 2022

hasChild (edmund_bright_1813)={ john_bright_1842, mary_bright_1845}

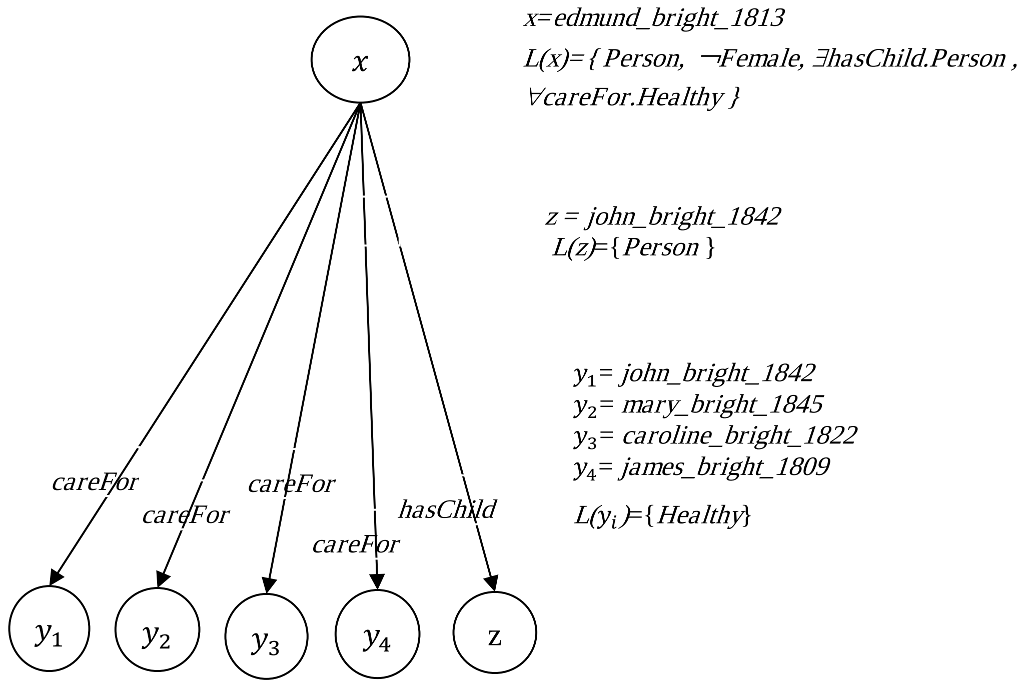

Figure 4. Tree of *HappyFather*

Starting with *edmund_bright_1813*, from the tree model, the relation *hasChild (edmund_bright_1813)* is satisfied. However, *∀careFor.Healthy* is not satisfied since *james_bright_1809* is an instance of *CareFor (edmund_bright_1813*) but not an instance of $Healthy^I$. This violates the requirement of universal quantification. Therefore, *HappyFather* is not satisfiable with respect to the interpretation.

The satisfiability of *HappyFather* based on tableaux-based algorithm and the proposed algorithm $\delta(X, Y)$ in Example 1 and 2 respectively all terminate. However, *HappyFather* was roughly satisfiable with $\delta(X, Y)$, and is not satisfiable while investigating with the tableaux-based interpretation as expected. This is where $\delta(X, Y)$ differs from the tableaux-based algorithm. Since tableaux-based algorithm cannot interpret uncertain relations or concepts in an approximate manner.

Example 3: Satisfiability model of *MotherWithoutDaughter ≡Mother⊓ ∀hasChild.¬Woman* using the tableaux-based algorithm

Step1: Expansion of *MotherWithoutDaughter*

MotherWithoutDaughter ≡ Mother⊓ ∀hasChild.¬Woman.

≡(Woman ⊓ hasChild.Person) ⊓ ∀hasChild.¬Woman

≡((Person ⊓Female) ⊓ hasChild.Person) ⊓ ∀hasChild.¬(Person ⊓Female)

≡((Person ⊓Female) ⊓ hasChild.Person) ⊓ ∀hasChild. (¬Person ⊔¬Female)

≡((Person⊓Female)⊓≥ 1hasChild.Person)⊓

(∀hasChild.¬Person⊔ ∀hasChild.¬Female)

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/5
DOI: https://doi.org/10.54623/fue.fcij.7.1.5

$$\equiv \begin{cases} \text{Person} \sqcap \text{Female} \sqcap \exists hasChild.Person \ \sqcap \ \forall hasChild.\neg Female\ (i) \\ \text{Person} \sqcap \text{Female} \sqcap \exists hasChild.Person \sqcap \ \forall hasChild.\neg Person (ii) \end{cases}$$

Step 2: Construction of a tree model for concept *MotherWithoutDaughter*

Figure 5 represents the tree model of *MotherWithoutDaughter*. Node x_2 is blocked for further expansion since $L(x_2)$ contains a clash.

From the knowledge in the Abox, the following interpretations can be established.

*hasChild(*sarah_webb*)={james_bright_1809, william_bright_1827}*

¬Female ^{I}= f(Person) - f(Female)={edmund_bright_1813, james_bright_1809, william_bright_1827}

Since *hasChild(*sarah_webb*)* ⊆ *¬Female^I*, y_1 , y_2 all satisfy $L(y_i)$={ Person, ¬Female}. Thus *MotherWithoutDaughter* is satisfiable.

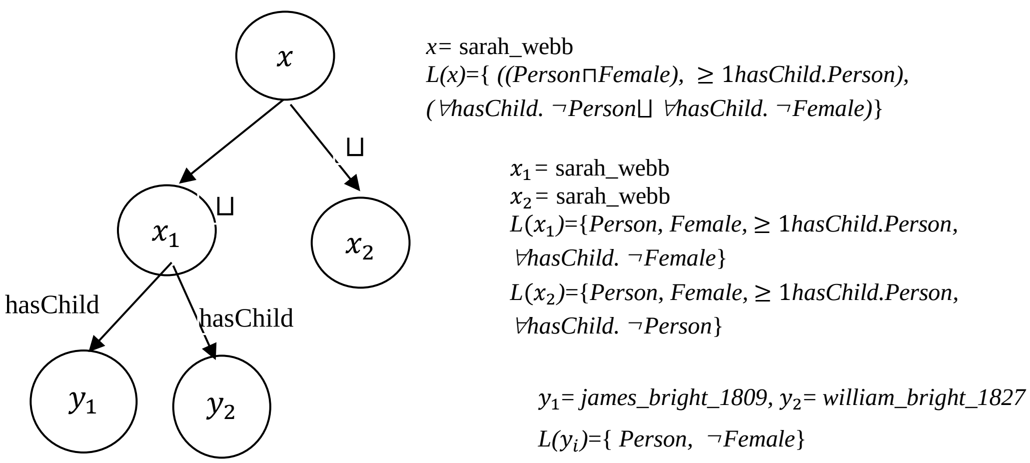

Figure 5. Tree Model of *MotherWithoutDaughter*

Example 4: Assuming the approximation of the concept *motherwithoutdaughter* using the proposed algorithm.

MotherWithoutDaughter≡Mother⊓ ∀hasChild. ¬Woman

Similarly to example 1, Table 2 is obtained by applying the expansion rules on *MotherWithoutDaughter*.

Table 3. Expansion model of MotherWhithoutDaughter

Steps	Domain	Relation	Range	Description
1 2 3 4 5 6 7	{{ MotherWithoutDaughter }} {{ Mother , ∀ haschild.Man}} {{ Woman , ≥1haschild.Person , ∀haschild.Man}} {{Person, Female , ∃haschild. Person, ∀ haschild.Man}} { {Person, Female ∀ haschild.man}} { {Person, Female }} { {Person, Female }}	haschild haschild haschild	{{ Person }} {{Person, Woman}} {Person, ¬ Female}}	α=-,β=1,γ=∞ α =+, β=1,γ=∞ α =+, β=1,γ=∞

Finally, one needs to establish that the relation: *haschild: { f(Person)∩ f(Female) }→{ f(Person)∩ f(*

¬ *Female) }* is defined and satisfies the cardinality properties by using the Abox knowledge.

f(Person)∩f(¬ Female)={ edmund_bright_1813, john_ bright _1842, james_bright_1809, william_bright_1827}

f(Person)∩f(female)={ sarah_webb, *mary_ bright _1845, eliza_bright_1825,* caroline_bright_1822 *}*

Because *person, female* are all crisp and the role *hasChild* is also a crisp role, the approximation of *MotherWithoutDaughter* is crisp.

Consider the individual sarah_webb then, the set of *hasChild(*sarah_webb)to the range (f(Person)∩f(¬ Female)) is defined as follow

hasChild (sarah_webb)={james_bright_1809, william_bright_1827}

The cardinality constraint of the relation is $\Delta=(\alpha =+, \beta=1,\gamma=\infty)$ as shown in table 3

The cardinality of *hasChild*(sarah_webb) with respect to the range of approximation (f(Person)∩f(¬ Female)) is greater than 0 and, the cardinality of hasChild(sarah_webb) to ¬ (f(Person)∩f(¬ Female)) is also is 0. Therefore the cardinality constrain is satisfied. Thus, Sarah_web is an instance of *MotherWithoutDaughter*. Consequently, *MotherWithoutDaughter* is satisfiable.

The satisfiability of *MotherWithoutDaughter* based on tableaux-based algorithm and that of $\delta(X,Y)$ all terminate and show that *MotherWithoutDaughter* is satisfiable with respect to the available knowledge. Since *MotherWithoutDaughter* is a crisp concept, it was expected that the two algorithms should terminate with the same decision.

6.Conclusion

This paper presents a reasoner for vague DL ontologies to approximate membership thereby providing room for soft reasoning over $\mathcal{ALC}$ ontologies. We have shown through examples that, the satisfiability of ontologies with uncertainty can be approximated by constructing a tableaux-based like model with roles and concepts interpretation approximated through rough set. The separation of the representation of uncertainty from the reasoning mechanism presented here helps in achieving this without necessarily remodeling existing ontologies. Although $\mathcal{ALC}$ is limited in terms of expressivity, the same principle can be extended to expressive description logic since rough set supports all constructs used for defining complex concepts and roles of expressive description logics. Future works will look into achieving a satisfiability reasoning of expressive DL such as SROIQ which is the core logic of OWL.

References

[1] Karanikola, L., & Karali, I. (2018). Towards a Dempster–Shafer Fuzzy Description Logic—Handling Imprecision in the Semantic Web. IEEE Transactions on Fuzzy Systems, 26(5), 3016-3026.

[2] Pileggi, S. F. (2018). A human-inspired model to represent uncertain knowledge in the Semantic Web. In *International Conference on Computational Science* 254-268. Springer, Cham.

[3] Huitzil, I., Bobillo, F., Gómez-Romero, J., et al. (2020) Fudge: fuzzy ontology building with consensuated fuzzy datatypes. Fuzzy Sets and Systems 401, 91-112

[4] Cota, G., Riguzzi, F., Zese, R., Bellodi, E., Lamma, E. (2018). A Modular Inference System for Probabilistic Description Logics. In: Ciucci, D., Pasi, G., Vantaggi, B. (eds) Scalable Uncertainty Management. SUM 2018. Lecture Notes in Computer Science(), vol 11142. Springer, Cham. https://doi.org/10.1007/978-3-030-00461-3_6

[5] Klinov P. and P. Bijan (2013) Pronto: A Practical Probabilistic Description

Logic Reasoner. In Uncertainty Reasoning For The Semantic Web II. Lecture Notes In Computer Science vol.7123, pp. 59-79

[6] Kana AFD, BO Akinkunmi (2016)Modeling uncertainty in ontologies using rough set International Journal of Intelligent Systems and Applications 8 (4), 49

[7] Keet C.M. (2010) Ontology engineering with rough concepts and instances. 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW'10), P. Cimiano and H.S. Pinto (Eds.). 11-15 October 2010, Lisbon, Portugal. Springer LNAI 6317, 50.

[8] Borgwardt S. & R. Peñaloza (2017)Fuzzy Description Logics – A Survey S. Moral et al. (Eds.): SUM 2017, LNAI 10564, pp. 31–45, 201

[9] Cross V.& S. Chen(2018) Fuzzy Ontologies: State of the Art Revisited G. A. Barreto and R. Coelho (Eds.): NAFIPS 2018, CCIS 831, pp. 230–242.

[10] Qasim, I., Alam, M., Khan, S. *et al.* (2020). A comprehensive review of type-2 fuzzy Ontology. *Artif Intell Rev* 53, 1187–1206

[11] Liu B.., J. Li and Y. Zhao (2011)A Query-specific Reasoning Method for Inconsistent and Uncertain Ontology, in International Multi-Conference of Engineers and Computer scientists , Hong Kong

[12] Jain S., K.R. Seeja, R. Jindal (2021)A fuzzy ontology framework in information retrieval using semantic query expansion International Journal of Information Management Data Insights, 1 (1)

[13] Rajasekaran I., M., Govindan, N., Divakarla Naga Satya, R.K. *et al.* (2021).Fuzzy rule based ontology reasoning. *J Ambient Intell Human Comput* 12, 6029–6035

[14] Borgwardt, S., & Peñaloza, R. (2016). Reasoning in fuzzy description logics using automata. *Fuzzy Sets and Systems*, *298*, 22-43.

[15] Huitzil, I., Molina-Solana, M., Gómez-Romero, J. and Bobillo, F. (2021), "Minimalistic fuzzy ontology reasoning: An application to Building Information Modeling", Applied Soft Computing, 103, 107-158

[16] Bobillo F. and Straccia U.(2016) The fuzzy ontology reasoner fuzzyDL. Knowledge-Based Systems 95(1) 12-34

[17] Zhu J., G. Qi and B. Suntisrivaraporn (2013) Tableaux Algorithms for Expressive Possibilistic DescriptionLogics, in IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies.

[18] Baader, F., Horrocks, I., Lutz, C., Sattler, U.: (2017) An Introduction to Description Logic. Cambridge University Press, Cambridge

[19] Pawlak Z.(1982) Rough sets, International Journal of Information and Computer Sciences 11, 341-356.

[20] Horrocks I, U. Sattler, and S. Tobies(1999) Practical Reasoning for Expressive Description Logics. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Proc. of the 6th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR'99)*, number 1705 in Lecture Notes in Artificial Intelligence, pages 161-180. Springer

[21] Horrocks I and Sattler U.(2001) Ontology Reasoning in the *SHOQ*(D) Description Logic. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)* 199-204. Morgan Kaufmann, Los Altos.

[22] Rudolph S. (2011)Foundations of description logics. In Axel Polleres, Claudia d'Amato, Marcelo Arenas, Siegfried Handschuh, Paula Kroner,

Sascha Ossowski, and Peter F. PatelSchneider, editors, Reasoning Web. Semantic Technologies for the Web of Data – 7th International Summer School 2011, volume 6848 of LNCS, pp 76–136. Springer

[23] Cucala DT, Cuenca Grau B, Horrocks I (2019) Sequoia: a consequence based reasoner for SROIQ. In: Proceedings of the 32nd international workshop on description logics (DL 2019), CEUR workshop proceedings, vol 2373. CEUR-WS.org. http://ceur-ws.org/Vol-2373/paper-27.pdf

[24] Tsarkov D. and Horrocks I (2006) FaCT++ description logic reasoner: System description. In Ulrich Furbach and Natarajan Shankar, editors, Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR'06), volume 4130 of LNCS, 292-297. Springer

[25] Haarslev, Volker, and Ralf Möller. "Racer: A Core Inference Engine for the Semantic Web." *EON*. Vol. 87. 2003.

[26] Habiballa H.(2007) Resolution strategies for fuzzy description logic. In Proceedings of the 5th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2007), 2, pp. 27–36.

[27] Kazakov Y.(2009). Consequence-driven reasoning for horn SHIQ ontologies. In Craig Boutilier, editor, Proceedings of the 21st International Conference on Artificial Intelligence (IJCAI'09), 2040-2045.

[28] Horrocks, Ian, Boris Motik, and Zhe Wang. (2012) The HermiT OWL Reasoner."*ORE* 858

[29] Horrocks, I. (2002). Reasoning with expressive description logics: Theory and practice. *Proceeding of the 19th International Conference on Automated Deduction (CADE 2002), number 2392 in Lecture Notes in Artificial Intelligence* 1-15

[30] www.cs.man.ac.uk/~stevensr/ontology/family.rdf.owl

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/5
DOI: https://doi.org/10.54623/fue.fcij.7.1.5