

Proposed Representation Approach Based on Description Logics Formalism

Yasser Yahiaoui

Institute of sciences & technologies department of mathematics & Informatics Engineering, Naâma University center,
Naâma, Algeria
E-mail: alyahiaoui@gmail.com

Ahmed Lehireche

EEDIS Labo, University of Sidi Bel Abbes, Algeria
E-mail: elhir@univ-sba.dz

Djelloul Bouchiha

Institute of sciences & technologies department of mathematics & Informatics Engineering, Naâma University center,
Naâma, Algeria
E-mail: bou_dje@yahoo.fr

Abstract—The most familiar concept in Artificial intelligence is the knowledges representation. It aims to find explicit symbolization covering all semantic aspects of knowledge, and to make possible the use of this representation to produce an intelligent behavior like reasoning.

The most important constraint is the usability of the representation; it's why the structures used must be well defined to facilitate manipulation for reasoning algorithms which leads to facilitate their implementation.

In this paper we propose a new approach based on the description logics formalism for the goal of simplification of description logics system implementation. This approach can reduce the complexity of reasoning Algorithm by the vectorisation of concept definition based on the subsumption hierarchy.

Index Terms—Knowledge representation, subsumption hierarchical attribute, reasoning Algorithm, description logics, vectorial reasoning.

I. INTRODUCTION

The knowledge representation is a very important aim for artificial intelligence; it can be a beginning for intelligent behavior generation. Lots of representation formalisms are created for this goal. They are constrained by being well structured, usable, simplified, and expressive like the most important constraint.

The description logics formalism is one of the most important representation formalism. This last based on two kinds of relations: the subsumption and the equivalence that define expressions using constructors like union (disjunction), intersection (conjunction), and negation (complement). This formalism provides more detail by the quantification and the restrictions possible by the existential quantifier, the universal quantifier and

the numerical restriction. These tools provide a strong and expressive formalism of knowledge representation. And allows to define The minimal language AL as “contains the atomic concepts, the universal concept, the bottom concept, atomic negation, intersection, value restriction and limited existential quantification”. [1]. It leads to seeking for best reasoning Algorithms based on these definitions.

Description logics are defined as a family of logic-based knowledge representation language that can be used to represent the terminological knowledge of an application domain in structured way [2].

To rich more expressivity of knowledge representation DLs languages have seen many augmentations, and developments.

The implementation of DLs systems faces to problematic the effectiveness of reasoning Algorithms and expressivity of representation that means: “DLs systems need booth expressive logics and fast reasoner procedures for deciding subsumption (or equivalently satisfiability) in such logics have discouragingly high worst-case complexities, normally exponential with respect to the problem size” [3].

In this paper, we present a new representation approach based on description logics, which simplify the implantation of DLs systems using vectorisation of concept definition after an automated investigation of direct subsumption relations. This representation replaces the complex expressions by an attribute that we called Subsumption Hierarchical Attribute *SHA*.

The reminder of this paper figures like following: a background speaking about motivations and defining briefly the field of study with a brief subsection illustrating related works, the second section is called contribution; where we describe the proposed approach, and the third represent the study of the convergence of reasoning algorithm, finally we discuss advantages and weaknesses of the proposed approach.

II. MOTIVATIONS

DLs systems are really presented like a strong representation formalism but it allows some difficulties that can be rich in some cases the undecidability problem. Seeking for more expressivity to define models of an existing reality leads to complex manipulations which discourage the works on reasoning Algorithms implementations.

The smalls augmentation on simple DLs language can increase the complexity of any deduction procedure, that means “even a seemingly simple addition to a very small language can lead to subsumption determination becoming NP-hard.”[4]

Also it is proved that “adding the ability to represent equalities of role compositions makes the complexity of the subsumption problem leap from quadratic to undecidable.”[5]

As result of these remarques thier are three possible responses wich are :

- Provide an incomplete implementation of the DL reasoner, in the sense that there are inferences sanctioned by the standard semantics of the constructors that are not performed by the algorithm.
- Provide a complete implementation of a specific DL reasoner, acknowledging that in certain circumstances it may take an inordinate amount of time.
- Carefully devise a language of limited expressive power for which reasoning is tractable, and then provide a complete implementation for it.[6]

These difficulties are reflecting on implementation. The objectif of performing DLs systems leads to think about conception of representative structure. This lasts must be simple in creation and use, and also powerly expressive.

III. DLS SYSTEMS PERFORMING

In the current generation of DL-KRS, the need for complete algorithms for expressive languages has been the focus of attention. The expressiveness of the DL language required for reasoning on data models and semi-structured data has contributed to the identification of the most important extensions for practical applications.[7]

The need of complete reasoning Algorithms, acting on expressive DLs knowledge bases has introduced to lot of optimization techniques. These two properties represent the two majors' objectives in DL-KRS implementations. The completeness represents the ability to deduce all true facts using the rules on system. Where the expressiveness, represent the power of a representation to characterize knowledge.

Before designing and implementing a DL-based knowledge representation system, the implementers should be clear about the goals that they are trying to

meet and against which the performance of the system will ultimately be measured.[3]

For this fact testing and optimizing DL-KRS is relative to the goals of implementation. It is why, it is to be admitted that the primary goal is the utility in the developed application.

But the generic goals in performing DLs system are the efficiency and effectiveness. This first one rises in the use of memory and the time customization, where the second is measured by the ability to explicit implicitly represented knowledge in KB.

This proposed approach has as goal to create a complete efficient reasoning algorithm using vectorial representation derived from the inspection of direct subsumption relationship in DLs knowledge base.

IV. SUBSUMPTION REASONING IN DLS

Reasoning with DL knowledge bases is a deduction process which extracts not only the facts explicitly asserted in a knowledge base but also their logical consequences as well [8]. This deduction makes explicit the hidden subsumption relationship between two computable concepts using structured reasoning Algorithms, then as consequence the determination of the hierarchy of concepts defined in a TBox.

The advantage of these Algorithms is that by pre-processing concept descriptions and storing them in a normal form [9], subsequent subsumption (subsuming) tests, of which many will be required to maintain the hierarchy concept that can be performed relatively and efficiently [10].

Disadvantages are that the structural comparisons required to deal with more expressive description languages become extremely complex [11]. Although it has a negative effect, it is known that it can achieves more expressivity and completeness for the reasoning Algorithm by transforming the subsumption problem to non-satisfiability problem. “The satisfiability... problem can then be solved using a provably sound and complete algorithm based on the tableaux calculus” [12].

Applying some simple optimization techniques to a tableau algorithm can result in performance, which is comparable with incomplete structural algorithms [13]. The reason of this work is to investigate the impact of a simplified representation on the performance, because this representation can carry the meaning of subsumption and consequently the one of the satisfiability considered as in a finite domain.

It is to be noted that it is very simple to speak about assertions Box because it define for each simple predicate -the concept- in which every individual is occurring and for each binary predicate -roles- every pair of individual who satisfy the role definition.

V. RELATED WORKS

The searchers in the field are working for the DLs systems optimization but we can't find in literature works

on vectorial form derived from the subsumption relationships. This fact makes the novelty presented in this paper.

The efficiency of reasoning algorithms is viewed like a problem of optimization that reduces the complexity of an algorithm and gives the same or better level of performance. In literature, we find some works aiming at the same objectives but touching different aspects as: Nenad Krdžavac and his collaborators, in 2009 declare “The implementations are not based on original definition of the abstract syntax, but they require transformation of the abstract syntax into concrete syntax implementation languages use]. [The use of model-driven engineering principles for the development of a DL reasoner where a definition of a DL abstract syntax is provided by means of metamodels” [14]. We find also the work Hustadt with his groupe in 2007 “developed a novel reasoning algorithm that reduces a SHIQ knowledge base to a disjunctive Datalog program while preserving the set of ground consequences” [15]. And recently, in 2012, the work presented by Cuenca with his collaborators “proposed a theoretical framework and practical techniques for establishing formally provable and Algorithmically variable completeness guarantees for incomplete ontology reasoners” [16].

VI. CONTRIBUTION

A. Subsumption Hierarchical Attribute

The SHA represents a vectorial attribute that can carry the subsumption meaning, each component represents a nodes of the tree of subsumption hierarchy.

Every SHA vector gives the path from the root to the node representing the concept, it’s why it is possible to find one concept defined with more than one SHA vector.

It is known that we can reduce every equivalence relation to subsumption: A is equivalent to B if and only if A is subsumed by B and B is subsumed by A (A and B represent concepts). By this way we can construct the dependency graph of subsumption.

$$D \sqsubseteq C \quad (1)$$

$$B \sqsubseteq C \quad (2)$$

$$A \sqsubseteq D \quad (3)$$

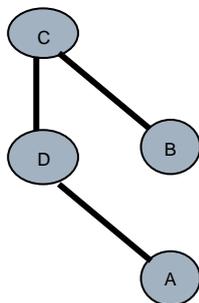


Fig.1. The subsumption hierarchical structure

Then, every node will have a number by sequential codification that begins by “1” for each level. The path from root (more general concept) to the concerned node makes the representative vector using this numbers. See figure 2.

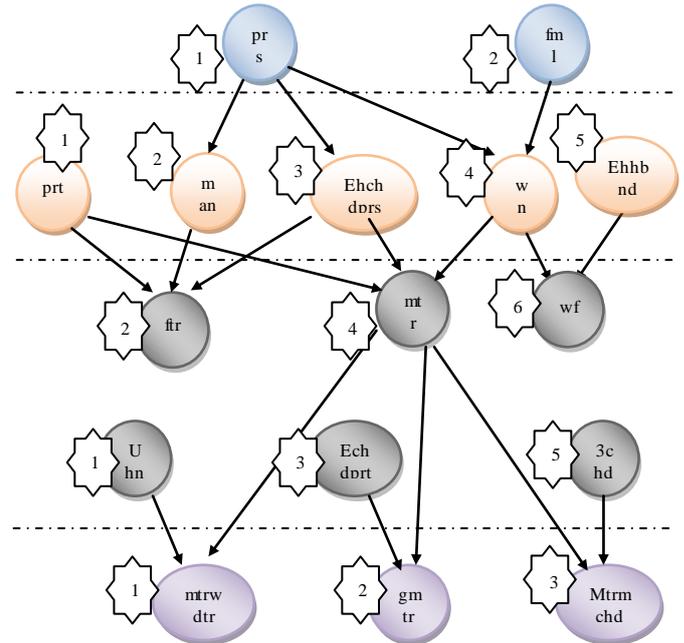


Fig.2. The subsumption hierarchical indexation

This table illustrate names of nodes in figure2

Table 1. Abbreviation for human being knowledge bases

Abbreviation	Concept name
Prs	Person
Fm	Female
Prt	Parent
Man	Man
Echdprs	\exists haschild. person
Wn	woman
Ehhbnd	\exists hashusband
Ftr	Father
Mtr	Mother
Wf	Wife
Uhwn	\forall haschild. woman
Echdprt	\exists haschild. parent
3chd	≥ 3 haschild
Mtrwdtr	Mother without daughter
Gmtr	Grandmother
Mtrmchd	Mother with many children

B. Automated Generation Process

The first step of the SHA generation process is the direct subsumption detection which consists in representing the axioms defined by equivalence relation in subsumption based form. This representation is based on implications existing between the two kinds of representation. Like sets treatment each definition using

constructors can be reduced to a group of subsumption (inclusion) relation as follows.

$$C \equiv D \cap B \Rightarrow (C \subset D) \wedge (C \subset B) \quad (4)$$

$$C \equiv D \cup B \Rightarrow (D \subset C) \wedge (B \subset C) \quad (5)$$

$$C \equiv \neg B \Rightarrow B \cap C \equiv \Phi \quad (6)$$

Reducing the complex expressions to simple subsumption expressions needs some iteration based on other rules after the application of this specified below as initial treatments. For reasons of expressiveness, indicators, to represent the infinite and the finite sub-domain, are needed. The infinite sub-domain comes from the relations of subsumption between a concept and an expression like

$$C \subset (D \cup B) \quad (7)$$

which means that there is a part of a set which can be a small part or big part or the whole of the super set. This probable forms cause less expressive representation and

make useful the use of the infinity indicators for each sub-domain.

$$(D \cap B) \subset C \Rightarrow \exists A, (D \cap B) \cap C \equiv A \Rightarrow (A \subset D) \wedge (A \subset B) \wedge (A \subset C) \quad (8)$$

$$(D \cup B) \subset C \Rightarrow (D \subset C) \wedge (B \subset C) \quad (9)$$

$$C \subset (D \cap B) \Rightarrow (C \subset D) \wedge (C \subset B) \quad (10)$$

$$C \subset (D \cup B) \Rightarrow (C \subset D) \vee (C \subset B) \vee (C \subset (D \cap B)) \quad (11)$$

For the roles level, the role is seen as binary relation between two sets of individuals. The expression $R.C$ corresponds to the individuals in relation with C by R . For example, the sentence “individuals having a female child” is presented $\exists \text{hasChild. Female}$, and the sentence “individuals all of whose children are female” is presented $\forall \text{hasChild. Females}[03]$. By the same way the number restriction is represented to integrate cardinalities. (See figure03)

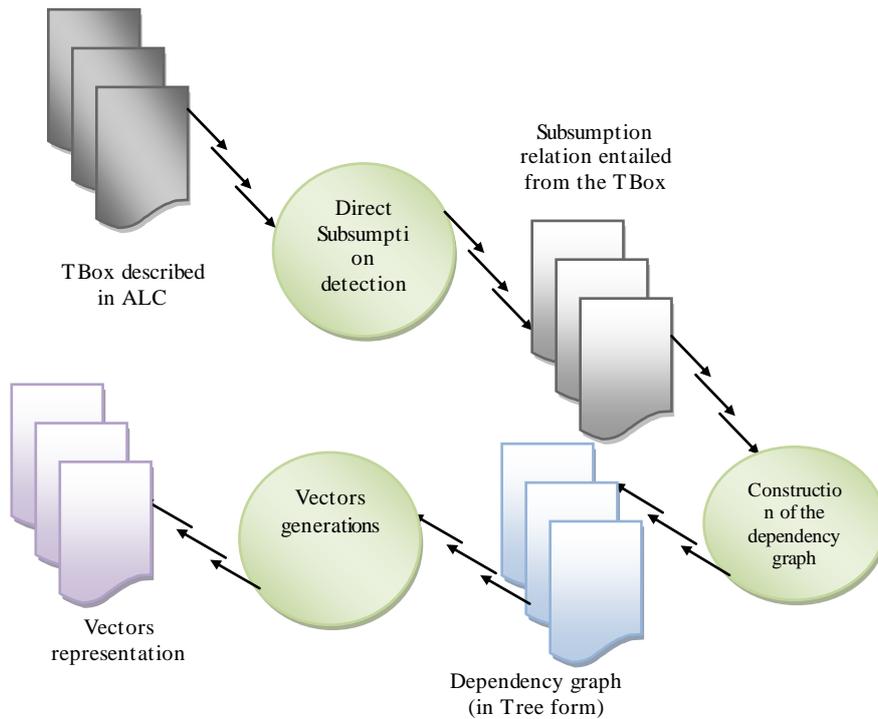


Fig.3. The SHA generation process

This proposed process is completely computable with high effectiveness degree. But it's to be mentioned that for implementation we use a reduced syntax.

C. Automated Process

To accomplish this task, a simplified syntax has been

used. The implemented interface analyse the introduced text respecting the polish notation generated by the proposed reduced syntax and generate SHA vectors after inspecting the direct subsumption relation between concepts.

$S \rightarrow \% OP / C$
 $OP \rightarrow CON(S,S) / Neg(S) / T / R.C/$
 $T \rightarrow EX / UN / <=Number / >=Number$ The terminal alphabet is giving like: Conj for conjunction, Disj for Disjunction,
 $CON \rightarrow Conj / Disj$
 $C \rightarrow <concept name>$
 $R \rightarrow <role name>$

Using this rule the application is made to provide an interface for integration and generation of SHA like in following figures.

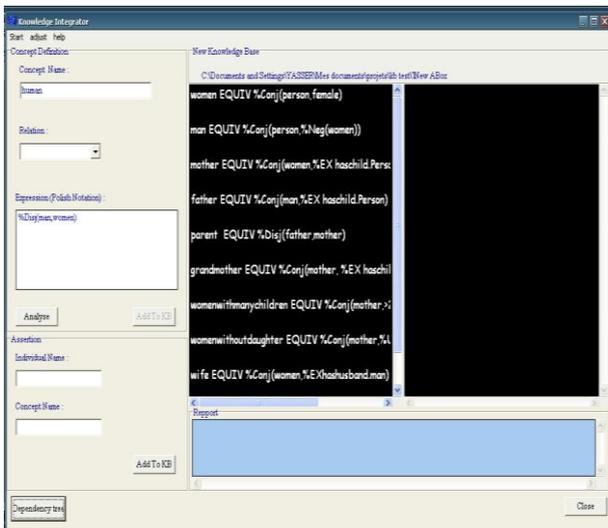


Fig.4. The main frame of the SHA generator soft ware.

After being analyzed and add line by line the resulted text representing a part of the knowledge base is treated to inspect the direct subsumption relation between concepts. This task is necessary to generate SHAs. (see figure 5)

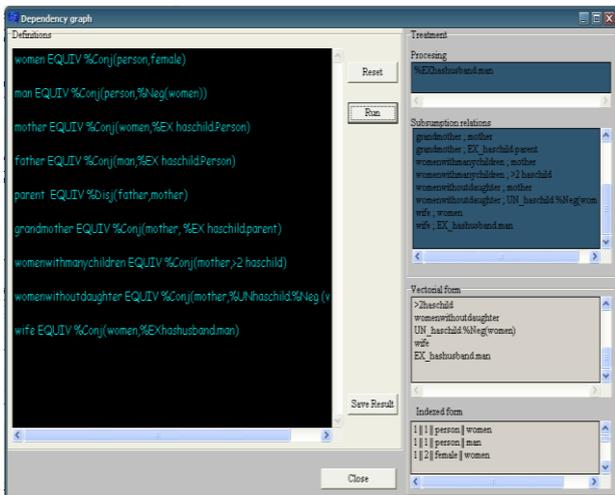


Fig.5. Indexed hierarchy generator

D. Rules in Vectorial Form

In this section we present how to adapt the reasoning rules on the new representation structures. Each component in SHA has a signification thing that makes the deduction process based on vectors comparing.

In follows the Tableau ALC rules can be transformed to deal with the vectorial representation with graphical illustrates.

The \cap -rule

If \mathcal{A} contains $(C_1 \cap C_2)(x)$, but not both $C_1(x)$ and $C_2(x)$ then

$$A' \equiv A U \{C_1(x), C_2(x)\} \quad (12)$$

Let's suppose \mathcal{A} , C_1 and C_2 concepts represented by $V(a_1, a_2, \dots, a_p)$, $V_1(c_1, c_2, \dots, c_p)$, and $V_2(b_1, b_2, \dots, b_p)$

\mathcal{P} is the constructed by the conjunction between C_1 & C_2 which yields to two vectors $(c_1, \dots, c_k, 0, \dots, 0)$ and $(b_1, \dots, b_k, 0, \dots, 0)$ with $c_k = b_k$.

Condition

If we find another concept \mathcal{A} with vector V of the form $(a_1, \dots, a_k, 0, \dots, 0)$ where $a_k = c_k = b_k$

Action

1. Create a node \mathcal{A}' with representative vector of the form $(a_1, -1, \dots, -1, t, 0, \dots, 0)$ where t-1 is number of concepts in level k-1.
2. Add new vectors to define \mathcal{A} , C_1 and C_2 like follows

$$\begin{aligned} \mathcal{A} &\rightarrow (a_1, -1, \dots, -1, t, a_k, \dots, 0) \\ C_1 &\rightarrow (a_1, -1, \dots, -1, t, c_k, \dots, 0) \\ C_2 &\rightarrow (a_1, -1, \dots, -1, t, b_k, \dots, 0) \end{aligned}$$

And by the way each concept subsumed by \mathcal{A} , C_1 or C_2 must have new vector defined by taking the left side of this three vectors.

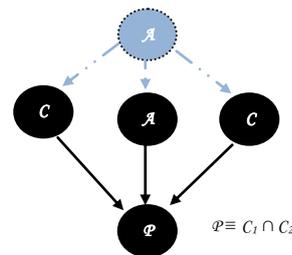


Fig.6. Representation of the rule of conjunction applied to the vectorial form

The U -rule

If \mathcal{A} contains $(C_1 \cap C_2)(x)$, but neither $C_1(x)$ nor $C_2(x)$ then

$$A' \equiv A U \{C_1(x)\} \text{ and } A'' \equiv A U \{C_2(x)\} \quad (13)$$

Let's suppose \mathcal{A} , C_1 and C_2 concepts represented by $V(a_1, a_2, \dots, a_p)$, $V_1(c_1, c_2, \dots, c_p)$ and $V_2(b_1, b_2, \dots, b_p)$

\mathcal{P} is the constructed by the disjunction between C_1 & C_2 which yields to two vectors $(a_1, \dots, a_k, a_k, a_{k+1}, \dots, 0)$ and

$(a_1, \dots, a_{k-1}, a_k, b_{k+1}, \dots, 0)$ representing C_1 and C_2 respectively

Condition

If \mathcal{A} is represented by a vector in the form $(a_1, \dots, a_{k-1}, 0, \dots, 0)$ & \mathcal{P} is represented by a vector of the form $(a_1, \dots, a_{k-1}, a_k, 0, \dots, 0)$ and we find C_1 and C_2 having respectively vectors of the form $(a_1, \dots, a_{k-1}, a_k, c_{k+1}, \dots, 0)$, $(a_1, \dots, a_{k-1}, a_k, b_{k+1}, \dots, 0)$.

Action

1. Create two new nodes \mathcal{A}' and \mathcal{A}'' with vectors $(a_1, -1, \dots, -1, t, 0, \dots, 0)$ $(a_1, -1, \dots, -1, t+1, 0, \dots, 0)$ while $t-1$ is number of concept existing in level $k-2$.
2. Add new vectors to define \mathcal{A} , C_1 and C_2 like follows

$$\begin{aligned} \mathcal{A} &\rightarrow (a_1, -1, \dots, -1, t, a_k, 0, \dots, 0), (a_1, -1, \dots, -1, t+1, a_k, 0, \dots, 0) \\ \mathcal{P} &\rightarrow (a_1, -1, \dots, -1, t, a_k, a_k, 0, \dots, 0), (a_1, -1, \dots, -1, t+1, a_k, a_k, 0, \dots, 0) \\ C_1 &\rightarrow (a_1, -1, \dots, -1, t, -1, -1, c_{k+1}, \dots, 0), (a_1, -1, \dots, -1, t+1, -1, -1, c_{k+1}, \dots, 0), (a_1, -1, \dots, -1, t, a_k, -1, a_k, c_{k+1}, 0, \dots, 0), (a_1, -1, \dots, -1, t, a_k, -1, a_k, c_{k+1}, 0, \dots, 0) \\ C_2 &\rightarrow (a_1, -1, \dots, -1, t+1, -1, -1, b_{k+1}, \dots, 0), (a_1, -1, \dots, -1, t+1, -1, -1, b_{k+1}, \dots, 0), (a_1, -1, \dots, -1, t+1, a_k, -1, a_k, b_{k+1}, 0, \dots, 0), (a_1, -1, \dots, -1, t, a_k, -1, a_k, b_{k+1}, 0, \dots, 0) \end{aligned}$$

By this same method we define new vectors for every subsumed concepts derived from this concepts

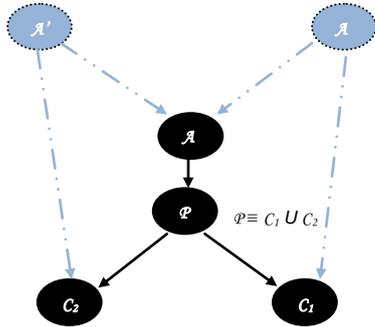


Fig.7. Representation of the rule of disjunction applied to the vectorial form

The \exists -rule

If \mathcal{A} contains $(\exists R.C)(x)$ but there is no individual name z such that $C(z)$ and $R(x,z)$ are in \mathcal{A} then

$$A' \equiv A \cup \{C(y), R(x, y)\} \quad (14)$$

where y is an individual name not occurring in \mathcal{A}

In this rule the condition specified the individuals z such as $R(x,z)$ occurring in C but not in A which contains the concept defined by $(\exists R.C)(x)$. This leads to a vectorial definition like in follows.

Note that when the condition is verified the actions need to create some new nodes to represent the sub concepts of C and A .

Lets define $A_1 = \{a, A(a) \wedge R(x,a) \text{ with } C(x)\}$ and $A_2 = \{y, C(y) \wedge R(x,y) \text{ with } c(x) \wedge \neg A(y)\}$ then we imagine when A and C are represented by $(a_1, \dots, a_k, 0, \dots, 0)$ and $(a_1, a_2, \dots, a_k, 0, \dots, 0)$

Condition

If there is no individual z occurring in C and not in A having relation with x such as $R(x,z)$, which means an individual occurring in A_2 .

Action

Create A' $(a_0, -1, \dots, -1, t, 0, \dots, 0)$ where $(t-1)$ is the number of concept in level $(k-1)$ and $A' \equiv A \cup A_2$, add new vectors in definitions of A, A_2 and there sub concepts are like in precedent.

$$\begin{aligned} \mathcal{A} &\rightarrow (a_1, -1, \dots, -1, t, a_k, 0, \dots, 0) \\ \mathcal{A}_2 &\rightarrow (a_1, -1, \dots, -1, t, a_k, -1, -1, a_k, 0, \dots, 0) \end{aligned}$$

With this same method we define new vectors for every concept subsumed by all concepts having new representatives vectors.

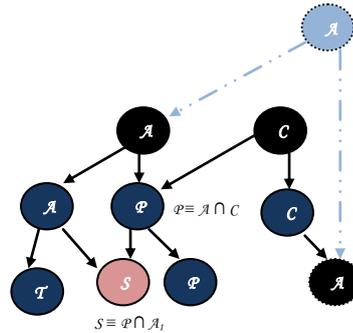


Fig.8. Representation of the rule of existential quantification applied to the vectorial form

The \forall rule

If \mathcal{A} contains $(\forall R.C)(x)$ and $R(x,y)$, but it does not contains $C(y)$ then

$$A' \equiv A \cup \{C(y)\} \quad (15)$$

We suppose that $\mathcal{A}(a_1, \dots, a_k, 0, \dots, 0)$ and $C(a_0, a_1, \dots, a_k, 0, \dots, 0)$

The concept $\mathcal{A}_1 = \{x, \mathcal{A}(x) \wedge R(x,y) \text{ with } C(y)\}$

Condition

If \mathcal{A} contains \mathcal{A}_1 and not $C(y)$

Action

Create $\mathcal{A}'(a_1, -1, \dots, -1, t, 0, \dots, 0)$ while $(t-1)$ is number of concept in level $(k-1)$

Add vectors for \mathcal{A} and C and all of their subsumed concepts

$$\begin{aligned} \mathcal{A} &\rightarrow (a_1, -1, \dots, -1, t, a_k, 0, \dots, 0) \\ \mathcal{C} &\rightarrow (a_1, -1, \dots, -1, t, a'_k, 0, \dots, 0) \end{aligned}$$

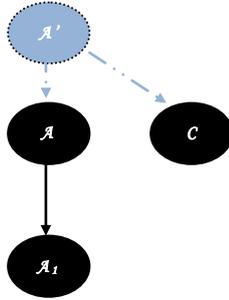


Fig.9. Representation of the rule of universal quantification applied to the vectorial form.

VII. CONVERGENCE OF THE ALGORITHM

Every one of the four rules is conditioned with binary condition (true /false). The rule figure in form of `if condition then action` every action makes new *SHA* to represent new nodes. This means that the probability of creating new *SHAs* is a Bernoulli distribution. And by iteration it becomes Binomial distribution. That means that we can calculate the number of additional *SHAs* by the function f like in follows.

$$f(x_1, x_2, x_3, x_4) = N_1 \cdot T_1(x_1) + N_2 \cdot T_2(x_2) + N_3 \cdot T_3(x_3) + N_4 \cdot T_4(x_4) \quad (16)$$

&
 $p = 1 - q$

Where

$$T_i(x) = C_x^n \cdot P^x \cdot (1 - P)^{n-x} \text{ With } 0 < x < n,$$

x_i success number; it represents the case when conditions of rule is satisfied,
 n for number of iteration (experiments).

N_i represent number of *SHAs* created by the rule i when condition is satisfied.

$$C_x^n = \frac{n!}{x!(n-x)!} \quad (17)$$

Then

$$T_i(x) = \frac{n!}{x!(n-x)!} \cdot P^x \cdot (1 - P)^{n-x} \quad (18)$$

To prove that the Algorithm will attend a stop point.

Now we accept that if all $T_i(x)$ converge f converge also. And the limit of the augmentation function must be equal to "0" to riche stability case in which the inference stops.

$$\lim_{x \rightarrow +\infty} T_i(x) = \lim_{\substack{x \rightarrow +\infty \\ n \geq x}} \left(\frac{n!}{x!(n-x)!} \cdot P^x \cdot (1 - P)^{n-x} \right) \quad (19)$$

To prove the convergence, we know that $0 \leq x \leq n$ that means that if $x \rightarrow +\infty, n \rightarrow +\infty$ because if we like to have a big x we must have a bigger n

Then

$$\begin{aligned} T_i(x) &= \frac{n!}{x!(n-x)!} \cdot P^x \cdot (1 - P)^{n-x} \\ &= \frac{n!}{(n-x)!n^x} \cdot \frac{(nP)^x}{x!} \cdot \frac{(1-P)^n}{(1-P)^x} \end{aligned} \quad (20)$$

We accept these three additional constraints

- 1- $\lim_{n \rightarrow \infty} \frac{n!}{(n-x)!n^x} = 1$
- 2- $\lim_{P \rightarrow 0} (1 - P)^x = 1$
- 3- $\log(1 - P)^n = n \log(1 - P)$ when $n \rightarrow \infty$ and $P \rightarrow 0$.

We deduce that

$$\lim_{\substack{n \rightarrow \infty, \\ P \rightarrow 0, \\ nP \rightarrow \lambda}} (1 - P)^n = e^{-\lambda} \quad (21)$$

As conclusion

When $n \rightarrow +\infty, p \rightarrow 0$ and $nP \rightarrow \lambda$ where

$$\frac{n!}{x!(n-x)!} \cdot P^x \cdot (1 - P)^{n-x} \quad (22)$$

Can be approximate to

$$e^{-\lambda \frac{(\lambda)^x}{x!}} \quad (23)$$

Now, $\rightarrow +\infty, (0 \leq x \leq n)$

$$\lim_{\substack{x \rightarrow +\infty \\ n \geq x}} T_i(x) = \lim_{\substack{x \rightarrow +\infty \\ n \geq x}} e^{-\lambda \frac{(\lambda)^x}{x!}} = 0 \quad (24)$$

We know that if every $T_i(x) \rightarrow 0$ means that the Algorithm achieves stability. Other way the limit "0" means that there is no inference, no augmentation or no new deductions which means the case of stability.

VIII. DISCUSION

The advantages of this approach is that; in one hand, it enables an easier implementation, it makes the reasoning Algorithm a simple process manipulating significant vectors, also the generation of the *SHAs* is computable process. In the other hand, intuitively that effectiveness will not decrease and efficiency will increase because of the use of vectors comparing which makes the algorithm leas complex.

The disadvantages are that at this time we can't affirm that the *SHAs* are expressive because it is known that reducing equivalences to subsumption relationships decreases the expressivity level. But also we can't affirm in the other hand that the representation with *SHAs* is leas expressive thing which is let for future study.

IX. CONCLUSION

In this work we presented the SHA like new representation able to carry the semantic aspect of concept definition in ALC Languages. The vectorial form leads to simplified form of manipulation which yields to easy implementation of reasoning algorithm. The aim achieved here is the efficiency at the level of algorithms by acting on the representation.

Computable generation process of this new representation and simplicity of manipulation of representatives vectors encourage the investigation on DLs KRS respecting this proposed approach.

The future of this proposed approach can be decided after implementation of DL system based on SHA to enable searchers to test and check all aspects like completeness and expressivity which are the main aimed goals in this field.

REFERENCES

- [1] Souleymane Koussoube, Roger Noussi, Balira O. Konfe, «Using Description Logics to specify a Document Synthesis System» I.J. Intelligent Systems and Applications, 2013, 03, 13-22.
- [2] F. V. Harmelen, and al; «Handbook of knowledge representation »P135 Elsevier 2008.
- [3] Franz Baader, and al; «the description logics handbook: theory, implementation and application »P 345 cambredge university press 2003.
- [4] Levesque, H. J., & Brachman, R. J. (1987). Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3 (2), 78{93.
- [5] Patel-Schneider, P. F. (1989). Undecidability of subsumption in NIKL. *Artificial Intelligence*, 39 (2), 263{272.
- [6] Alex Borgida & Peter F. Patel-Schneider; (1994) “A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Logic”, *Journal of Artificial Intelligence Research* 1 (1994) 277 308.
- [7] D. Nardi, R. J. Brachman. An Introduction to Description Logics. In *The Description Logic Handbook*, edited by F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, Cambridge University Press, 2002, pages 5-44.
- [8] Sergio Tessaris; “questions and answers: reasoning and querying in description logic” a PhD Thesis at university of Manchester the faculty of science and engineering, April 2001.
- [9] Ian Horrocks. “Optimising Tableaux Decision Procedures For Description Logics”. A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Science and Engineering. University of Manchester, 1997 (p20).
- [10] A. Borgida and P. F. Patel-Schneider. “A Semantics and Complete Algorithm For Subsumption In The Classic Description Logic”. *Journal of Artificial Intelligence Research*, 1:277{308, 1994.
- [11] Ian R. Horrocks. *Optimising Tableaux Decision Procedures for Description Logics*. A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Science and Engineering. 1997.
- [12] R. M. Smullyan. *First-Order Logic*. Springer-Verlag, 1968.

- [13] F. Baader, B. Hollunder, B. Nebel, and H.-J. Pro Ulrich. An empirical analysis of optimization techniques for terminological representation systems. In B. Nebel, C. Rich, and W. Swartout, editors, *Principals of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR'92)*, pages 270{281. Morgan-Kaufmann, 1992. Also available as DFKI RR- 93-03.
- [14] Nenad Krdžavac, Dragan Gašević, and Vladan Devedžić; “Model Driven Engineering of a Tableau Algorithm for Description Logics” *ComSIS Vol. 6, No. 1, June 2009*.
- [15] B. Cuenca Grau, B. Motik, G. Stoilos and I. Horrocks; “Completeness Guarantees for Incomplete Ontology Reasoners: Theory and Practice”, 2012, Volume 43, pages 419-476.
- [16] Ullrich Hustadt, Boris Motik, Ulrike Sattler; “Reasoning in Description Logics by a Reduction to Disjunctive Datalog” *Journal of Automated Reasoning* October 2007, Volume 39, Issue 3, pp 351-384.

Authors' Profiles



Yasser Yahiaoui has completed ING diploma from UDL Sidi bel Abbes – Algeria- (2005), with final curriculum project on neural networks mapping for OCRs. MAGISTER (magisterium diploma) from Bechar university –Algeria- (2008) on subject of artificiel intelligence at the service of edicational software, he returne to UDL for preparing doctorat of science from 2009 till now in artificial intelligence and knowledge ingneering.

Yasser is a member of info-team a team of searcher on computer science of university centre of Naama – Algeria- and a member of the Knowledge Engineering Team at the EEDIS laboratory. He works as teacher assistant at university centre of Naama -Algeria- where he is the head of teaching team of informatics specialty.



Ahmed Lehireche has completed respectively ING Diploma from ESI of Algiers (1981) with the final curriculum project at the IMAG (France), MAGISTER Diploma from USTOran (1993) and RDOCTORAT D'ETAT Diploma from UDL Sidi bel Abbes (2005). He is working as a Director of research, head of the Knowledge Engineering Team at the EEDIS laboratory and full Professor at the computer science department of UDL Sidi bel Abbes. He is mainly concerned with AI, Computer Science Theory and Semantics in IT.

Ahmed's name figures as author on lot of scientific papers concerned in his research areas which are knowledge engineering, web engineering, artificial intelligence sf.



Djelloul Bouchiha received his Engineer degree in computer science from Sidi Bel Abbes University, Algeria, in 2002, and M. Sc. in computer science from Sidi Bel Abbes University, Algeria, in 2005, and Ph. D. in 2011. Between 2005 and 2010, he joined the Department of Computer Science, Saida University, Algeria, as a Lecturer. He

became an Assistant Professor since January 2011. Currently, he is an Assistant Professor at the University Center of Naama. His research interests include semantic web services, web reverse-engineering, ontology engineering, knowledge management and information systems.

How to cite this paper: Yasser Yahiaoui, Ahmed Lehireche, Djelloul Bouchiha, "Proposed Representation Approach Based on Description Logics Formalism", *International Journal of Intelligent Systems and Applications (IJISA)*, Vol.8, No.5, pp.1-9, 2016. DOI: 10.5815/ijisa.2016.05.01