

## Chapter 2

# Representing Knowledge Effectively Using Indian logic

*Mahalakshmi G.S. and Geetha T.V. \**

**Abstract:** Knowledge becomes the key factor for intelligence. Representing knowledge is the fundamental requirement for Inference and reasoning mechanisms. Inferences will prove efficient only when knowledge is represented and retrieved more naturally. In other words, the pattern of knowledge representation should match human way of knowledge representation to enable mimicking of human inferences. Every research in artificial intelligence had proposed striking advances in knowledge representation mechanisms. This paper discusses such an aspect of knowledge representation adapted from Indian philosophy. The paper also presents a short comparison with other knowledge representation techniques.

---

**Keywords:** Logic, Knowledge representation, Indian philosophy, Nyaya Sastra, Ontology

### 1. INTRODUCTION

One of the fundamental issues in artificial intelligence is the problem of knowledge representation. Intelligent machines must be provided with a precise definition of the knowledge that they possess, in a manner, which is independent of procedural considerations, context-free, and easy to manipulate, exchange and reason about. Any comprehensive approach to knowledge representation has to take into account the inherently dynamic nature of knowledge. As new information is acquired, new pieces of knowledge need to be dynamically added to or removed from the knowledge base.

For inferences, decision-making, dialogue-exchange or machine learning, the fundamental issue involved is the utilisation of reasoning. Reasoning is the process of arriving at new conclusions. To reach a conclusion, we generally conclude certain investigations. Therefore, if the investigations are not formally represented using a knowledge representation language which is clear and user-friendly, performing reasoning shall become a daunting task.

Lau et al (2003) have outlined a research agenda for automatically acquiring procedural knowledge for use in autonomic systems. It is based on learning procedures by observing experts perform these procedures on live systems, and dynamically building a procedural model that can be executed on a new system to repeat the same task. Over time, as more traces are collected, the procedural model is

---

\* Department of Computer Science and Engineering, Anna University, Chennai 600025, India  
mahalakshmi@cs.annauniv.edu, tvgeedir@cs.annauniv.edu

updated. This dynamic learning process also enables the procedure to adapt to changing conditions over time.

A Learnable behavioral model for autonomous virtual agents (Toni and Daniel 2006) and a knowledge delivery service has also been proposed. At this juncture, social learning in robots is worth-mentioning. A reinforcement learning model which deals with human teachers teaching robots has been proposed (Thomaz et al 2006). Here, the learner robots contribute to learning by revealing their internal states to help guide the teaching process. Teacher and learner read and respond to each other, to more effectively guide the learner's exploration.

Virtual communities are relatively a new approach to distributed knowledge sharing. Research in virtual communities show an increase in the need to share knowledge across a distributed environment in present web-based lifestyle. Setting-up successful virtual communities represents an effective way for facilitating the circulation of knowledge in organizations and groups. Perceptual, deliberative, operational, and meta-cognitive means of attention management for virtual community members have also been attempted. Use of Ontology facilitates the sharing of world knowledge between virtual communities (Davies et al 2004). Dynamic matching of distributed ontologies facilitates co-operative learning in virtual knowledge communities.

In a nutshell, with the rise of intelligent front ends to AI systems, it has become conventional to represent the inputs in natural languages and thereafter, the world knowledge embedded in the input is extracted with the help of common sense ontologies. The reason is to reduce the information overload to AI systems by avoiding extensive pre-processing. However, introduction of ontologies to AI systems have a great impact on the performance and accuracy of operation of most systems. Practical application of Ontology based systems in real-time was quite challenging in spite of the benefits of Ontology to modern AI society.

## 2 ONTOLOGY AND KNOWLEDGE REPRESENTATION

In the context of multiple entities participating in knowledge sharing, a common Ontology can serve as a knowledge-level specification of the ontological commitments of a set of participating entities that are involved in discussion. Ontology is a formal mechanism of representing the world knowledge, out of which effective and easy reasoning is possible during knowledge sharing. Ontological commitments are agreements that define a clear boundary of how to *view* the Ontology. This means that the specification format is independent of the internal representation such that the ontological commitments are defined only at the knowledge level.

Inconsistencies are conflicting information present within the ontologies which inhibit reasoning and inference procedures. Inferences made with such approaches will never be adequate enough to better reasoning while knowledge sharing. The most challenging and important problem is *Ontology evolution*, which is the problem of modifying an Ontology in response to a certain change in the domain or its conceptualization. As Flouris et al (2006) summarises, there are several cases where Ontology evolution is applicable: change in the world knowledge, change in the domain, change in the specification or conceptualisation, change in the perspective, change in user needs or change in belief. This is the strong reason behind our motivation for identifying improvements in knowledge representation.

### 3 KNOWLEDGE REPRESENTATIONS – PROPERTIES AND CHALLENGES

#### 3.1 Properties of knowledge representation

Knowledge representation cannot be defined in pure epistemological terms. Representation and reasoning are intertwined with knowledge representation. The attempt to deal with representation as knowledge content alone leads to an incomplete conception where reasoning may be put aside. The use of a representation as a medium of expression and communication matters because we must be able to speak the language in order to use it. If we cannot determine how to say what we are thinking we cannot use the representation to communicate with the reasoning system. Several measures of a good knowledge representation may be listed as follows:

- Support to efficient reasoning
- Expressivity – how expressive the knowledge is
- Adequacy – is the represented knowledge adequate
- Satisfiability – role of knowledge which satisfies the goal
- Quality – quality of knowledge within the knowledge representation
- Uncertainty – how much certain the expressed knowledge is
- Consistency – how much consistent the knowledge is

#### 3.2 Challenges of Knowledge representation

The principal notion of a good knowledge representation should be to understand and describe the richness of the world. Apart from this, other challenges are as follows:

- to cope with dynamism in the world knowledge
- to preserve consistency of information across domains
- to accept belief revisions in the knowledge covered under representation
- to represent beliefs in a easy and resourceful manner
- to facilitate better reasoning and inferences over the represented knowledge
- to enable changes from the knowledge definition perspective
- to adapt to addition of necessary information with change in the specification or conceptualization.

The above notion can be better understood by describing more abstractly, the five roles of knowledge representation (Davis et al 1993):

- A knowledge representation is most fundamentally a surrogate; used to enable an entity to determine consequences by thinking rather than acting.
- It is a set of ontological commitments
- It is a fragmentary theory of intelligent reasoning expressed in terms of three components; (1) the representation is fundamental conception of intelligent reasoning (2) the set of inferences the representation sanctions and (3) the set of inferences it recommends.
- It is a medium for pragmatically efficient computation
- It is a medium of human expression; i.e. a language in which we say things about the world

## 4 VARIETIES OF KNOWLEDGE REPRESENTATION

In the field of AI, many researchers have addressed the problem of knowledge representation; in this area semantic networks played an important role. In semantic networks the knowledge is described by nodes and links. While Quillian aimed at the representation of word meanings [Quillian 68], semantic networks have also been used to model propositions, events, spatial relationships and so on. Since semantic networks failed in providing a unique semantic interpretation, several researchers examined the "semantics of semantic networks" (Woods 75, Brachman 79).

### 4.1 Production Systems

Production systems (Newell, 1973) are a form of knowledge representation which provides for a good deal of flexibility. Productions are condition-action pairs. The set of productions by itself is quite unstructured. To make it work it needs two kinds of control processes. One is a short-term memory buffer: only the data currently held in that buffer can activate the condition of a production; thus the flow of data in and out of short-term memory determines in part what productions are executed. However, since it will frequently be the case that more than one production condition matches the data in the short-term memory buffer, some kind of conflict resolution procedure is required. Production systems have their disadvantages, too. It is not easy to understand what actually happens in a large production system. The reason is that the interactions within productions may have surprising outcomes.

### 4.2 Associative networks

An associative network is a structure in which the nodes are unanalyzed concepts, and the links are unlabeled, but vary in strength. Knowledge is thus represented as a network of ideas, with interconnections determined by the laws of association like resemblance, contiguity, and cause-effect, to name a few. While associative nets are messy, perceptually based, semantic nets are orderly and conceptually based. There are different categories of semantic nets [Brachman 1979]. They are:

1. Logical nets: links represent logical relations and nodes are predicates and propositions
2. Epistemological nets: Links are inheritance and structuring relations and nodes are concept types, rather than particular concepts
3. Conceptual nets: Links are semantic relations; nodes are primitive objects or actions
4. Linguistic nets: Primitives are language-dependent; meanings are derived from context and it changes as the network grows

### 4.3 Frames

The major leap forward from semantic nets was towards more structured knowledge representation, called frames [Minsky 75]. A frame is a data structure that is typically used to represent a single object, or a class of related objects, or a general concept (or predicate). Frames are typically arranged in a taxonomic hierarchy in which each frame is linked to one parent frame. A collection of frames in one or more inheritance hierarchies is a knowledge base. Frames consist of a heading and various slots. Each slot has its default value which can be activated if no other information is available. Thus, as soon as a frame is invoked, a great deal of well-organized knowledge becomes available, without the need for elaborate computations.

The slots of a frame describe attributes or properties of the thing represented by that frame. In addition to storing values, slots also contain restrictions on their allowable values. Every slot has several

components like slot name, value, value restriction, procedure, justification etc. which are referred to as facets. Frames typically allow the specification of default slot values, perspectives and attached procedures. Collections of frames can be combined to frame-systems. The expressive power of frame systems makes it impossible to provide a well defined semantics for them. Both, elements of different network formalisms and basics of the frame theory, have influenced the KL-ONE like representations.

#### 4.4 Conceptual graphs

The basic building blocks of KL-ONE representations are "concepts", i.e. structured conceptual objects. "Roles" are possible relationships between two concepts. The subsumption relation organizes the concepts in concept taxonomy. Concepts are described with respect to their super concepts by restricting and differentiating roles. In particular, roles can be restricted by the number (number restriction) and the range (value restriction) of allowed role fillers. If the specified restrictions constitute necessary and sufficient conditions for the concept, it is called a defined concept, whereas primitive concepts only need necessary conditions. Classification, an important inference mechanism of KL-ONE like systems, inserts concepts at the correct place in the concept hierarchy.

A logical reconstruction of KL-ONE revealed that the semantic status of a number of notions of KL-ONE was rather unclear. TSLs are formal knowledge representation languages derived from KL-ONE providing well-defined semantics which enables the decision whether the inferences are sound and complete. A number of KR systems based on TSLs have been developed, for instance, Krypton [Brachman et al. 85], KL-Two [Vilain 85], Back [Peltason et al. 89], Loom [MacGregorBates 87]. Besides a component for defining concepts and reasoning about the relationships between concepts (terminological component, TBox) these systems include an assertional component (ABox) that allows the definition of assertions about individuals.

A conceptual graph is a network of concept nodes and relation nodes (Conceptual Graph Standard 2002). The concept nodes represent entities, attributes, or events (actions) while the relation nodes identify the kind of relationship between two concept nodes. The main characteristics of conceptual graphs are:

- Concepts and relations replace predicates and arguments from predicate logic
- A relation's signature defines what concept types it can relate
- Concepts allow referents to specify an individual or a set of individuals of a certain type
- A type hierarchy can be defined on concepts
- Different graphs can be related through a co reference link on an identical concept

#### 4.5 Description Logics

Description logics (DL) are both class-based and logic-based knowledge representation languages, which allow for modeling an application domain in terms of objects, classes and relationships between classes, and for reasoning about them (Baader et al 2002). Unlike object-oriented languages used in databases and programming languages, DL permit the specification of a domain by providing the definition of classes, and by describing classes using a rich set of logical operators. By using DL, one can specify not only the necessary conditions that objects of a given class must obey, but also the sufficient conditions for an object to belong to a certain class.

A knowledge base built using DL is formed by two components: the intensional one, called Tbox, and the extensional one, called ABox. The basic building blocks are concepts, roles and individuals. Concepts describe the common properties of a collection of individuals and can be considered as unary predicates, which are interpreted as sets of objects. Roles are interpreted as binary relations between objects. Each DL defines also a number of language constructs (such as intersection, union, role

quantification, etc.) that can be used to define new concepts and roles. DL support inference patterns which occur in many applications of intelligent information processing systems.

The techniques for reasoning in DL, refers to four different settings: 1) reasoning with plain concept expressions 2) reasoning with instances of concepts 3) reasoning with axioms expressing properties of concepts 4) reasoning with both instances of concepts and axioms. One of the main reasoning services of a DL system is to automatically build the subsumption hierarchy of classes, i.e., a graph showing all the subsumption relations between the classes of an application. The classification hierarchy of Ontology expressed as per DL provides useful information on the connection between different concepts, and it can be used to speed-up other inference services. The following section gives an insight into quite a few practically implemented DL systems and knowledge sharing systems.

KL-ONE, CLASSIC, BACK, LOOM, KRIS, CRACK are knowledge representation systems built on DL. KL-ONE (Brachman and Schmolze 1985) is one of the typical systems of earlier generation DL systems. KL-ONE inherently included the notion of inferring implicit knowledge from given declarations. It also supported generic concepts, which denote classes of individuals, and individual concepts to denote individuals. More important and unique to KL-ONE is the core idea of providing ways to specify concept definitions allowing a knowledge engineer to declare the relations between high-level concepts and lower-level primitives.

CLASSIC is a knowledge representation system (Alex et al 1989) based on DL designed for applications where only limited expressive power is necessary, but rapid responses to questions are essential. LOOM introduced the notion of Tbox and Abox for dealing with concept definitions and assertion regarding the concepts respectively (MacGregor 1991). The LOOM was developed for applications like natural language and image interpretation. A problem with the LOOM approach is that it is hard to characterize the source of incompleteness of reasoning algorithms, which might lead to unexpected behavior.

BACK is based on DL, implemented in PROLOG. The BACK architecture was designed to support incremental additions to the Abox and retraction of old information. Abox assertions can be retrieved from a database by automatically computing SQL queries. The BACK system was considered highly suitable for the applications where reasoning about time was important. FLEX (Quantz et al 1996), a successor of BACK included the facility for reasoning about equations and inequalities concerning integers. The CRACK system (Bresciani et al 1995) supported the DL language with all basic extensions in the constructor zone and the inference algorithms. KRIL (Aghila et al 2003) is based on extended DL which takes inspirations from Indian philosophy. KRIL basically provided all fundamental definitions and reasoning services of every other DL systems but the definition of knowledge base in KRIL followed the systems of Indian Logic.

#### **4.6 Extended Description Logics**

DL may be suitable for effective knowledge representation from a detailed knowledge base. However, DL systems failed to construct the atomic concept in terms of its member qualities. Also the relations defined possessed heavy limitations with respect to their scope and were only superficial, i.e. between atomic concepts. Extended DL system, besides satisfying the relation definitions at the conceptual level, also associated relations between concepts and its member qualities. Thus, by enhancing the number and type of relations that can be defined at the highest level of abstraction, inference becomes multi-relational rather than been restricted to the normal hierarchical, part-of and instantiation relations.

#### **4.8 The OAR Model**

The object-attribute relational model is one worth mentioning at this juncture. The nodes in OAR model are objects; links are relations. The objects are associated with attributes of their own. The novelty in OAR model (Wang 2006a) is that it matches the process of human cognition.

Indian philosophy serves as the strong foundation for the mathematical support behind the construction of knowledge base which is in the form of Indian logic based ontologies. In this context, Nyaya Sastra, the famous Indian school of Philosophy, has laid great stress on categorisation of world knowledge into a classification framework (Virupakshananda 1994). The following section elaborates on the motivation behind the selection of Indian philosophy for the new knowledge representation formalism discussed in this paper.

### **5 MOTIVATION**

For representing Ontology in a more meaningful manner, description logics (DL) have been used widely (Calvanese et al 2002). However, though the representation formalisms like DL, has the capability to deal with uncertain knowledge, they do not consider the presence of member qualities as an ‘inherent’ part of concept descriptions. In addition, relations defined possessed heavy limitations with respect to their scope and only existed between atomic concepts. Due to the added requirements of Ontology representation based on Nyaya, extensions were made both at the concept-constructor zone and the relation zone (Mahalakshmi et. al, 2002).

Modification of basic reasoning services of standard DL by Aghila et al (2003) created scope for definition of more descriptive concept elements based on Indian philosophy. However, extended DL Systems (Mahalakshmi et. al, 2007, Mahalakshmi and Geetha, 2008a) dealt with the basic reasoning services from philosophical perspective but the ontological classification was present only as part of reasoning mechanisms. Though concepts possessed ‘qualities’ defined by Nyaya Sastra, general relations existed between two concepts, and a special relation, ‘inherence’ existed (for applicable concepts) between concept and its member quality. i.e. the definition of ontological entities provided by extended DL does not tackle the special requirements dictated by Nyaya Sastra’s classification framework.

Other Nyaya recommendations like ‘invariable concomitance’ relation, ‘contact-contact’ relation and the like (Virupakshananda 1994) were not incorporated in extended DL Systems (Aghila et al 2003). This motivated us to design a special knowledge representation formalism which completely inherits all the classification recommendations of Nyaya Sastra.

### **6. NORM – NYAYA ONTOLOGY REFERENCE MODEL**

Nyaya Ontology Reference Model (NORM) defines the standards for constructing Ontology, based on the recommendations of the epistemology definitions of Nyaya-Vaisheshika schools of Indian philosophy. NORM (Mahalakshmi G.S. and Geetha T.V. 2008b), is organized as two-layer Ontology, where the upper layer represents the abstract fundamental knowledge and the lower layer represents the domain knowledge. According to NORM, a node in the Ontology is composed of an enriched concept which is related implicitly to its member qualities and explicitly to other peer concepts, by means of relations (Wada 1990).

A node of Nyaya-Vaisheshika based Ontology has the following structure (Figure 1). Every concept of the world knowledge shall be thoroughly classified as per NORM structure. The abstract and domain concepts form a strict classification hierarchy. The traits of the top-level concepts are applicable down

the hierarchy. Every concept in the NORM model has links to other concepts by external relations (Figure 1a). A concept is made of qualities. In addition, the qualities are bounded to the concept by internal relations. The qualities may also be related to each other, which are depicted as dotted edges (Figure 1b). Every quality has a set of values, and the values are bounded to the qualities by grouping relations (Figure 1c). This model is inspired by the various recommendations of classifications of world knowledge according to Nyaya-Vaisheshika.

According to NORM, a concept is an abstract entity which embeds several qualifying attributes of its own. The attributes are bound to the concept by relation existence. An attribute is a sub-property of a concept / relation which is used to define the concept set. Attributes are optionally supported by values. Quality may be ‘mandatory’ or ‘optional’ type. The ‘mandatory’ attribute of a concept shall be made ‘exclusive’ or ‘exceptional’ but not ‘optional’. The ‘optional’ attribute of a concept shall be made ‘mandatory’ which contributes more to inferential learning; but the vice versa is not allowed and is considered as violation of Ontology definitions. The qualifying attributes of a concept are not present as only the feature set of that concept; instead, every qualifying attribute is related to or bound to the concept by relations. Relations may be invariable, exclusive, exceptional etc. The following section discusses the system of classification recommended by Nyaya Sastra.

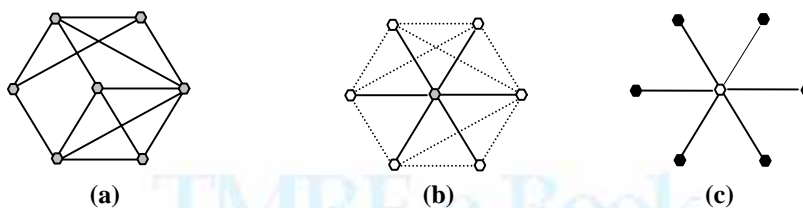


Figure 1 NORM Model for Cognitive Knowledge representation

(a) Ontology with concepts as nodes and external relations as edges, (b) a concept with qualities as nodes, internal relations as thin edges, tangential relations as dotted edges, (c) a quality with values as nodes, grouping relations as edges (Mahalakshmi G.S. and Geetha T.V. (2008b).)

## 7 Nyaya System of Classification

According to Nyāya Sastra (Sinha and Vidyabhusana 1930), every concept is classified into seven categories: substance, quality, action, generality, particularity, inherence and negation. Among these, the substance is of nine kinds: earth, water, light, air, ether, time, space, soul and mind. Every substance is threefold: body, organ and object. The object of light is fourfold: earthly, heavenly, gastric and mineral. Every substance is said to possess some quality. The quality is of twenty-four varieties which in turn possess values (Figure 2).

The permissible action associated with the substance is of five types: upward motion, downward motion, contraction, expansion, and motion. Generality is either more comprehensive or less comprehensive. Negation is of four varieties: antecedent negation (or prior negation), destructive negation (or posterior negation), absolute negation and mutual negation. Out of the nine substances, odour persists only in earth and is inherent. Earth exists in all the seven colors. Air has no color; water is pale-white in color and light is bright-white in color. Air has touch. Water has cold-touch and light has hot-touch. Dimension (or magnitude), distinctness, conjunction and disjunction are present in all the nine substances. Remoteness and Proximity is found in earth, water, light, air and mind. Heaviness or Weight is only in earth and water. Viscidity is present only in the substance, Water.

**Color:** white, blue, yellow, red, green, brown, varied  
**Taste:** sweet, acid, saline, pungent, astringent, bitter  
**Odour:** fragrant, foul  
**Touch:** cool, hot, lukewarm  
**Number**  
**Magnitude:** atomic, large, long, short  
**Separateness**  
**Conjunction**  
**Disjunction**  
**Remoteness:** spatial, temporal  
**Proximity:** spatial, temporal  
**Weight**  
**Fluidity:** natural, artificial  
**Viscosity**  
**Sound:** articulate, inarticulate  
**Intellect**  
**Pleasure:** *Remembrance, apprehension: Erroneous apprehension, valid apprehension:*  
                   Valid apprehension : perception, inference, analytical knowledge and verbal testimony  
**Pain**  
**Desire**  
**Aversion**  
**Volition**  
**Merit**  
**Demerit**  
**Tendency**

Figure 2 Ontological Classification of Nyāya-Vaiśeṣika Qualities

## 8 NYAYA DESCRIPTION LOGIC (NDL)

The knowledge representation language of ‘Gautama’ is christened as ‘Nyaya Description Logics’ (NDL). The NDL commands are classified into concept/relation definition language (CRDL), concept/relation manipulation language (CRML) and a set of editing commands and a query language (Mahalakshmi and Geetha 2008a). This knowledge representation language can be further used to define, manipulate and query the various levels of knowledge. CN refers to Concept name, QN refers to Quality Name, V – Quality value (Ex: color – Indigo: quality: color, value: Indigo) RN refers to Role name, I refer to Instance and Rdesc refers to Role descriptions. Following the above norms of definition of knowledge representation languages (as description logic commands), here, we define the sample Nyaya logic commands which are listed in Table A1.1.

The CRDL and CRML commands are seldom needed for creating the Ontology since, ‘Gautama’, the Ontology editor provides enough facilities for creation and updation services. The query language shall be used with the RDF generated by Gautama, to query about various parts of the Ontology. Here, we discuss few commands of the query services.

- Concept-satisfiable – This takes a concept name as the parameter and checks whether the addition of the concept will not violate the Ontology definitions that exist prior to the execution of this command
- Concept-subsumes – This takes two concepts as input, and checks whether the first concept subsumes the second concept. This is one of the famous reasoning service provided by any Ontology-based reasoner.
- Concept ancestors and Concept-descendants – These commands list the ancestral / descending concepts in the Ontology hierarchy. Role-ancestors and Role-descendants also have similar purpose.
- Sub-concept, Super-concept – These commands retrieve the child nodes or parent nodes of the parametric concept from the Ontology hierarchy
- Chk-concept-related – This command has three variations. It either checks whether a concept is related to another concept, through a particular relation name or through a particular set of relation categories.
- Chk-quality – This command checks the entire Ontology hierarchy to check if the required quality is available in the Ontology
- Chk-concept-quality – This command checks the entire Ontology hierarchy to check if the particular concept has the required quality.
- All-concepts, all-qualities, all-roles, all-instances – These commands just lists all the concepts, qualities, roles or instances available in the Ontology.
- Retrieve-direct-concepts, retrieve-indirect-concepts – The first commands take an instance as input, and retrieve all the directly related concepts to those instances; The second command take the instance as input and retrieves all the second and higher degree concepts related to those instances. For example, if ‘TooToo’ is an instance of penguin, the first command may retrieve ‘penguin’ as the result; the second command will retrieve all the ancestors of penguin which are conceptually related to penguin. Retrieve-direct-instances, retrieve-indirect-instances also serve the same purpose.

The following are the various tags defined for the RDF of Indian logic Ontology generated according to NORM.

- <rdf:concept> - This tag is used to declare a concept prior and after its definition
- <rdf:name> - This tag is used to declare the name of a concept / quality / relation.
- <rdf:desc> - This tag is used to create descriptions or definitions for a particular concept
- <rdf:axiom> - This tag is used to create concept axioms
- <rdf:quality> - This tag is used to create member qualities for a given concept
- <rdf:type> - This tag is used to declare the type of a concept / quality / relation
- <rdf:role> - This tag is used to declare the role of a concept / quality
- <rdf:category> - This tag is used to declare the category of relation like external, internal, tangential or grouping.
- <rdf:operator> - This tag is used to declare the logical operators like and, or while creating the concept axioms of the Ontology

‘Gautama’ is a tool (Mahalakshmi and Geetha, 2009a) which facilitates the building of Ontology according to Nyaya Sastra’s classification scheme. It should equally be possible to build the Ontology

from the scratch, without using the tool and just by opening a notepad and writing the concepts along with the defined RDF tags for Indian logic ontologies.

The sample RDF generated for a simple Ontology for 'birds' domain is given in Figure 3. The facilities for interacting with the knowledgebase are done through knowledge representation languages. These languages form part of the reasoning mechanisms of RDF schema.

```

<rdf:concept>
  <rdf:name>bird</rdf:name>
</rdf:concept>

<rdf:concept>
  <rdf:name>pigeon</rdf:name>
  <rdf:axiom>bird</rdf:axiom>
  <rdf:desc>
    <rdf:quality>
      <rdf:name>action</rdf:name>
      <rdf:type>exclusive</rdf:type>
      <rdf:operator>and</rdf:operator>
      <rdf:value>
        <rdf:name>walk</rdf:name>
        <rdf:operator>and</rdf:operator>
      </rdf:value>
      <rdf:value>
        <rdf:name>fly</rdf:name>
        <rdf:operator>inclusive</rdf:operator>
        <rdf:role>
          <rdf:name>isa</rdf:name>
          <rdf:category>VVrelationship</rdf:category>
          <rdf:type>symmetric</rdf:type>
          <rdf:relToValue>walk</rdf:relToValue>
          <rdf:operator>and</rdf:operator>
        </rdf:role>
      </rdf:value>
      <rdf:role>
        <rdf:name>hasA</rdf:name>
        <rdf:category>QVrelationship</rdf:category>
        <rdf:type>transitive</rdf:type>
        <rdf:relToValue>walk</rdf:relToValue>
        <rdf:operator>or</rdf:operator>
      </rdf:role>
      <rdf:role>
        <rdf:name>hasA</rdf:name>
        <rdf:category>QVrelationship</rdf:category>
        <rdf:type>transitive</rdf:type>
        <rdf:relToValue>fly</rdf:relToValue>
        <rdf:operator>and</rdf:operator>
      </rdf:role>
      <rdf:role>
        <rdf:name>isA</rdf:name>
        <rdf:category>QCrelationship</rdf:category>
        <rdf:type>transitive</rdf:type>
        <rdf:relToConcept>pigeon</rdf:relToConcept>

```

```

        <rdf:operator>and</rdf:operator>
    </rdf:role>
</rdf:quality>
<rdf:quality>
    <rdf:name>consumes</rdf:name>
    <rdf:operator>and</rdf:operator>
    <rdf:type>exclusive</rdf:type>
    <rdf:role>
        <rdf:name>isA</rdf:name>
        <rdf:category>QQrelationship</rdf:category>
        <rdf:type>transitive</rdf:type>
        <rdf:relToQuality>action</rdf:relToQuality>
        <rdf:operator>and</rdf:operator>
    </rdf:role>
    <rdf:role>
        <rdf:name>isA</rdf:name>
        <rdf:category>QCrelationship</rdf:category>
        <rdf:type>transitive</rdf:type>
        <rdf:relToConcept>pigeon</rdf:relToConcept>
        <rdf:operator>and</rdf:operator>
    </rdf:role>
</rdf:quality>
<rdf:role>
    <rdf:name>isA</rdf:name>
    <rdf:category>CCrelationship</rdf:category>
    <rdf:type>transitive</rdf:type>
    <rdf:relToConcept>bird</rdf:relToConcept>
    <rdf:operator>and</rdf:operator>
</rdf:role>
</rdf:desc>
</rdf:concept>

```

Figure 3 NORM RDF to describe the concept ‘pigeon’

Table 1 NDL commands for querying with Gautama

CRDL	CRML
define-concept<CN, Level>	insert-quality<QN>
define-concept-axiom<CN, Cdesc>	delete-quality<QN>
disjoint-concept<C1, C2>	insert-values<QN, V1... Vn>
define-role-axiom<RN, Rdesc>	delete-values<QN, V1... Vn>
disjoint-role<R1, R2>	delete-concept<CN>
define-concept-role<RN, C1, C2>	delete-instance<I>
define-concept-qualities<CN, (QM, Qman.List) / (QO, Qopt.List) / (QE, Qexceptional.List) / (QX, Qexclusive.List)>	update-instance<I, Cnold, Cnnew>
define-quality-values<CN, QN, V1... Vn>	delete-role-filler<I1, I2, RN>
define-role-quality <RN, CN, Qreflexive.List / Qsymmetric.List / Qassymmetric.List / Qantisymmetric.List / Qtransitive.List / Qdirect.List / Qindirect.List / Qexclusive.List>	update-role-filler<I1, I2, Rnold, Rnnew>
	delete-role<RN>
	insert-role<RN>
	delete-concept-quality<CN, QN>
	delete-quality-
	value<CN, QN, VInvariableConcomittance.List / VExclusive.List / VInvariableConcomittance.List

define-quality-role< <i>RNreflexive.List</i> / <i>RNasymmetric.List</i> / <i>RNsymmetric.List</i> / <i>RNantisymmetric.List</i> / <i>RNtransitive.List</i> / <i>RNdirect.List</i> / <i>RNindirect.List</i> / <i>RNexclusive.List</i> , <i>CN</i> , <i>QN</i> >	/ <i>VDirect.List</i> > insert-quality-value< <i>CN</i> , <i>QN</i> , <i>VInvariableConcomittance</i> . <i>List</i> / <i>VExclusive.List</i> / <i>VInvariableConcomittance.List</i> / <i>VDirect.List</i> > update-quality-value< <i>CN</i> , <i>QN</i> , <i>Vold</i> , <i>Vnew</i> >
--	---

Table 1 (Continued)

Query language	Query language
concept-satisfiable< <i>CN</i> >	retrieve-direct-concepts< <i>I</i> >
concept-subsumes< <i>C1</i> , <i>C2</i> >	retrieve-indirect-concepts< <i>I</i> >
concept-disjoint< <i>C1</i> , <i>C2</i> >	retrieve-concept-fillers< <i>RN</i> , <i>C1</i> >
chk-concept< <i>CN</i> >	all-concepts< <i>I</i> >
concept-atomic< <i>CN</i> >	retrieve-qualities< <i>CN</i> >
concept-ancestors< <i>CN</i> >	retrieve-quality-value< <i>CN</i> , <i>QInvariableConcomittance</i> / <i>QExclusive</i> / <i>QExceptional</i> >
concept-descendants< <i>CN</i> >	retrieve-quality-value< <i>CN</i> , <i>QDirect</i> >
super-concept< <i>CN</i> >	chk-instance< <i>I</i> >
sub-concept< <i>CN</i> >	chk-instance-type< <i>I</i> , <i>CN</i> >
chk-concept-related< <i>C1</i> , <i>C2</i> >	chk-instance-related< <i>I1</i> , <i>I2</i> >
chk-concept-related< <i>C1</i> , <i>C2</i> , <i>RN</i> >	retrieve-direct-instances< <i>I</i> >
chk-concept-related< <i>C1</i> , <i>C2</i> , <i>RNreflexive.List</i> / <i>RNasymmetric.List</i> / <i>RNsymmetric.List</i> / <i>RNantisymmetric.List</i> / <i>RNtransitive.List</i> / <i>RNdirect.List</i> , <i>RNindirect.List</i> / <i>RNexclusive.List</i> >	retrieve-indirect-instances< <i>I</i> >
chk-quality< <i>QN</i> >	retrieve-instance-fillers< <i>RN</i> , <i>I1</i> >
chk-concept-quality< <i>CN</i> , <i>QN</i> >	all-instances< <i>CN</i> >
all-qualities	retrieve-related-instances< <i>RN</i> >
	retrieve-quality-value< <i>I</i> , <i>QN</i> >
	chk-role< <i>RN</i> >
	all-roles
	role-descendants< <i>CN</i> >
	role-ancestors< <i>CN</i> >

The Ontology editor based on Indian logic facilitates the user to carefully handcraft the Ontology based on Nyāya-Vaiśeṣika system of classification of concepts in the required domain. However, there are other noteworthy projects existing in the knowledge representation arena. Cyc, WordNet, Concept-Net and Mind-Net are to name a few (Matuszek et al 2005; Fellbaum 1998; Vanderwende et al 2005; Concept Net 2008). In future, adapting more ideas of building the Ontology from Indian philosophy would strengthen the outcome of the Ontology editor.

Thus, with a detailed manner of representing world knowledge, we could arrive at better inferences during common-sense reasoning, which will serve as a major leap forward in knowledge representation and interpretation services.

## 9 CASE STUDY

This section shortly discusses the case study across three knowledge representation formalisms namely, description logics, extended description logics and nyaya description logics.

Let us consider the example: *Cat is a mammal*

In DL, there will be two concept nodes *cat* and *mammal* generated and a relation *is-a* links between two concept nodes. In X-DL (extended DL) there will be same two concept nodes created and relation applied, but the new nodes will have more details added to it, namely, *cat is a living thing*. The reason is the class *mammal* has quality *soul* associated to it (Virupakshananda 1994), thus defining *cat* to be a *living thing*. At this juncture, Nyaya description logics also does the same. This is the same for *Penguin is a bird*. When a new statement like, *Will penguin fly?* comes, DL reasoners fail to say the correct answer. In X-DL, the attributes of *penguin* will be elaborated as ‘*one with boneless wings*’ in the axiom of *penguin*, therefore, X-DL and NDL would be able to justify that *penguin will not fly*.

However, for *mountain is smoky*, DL reasoners tend to do a little over the inferences. X-DL reasoners analyses the attributes of both the concepts *mountain* and *smoke* and tries to relate both the concepts by other intelligent means of relations like *smoke is over the mountain* and *mountain is not made of smoke*. But only NDL reasoners tend to analyse the reason of *smoke over mountain* and give out their inferences as *mountain has fire*. They associate the invariable concomitance relation (Virupakshananda 1994) between *smoke* and *fire*. Invariable relation says that ‘wherever x is present y is also present’. Thus wherever smoke exists, fire also exists. NDL reasoners mimick the way of human inferences (Mahalakshmi and Geetha 2009b) by finding the cause of the effect smoke over the mountain.

The reason is obvious. As said earlier, NDL does not only vary in defining the concept axioms but rather the classification of world entities take inspirations from Nyaya Ontological classification, and therefore, a concept or an instance of a concept is properly tagged and classified under its definitional hierarchy.

At this juncture, a small comparison with OAR model (Wang 2006b) would be wonderful. OAR model, also called as Object-Attribute-Relational model defines the concept axioms though in the style of human cognition, but not adequately enriched in relation with other existing information in the Ontology. For example, *cat is a mammal* would require only two object nodes *cat* and *mammal* with a link *is-a* between them, as in the case of DL reasoners. However, the improvement in OAR model is that the properties of object nodes are also taken into consideration while defining the object nodes. In NDL, we refer to the object nodes of OAR model as to concept nodes and instance nodes, and therefore, the definition of *cat* as a concept axiom would be more detailed as to defining its properties, relations between its properties, permissible values of the properties etc.

In other words, *an yellow cat* will be defined as a *cat* object with *yellow* as the object property in OAR model; in NDL it is like defining a *cat* concept and *yellow* as quality: color, to be inserted into the knowledge base. However, the intelligence lies in the inference algorithm that, by commonsense definitions into NDL, a cat can never be yellow in color, and therefore, the *yellow* color may be due to some artificial coloring and therefore, this intelligence helps in eliminating inconsistencies from the knowledgebase.

## 10 CONCLUSION

This paper discussed the Nyaya Description Logics which is the most effective method to represent knowledge useful for inference and reasoning purposes. The methodology is more effective because it tackles inferences similar to the approach of human cognition. This paper also analysed the issues in existing knowledge representation formalisms. More mathematical analysis and detailed comparison of knowledge representation formalisms to promote NDL becomes our future work.

**REFERENCES**

- Aghila G., Mahalakshmi G.S. and Geetha T.V. (2003), 'KRIL - A Knowledge representation System based on Nyaya Shastra using Extended Description Logics', *VIVEK Journal*, ISSN 0970-1618, Vol. 15, No. 3, pp. 3-18.
- Alex B., Brachman R.J., McGuinness D.L. and Resnick L.A. (1989), 'CLASSIC: A structural data model for objects', *Proceedings of 1989 ACM SIGMOD International Conference on Management of Data*, pp. 59-67.
- Baader F., Calvanese D., McGuinness D., Nardi D. and Schneider P.F. (2002), 'The Description Logic Handbook: Theory, Implementation and Applications', Cambridge University Press.
- Brachman, R.J. (1979) On the epistemological status of semantic networks, in Findler, N., (Ed.) *Associative networks: representation and use of knowledge by computers*. New York: Academic Press
- Brachman R.J. and Schmolze J. (1985), 'An Overview of the KL-ONE Knowledge representation System', *Cognitive Science*, Vol. 9, No. 2.
- Bresciani E., Franconi and Tessaris S. (1995), 'Implementing and testing expressive DL, Description logics: a preliminary report', In Gerard Ellis, Robert A. Levinson, Andrew Fall and Veronica Dahl (eds.), *Knowledge Retrieval, Use and Storage*.
- Calvanese D., Giacomo G.D. and Lenzerini M. (2002), 'Description Logics: Foundations for Class-based Knowledge representation', Roma, Italy.
- Woods, W.A. (1975). What's in a link: Foundations for semantic networks. Bobrow, D.G. and Collins, A.M., Ed. *Representation and Understanding: Studies in Cognitive Science*. pp.35-82. New York, Academic Press.
- Conceptual Graph Standard, 2002 NCITS.T2 Committee on Information Interchange and Interpretation. <http://users.bestweb.net/~sowa/cg/cgstand.htm>.
- Concept Net (2008), <http://www.conceptnet.org>.
- Davis, R., Shrobe, H., Szolovits, P., (1993). What is a knowledge representation. *AI Magazine* 14 (1), 17-33.
- Davies J., Duke A. and Sure Y. (2004), 'OntoShare - An Ontology-based Knowledge Sharing System for virtual Communities of Practice', *Journal of Universal Computer Science*, Vol. 10, No. 3, pp. 262-283.
- Fellbaum C. (1998), 'WordNet - An Electronic Lexical Database', with a preface by George Miller.
- Flouris G., Plexousakis D. and Antoniou G. (2006), 'Evolving Ontology Evolution', *SOFSEM 2006: Theory and Practice of Computer Science*, LNCS Vol. 3831.
- Lau T., Wolfman S., Domingos P. and Weld D. (2003), 'Programming by Demonstration using Version Space Algebra', *Machine Learning*, Vol. 53, No. 1-2, pp. 111-156.
- MacGregor R. (1991), 'Inside the LOOM description classifier', *SIGART Bulletin*, Vol. 2, No. 3, pp. 8-92.
- R. MacGregor and R. Bates (1987). *The Loom Knowledge representation Language*. Technical Report, USC/Information Sciences Institute.
- Mahalakshmi G.S., Aghila G. and Geetha T.V. (2002), 'Multi-level Ontology representation based on Indian Logic System', 4th International Conference on South Asian Languages ICOSAL-4, India, pp. 4-15.

- Mahalakshmi G.S., Anupama N., Chitra R. and Geetha T.V. (2007), *Deepika – A Non-Monotonic Reasoning System Based On Indian Logic*, International Conference on Soft Computing Techniques in Engineering, SOFTECH-07, Avinashilingam Univ. for Women, India, pp. 470-476.
- Mahalakshmi G.S. and Geetha T.V. (2008a), ‘Gurukulam-Reasoning based Learning System using Extended Description Logics’, Dr. Estrella Pulido and Dr. Maria D. R-Moreno (eds.), International Journal of Computer Science and Applications (IJCSA) - Special Issue on New Trends on AI techniques for Educational Technologies, Technomathematics Research Foundation, Vol. 5, No. 2, pp. 14-32.
- Mahalakshmi G.S. and Geetha T.V. (2008b), ‘Reasoning and Evolution of consistent ontologies using NORM’, International Journal of Artificial Intelligence : Special Issue on Theory and Applications of Soft Computing, Indian Society for Development and Environment Research, Vol. 2, No. S09, pp. 77-94.
- Mahalakshmi G.S. and Geetha T.V. (2009a), ‘Gautama – Ontology editor based on Nyaya Logic’, Proc. of Third Indian Conference on Logic and Applications, Ramanujam R. and Sundar Sarukkai (eds.), Springer LNAI 5378, FoLLI series, pp. 234-245.
- Mahalakshmi G.S. and Geetha T.V. (2009b), ‘Argument Based Learning Communities’, KB Systems, Spl. issue on AI in Blended Learning, Elsevier, Vol. 22, No. 4, pp. 316-323.
- Matuszek C., Witbrock M., Kahlert R.C., Cabral J., Schneider D., Shah P. and Lenat D. (2005), ‘Searching for Common Sense: Populating Cyc from the Web’, In: Proceedings of 20th National Conference on Artificial Intelligence, Pittsburgh, Pennsylvania.
- Minsky, M., (1975) A framework for representing knowledge. In: Winston, P. (Ed.), *The Psychology of Computer Vision*. McGraw-Hill, New York, pp. 211–277 URL: <http://publications.ai.mit.edu/ai-publications/pdf/AIM-306.pdf>.
- C. Peltason, A. Schmiedel, C. Kindermann and J. Quantz. (1989) *The BACK System Revisited*. Technical Report kit - report 75, Projektgruppe KIT- Fachbereich Informatik-TU Berlin.
- Quantz J., Dunker G., Bergmann F. and Kener I. (1996), ‘The FLEX system’, Technica report, KIT-Report, Technische Universitat, Ber in, Germany, 175 <http://citeseer.ist.psu.edu/quantz95flex.html>.
- Quillian, M.R. (1968). *Semantic memory*. Minsky, M., Ed. *Semantic Information Processing*. pp.216-270. Cambridge, Massachusetts, MIT Press.
- Sinha N.L. and Vidyabhusana S.C. (1930), ‘The Nyāya Sutras of Gautama’, Translated by S.C. Vidyabhusana, edited by Nanda Lal Sinha, Sacred Book of Hindus, Allahabad, 1930, Reprinted in 1990, Delhi: Motilal Banarsidass
- Thomaz A.L., Hoffman G. and Breazeal C. (2006), ‘Reinforcement Learning with Human Teachers: Understanding How People Want to Teach Robots’, Proceedings of 15th IEEE International Symposium on Robot and Human Interactive Communication, pp. 352-357.
- Toni C. and Daniel T. (2006), ‘Learnable behavioural model for autonomous virtual agents: low-level learning’, In Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems, pp. 89-96.
- Vanderwende L., Kacmarcik G., Suzuki H. and Menezes A. (2005), ‘MindNet: An Automatically-Created Lexical Resource’, In Proceedings of HLT/EMNLP 2005 Interactive Demonstrations, Canada.
- M. Vilain (1985). The restricted language architecture of a hybrid representation system. In Proceedings of IJCAI-85, Los Angeles, Ca., pp. 547-551. IJCAI.
- Virupakshananda Swami (1994), ‘Tarka Samgraha’, Sri Ramakrishna Math, Madras.
- Wada T. (1990), ‘Invariable Concomitance in Navya-Nyaya’, Sri Garib Dass Oriental Series No. 101, Indological and Oriental Publishers, India.

- Wang Y. (2006a), 'On Concept Algebra and Knowledge representation', Proceedings of 5th IEEE International Conference on Cognitive Informatics (ICCI'06), Yao Y.Y., Shi Z.Z., Wang Y. and Kinsner W. (eds.).
- Wang Y. (2006b) 'The OAR model for Knowledge representation', Proceedings of Canadian Conference on Electrical and Computer Engineering, pp. 1727-1730.

TMRF e-Book