A probabilistic Framework for Information Integration and Retrieval on the Semantic Web

Livia Predoiu
Institute of Computer Science,
University of Mannheim,
A5,6, 68159 Mannheim, Germany
livia@informatik.uni-mannheim.de

Heiner Stuckenschmidt
Institute of Computer Science,
University of Mannheim,
A5,6, 68159 Mannheim, Germany
heiner@informatik.uni-mannheim.de

ABSTRACT

We present a framework for probabilistic Information Processing on the Semantic Web that is capable of representing ontologies, deductive databases, uncertain mappings between them, results of statistical instance classification and ontology learning. Our framework is built on a knowledge representation formalism called *Bayesian Description Logic Programs* because it is a probabilistic extension of Description Logic Programms and a fragment of Bayesian Logic Programs. We show in this paper how to perform information integration and retrieval within our framework by means of integrated reasoning.

1. INTRODUCTION

So far, the languages for the Semantic Web were dominated by traditional views on metadata models and logic-based knowledge representation. The major languages currently prevalent on the Semantic Web are RDF [1, 2] for representing metadata and the Web Ontology language OWL [3] for representing terminological knowledge in terms of ontologies basing on the Description Logics paradigm. Recently, the need for rule languages on the Semantic Web has been recognized. Rule languages enable e.g. property chaining which is not possible in the Description Logics on which OWL-Lite and OWL-DL are basing on. Thus, several rule language proposals for the Semantic Web have emerged, e.g. the Semantic Web Rule Language SWRL [4] and the Web Rule Language WRL [5]. However, all of these are logical languages with a model-theoretic semantics that makes a statement either true or false.

There are many arguments, why approaches that solely rely on traditional logics are not adequate. In an open environment such as the Semantic Web, we are frequently faced with uncertain, unreliable or contradicting information. Using traditional logics for representing this information will often lead to inconsistencies or wrong conclusions. We believe that probabilistic extensions of Semantic Web languages are needed to better address the challenges of web-based knowledge representation and to adequately model the uncertainty inherently present in web-based information. In particular, we see three sources of uncertainty on the Semantic Web that require probabilistic methods if we want to take advantage of these sources:

Document Classification:. Today, a major part of the information on the web is stored in documents (Web Pages, PDF Documents, etc.). A common way of linking documents to knowledge encoded in ontologies is to assign each document to one or more concepts representing its content. Different machine learning techniques have been applied to this problem [6]. Examples are naïve Bayes classifiers that estimate the probability of a document belonging to a topic based on the occurrence of terms in sample documents.

Ontology Learning:. The manual creation of ontologies has been identified as one of the main bottlenecks on the Semantic Seb. In order to overcome this problem, methods for learning ontologies from texts are investigated. Existing approaches normally use a combination of NLP and text mining techniques [7]. The result of the mining process can be interpreted in terms of a probabilistic judgement of the correctness of the learned relation.

Ontology Matching:. Normally, different sources use different ontologies to organize their information. In order to be able to access information across these different sources, semantic correspondences between the ontological entities in different ontologies have to be determined and encoded in mappings. A number of approaches for automatically determining such mappings have been proposed [8]. Normally, numbers are computed representing the similarity between ontological entities. These numbers represent the confidence that a matching holds and can be interpreted and used probabilistically.

1.1 An Example Scenario

We illustrate the need for probabilistic approaches by considering a retrieval scenario where different peers provide information about publications based on different bibliographic ontologies. Our example is based on the ontologies used in the ontology alignment evaluation challenge[1]. Assume a situation where a user is looking for publications about AI based on one local ontology O_1 which is specified

VLDB '07, September 23-28, 2007, Vienna, Austria.

[1] The complete ontologies can be found at http://oaei.ontologymatching.org/

by the following axiom:

$$Publication \sqsubseteq \forall keyword.Topic \sqcap \quad (1)$$
$$\forall author.Person \quad (2)$$

A query for publications about AI takes the following form:

$$Publication(x) \wedge keyword(x, AI). \quad (3)$$

There are techniques for answering such conjunctive queries over local description logic ontologies (e.g. [9]). The situation becomes more complicated, if we also want to include results from other sources. First of all, these sources may use a different ontology for describing publications. Publications could for example be described in another external ontology O_2 in the following way:

$$Publication \sqsubseteq Resource \quad (4)$$
$$Report \sqsubseteq Publication \quad (5)$$
$$\top \sqsubseteq \forall author^{-}.Resource \quad (6)$$
$$\top \sqsubseteq \forall author.Person \quad (7)$$

The first complication is that the second ontology O_2 talks about publications and authors but not about topics covered in the publications. This means that we might have information about certain publications like the two ones below, but no evidence whether they are relevant or not.

$$BDLP : Report, \quad (8)$$
$$BN : Report, \quad (9)$$
$$(BN, Judea) : author \quad (10)$$
$$(BDLP, Livia) : author, \quad (11)$$
$$(BDLP, Heiner) : author \quad (12)$$

We can overcome this problem by using machine learning techniques for classifying documents according to certain topics such as AI, logic and probability and to learn the hierarchical relations between these topics.

Another problem is the heterogeneity of the representations used for the different ontologies that require ontology matching techniques for determining which concepts, relations and instances of the different ontologies correspond to each other in each way. In our example scenario, *Reports* in O_2 might have to be regarded as *Publications* in O_1 as well[2], and there might have to be defined the relation between being about a certain topic and having a certain keyword.

1.2 Outline and Contribution

In this paper, we propose a framework for probabilistic information integration and retrieval based on an extension of existing Semantic Web languages with the ability to handle probabilistic information arising from the sources mentioned above. In order to be compatible with description logic-based languages such as OWL as well as rule-based approaches for knowledge modeling and deductive databases that naturally base on the logic programming as well, we take *Description Logic Programs* [10] (DLPs) as a starting point and extend them with probabilistic information. We call the resulting language *Bayesian Description Logic Programs* (BDLPs) because it is a subset of Bayesian Logic Programs as introduced in [11]. This enables us to build on known inference methods for Bayesian Logic Programs when reasoning with probabilistic information on the web.

This paper provides a substantial extension of [12] where the idea of Bayesian Description Logic Programs has been firstly proposed as a language that enables to integrate the logic programming and the description logics paradigm probabilistically. In this paper, we concretize the idea of integrating logic programming and description logics knowledge bases by means of BDLPs further and discuss integrated reasoning in more detail. Furthermore, we show how to perform probabilistic information retrieval within our framework. For this purpose we extend the BDLPs semantics to handle nonground queries. We exemplify how different kinds of tasks involving uncertainty when being performed automatically like document classification, ontology learning and ontology matching can be integrated within our framework with ontologies and logic programs or deductive databases that have been made accessable by peers in the Semantic Web. As discussed in section 5, such a framework has not been proposed yet.

The rest of the paper is organized as follows. We briefly introduce DLPs in section 2. In section 3, we describe Bayesian Description Logic Programs as a basis for our formalism. In section 4, we explain how this fomalism can be used for information retrieval and integration across heterogeneous ontologies. In section 5 we discuss related work. Finally, we present our conclusions in section 6.

2. REPRESENTING ONTOLOGIES AND QUERIES: DESCRIPTION LOGIC PROGRAMS

Description Logic Programs (DLPs) [10] lie in the expressive intersection of the knowledge representation formalisms of Description Logics (DL) [13] and of Logic Programming (LP) [14]. DLPs can be represented in a DL or an LP syntax. In this paper, we present the Logic Programming syntax of DLPs and look at them from the viewpoint of Logic Programming: The DLP language corresponds to a subset of Datalog without negation and without equality. Correspondingly, a Description Logic Program consists of a set of *rules* and a set of *facts*. Each rule has the form $H \leftarrow B_1 \wedge \ldots \wedge B_m$, where H and the B_i are atomic formulae and $m \geq 0$. An atomic formula consists of a predicate symbol p followed by a bracketed n-tuple of terms t_i, $p(t_1, \ldots, t_n)$ with $n \geq i \geq 0$. A term can be either a constant (i.e. an instance) or a variable (i.e. a placeholder for an instance). If all terms in an atomic formula are constants, the atomic formula is called a *ground atom.* The left hand side of a rule, H, is called *head* and the right-hand side of a rule, $B_1 \wedge \ldots \wedge B_m$, is called *body.* All variables in rules are universally quantified, although this is not explicitly written. For $i = 0$, the rule is called a *fact.* Only ground atoms are allowed in facts. In the DLP language, the predicates are only allowed to be 2-ary and the variable graph of the body of each rule is connected and acyclic.

In [10], it has been shown how to perform the so-called *DLP-fusion*, i.e. the bidirectional translation from the DL

[2] However, this is not evident, as it might be the case that publications only refer to officially published papers, but not to technical reports in the local ontology

syntax of the DLP language to the LP syntax. and vice versa. This means that DLPs provide a basis for achieving interoperability on the language-level. Although the DLP language has a restricted expressivity, it has several advantages. First, it has been shown that a large amount of existing Description Logic ontologies lie within the DLP fragment [15]. Furthermore, reasoning in the DLP language is decidable and has a much lower complexity than DL reasoning not only in theory but also in practice.

It turns out that the ontologies in our example scenario can be translated to DLP without loss of information. The definitions in the local ontology translate to the following rule set:

$$Topic(y) \leftarrow Publication(x) \wedge keyword(x, y) \quad (13)$$
$$Person(y) \leftarrow Publication(x) \wedge author(x, y) \quad (14)$$

In the same way, the ontology and the instances of the external source can be translated to DLP resulting into the following DLP representation:

$$Resource(x) \leftarrow Publication(x) \quad (15)$$
$$Publication(x) \leftarrow Report(x) \quad (16)$$
$$Resource(y) \leftarrow author(x, y) \quad (17)$$
$$Person(x) \leftarrow author(x, y) \quad (18)$$
$$Report(BDLP) \leftarrow \quad (19)$$
$$Report(BN) \leftarrow \quad (20)$$
$$author(BDLP, Livia) \leftarrow \quad (21)$$

$$author(BDLP, Heiner) \leftarrow \quad (22)$$
$$author(BN, Judea) \leftarrow \quad (23)$$

The uniform representation of local and external information in DLP now allows us to represent complex mappings between the different sources in terms of datalog rules thus building upon successful work in view-based schema integration [16].

3. REPRESENTING PROBABILISTIC INFORMATION: BAYESIAN DESCRIPTION LOGIC PROGRAMS

We identify Bayesian Description Logic Programs (BDLPs) [12] as the subset of Bayesian Logic Programs (BLPs) [11] that is a straightforward probabilistic extension of DLPs. We chose a subset of BLPs for representing probabilistic information because they allow a smooth representation of probabilistic information on top of logic programs and to use existing tools for probabilistic reasoning in Bayesian Networks as a basis for answering queries. DLPs are a knowledge representation formalism corresponding to an extension of Bayesian Networks [17] to Description Logic Programs.

3.1 The Language of BDLPs

The syntax of BDLPs is similar to DLPs. A Bayesian Description Logic Program consists of rules and facts that are built in the same way like with DLPs. However, one difference is that instead of the arrow ←, the symbol "|" is used in order to hint at the idea a of conditional probability densities. Another difference is that each rule has a probability distribution attached that specifies the probability for each state of the head atom given the states of the body atoms.

Another important difference between BDLPs and DLPs is that the atoms $p(t_1, \ldots, t_n)$ are *bayesian* which means that they are not necessarily boolean but can be associated to an arbitrary finite domain $D(p)$. To each predicate p_i of the BDLP, a unique domain $D(p_i)$ is associated. All ground atoms inherit the domain that belongs to their predicate. This means that a ground atom can have a state from the domain $D(p)$ that belongs to its predicate p. Hence, besides logical predicates whose domain consists of the truth values, BDLPs can also represent predicates with discrete or even continuous domains and assign probabilities to the different values in this domain. In the context of semantic web applications, non-logical predicates can for example be used to express uncertainty about datatype properties of a concept.

3.2 Semantics of BDLPs

A BDLP encodes a Bayesian Network (BN) in the sense that each ground atom in the model of the BDLP corresponds to a node in the network. Furthermore, if r is a rule in a BDLP:

$r : h(t_{h1}, \ldots, t_{hn})|b_1(t_{11}, \ldots, t_{1n}), \ldots, b_n(t_{n1}, \ldots, t_{nn})$.

and $D(h) = \{h_1, \ldots, h_m\}$ is the domain that the ground atoms containing the predicate h are associated to, then, for each combination of states the ground body atoms $(e_1, \ldots, e_n) \in D(b_1) \times \ldots \times D(b_n)$ can have and each state of the ground head atom h_i there is a function $cpd(r)(h_i|e_1, \ldots, e_n) : D(h) \mapsto [0, 1]$. This function is the conditional probability density of each of the random variables that are represented by the direct influence relationship between ground atoms encoded by such a rule. In terms of a Bayesian Network, each of these functions is translated to links connecting the node representing the possible instantiations of the head with the nodes representing the instantiations of the different atoms in the body. Note that rules with empty bodies are facts and for a fact f, the a-priori probability density is given in the same way.

A BDLP defines the structure of a Bayesian Network and provides a complete specification of the probability distribution the network encodes (The proofs for the correspondance of BLPs to Bayesian Networks which can be found in [11] hold for BDLPs as proper subset of BLPs as well). When a ground atom can be deduced by just one rule, the conditional probability is given by the distribution attached to the BLP rule. If, however, we have e.g. 2 rules with the same head predicate that can deduce the same ground head atom, then, the conditional probability density of the random variable that corresponds to the ground head atom needs to consider the possible states of all body atoms at once. For this purpose, so-called *combining rules* are used to generate a joint conditional distribution from the individual ones of the rules involved.

A combining rule is an algorithm that maps a finite set of conditional probability densities $\{p(h_i|a_{i1}, \ldots, a_{in_i})|m \geq i \geq 1, n_i \leq 0\}$, $m \geq 1$, to the conditional probability densities $p(h|b_1, \ldots, b_n)$ with $\{b_1, \ldots, b_n\} \subseteq \cup_{i=1}^{m}\{a_{i1}, \ldots, a_{in_i}\}$. As explained above, the combining rules are important to ensure that random variables that get e.g. by means of a reasoning process more direct parents, get also a valid conditional probability density. Combining rules can be different algorithms. [11] mentions as most simple combining rule e.g. the usage of the maximum of the former probability densities. More sophisticated ways of combining distributions are provided by variations of the noisy-or model. In this paper, we will use the previously mentioned combining rule *maximum* as it fullfils our purposes.

3.3 Querying a BDLP

The processing of queries posed to a BDLP B consists of two steps of a so-called *knowledge-based model construction* policy. If we consider the BDLP B without probability densities attached and substitute all occurences of the symbol $|$ by the symbol $\leftarrow$, we have a logic program B' which is called the *corresponding logic programm* of B. In the first step, the least Herbrand model H of B' is deduced by means of a Logic Programming Reasoner. In [11], a method is proposed that does not deduce the whole Herbrand Model, just the part of it that can be deduced by answering the query by means of SLD resolution. The Bayesian Network that corresponds to B can be created by means of the ground atoms in the Herbrand model. The part of the Bayesian Network which might be relevant to the query can be constructed by the ground atoms deduced by a top-down proof procedure for the query.

We briefly sketch how to construct the Bayesian Network or its part that is relevant to the query: For each fact f of B a node in the Bayesian Network is created and the probability density of f is attached to the node. For each rule r that has been used for properly deriving a (ground) fact in the head h, there are two possibilities:

- a node which corresponds to h does not exist in the BN:
  Then, such a node is created and for each body atom of the rule, an arc from the corresponding nodes in the BN[3] to the newly created node is created. The probability density of this rule in B is attached to this node.
- a node which corresponds to h exists in the BN:
  Then, for each body atom in the rule which has a corresponding node in the BN but no arc from this node to the node that corresponds to the head atom, such an arc is created. Afterwards the combining rule is applied to the probability density of r and the node corresponding to h. Thus, the node corresponding to h is equipped with a probability density which considers the probability densities of r and the rules that already have derived the same head atom h.

The resulting Bayesian Network BN can be used for querying for any probabilistic query. Analogously to probabilistic BLP queries, *a probabilistic BLDP query* is an expression of the form $?{-}Q_1, \ldots, Q_n | E_1 = e_1, \ldots, E_m = e_m$. This expression queries for the conditional probability $p(Q_1, \ldots, Q_n | E_1 = e_1, \ldots, E_m = e_m)$, i.e. the probability that the ground atoms Q_i are true given that we know some other ground facts E_j are each in the state e_j. BLP queries are only defined for the occuring query atoms, Q_i, and evidence atoms, E_i, beingt ground atoms that belong to the least Herbrand model H.

We extend the kinds of queries allowed in BLPs to allow also non-ground atoms as query atoms. Such queries ask for the probability of each valid grounding of the query. They are processed by first deducing all valid groundings of the query by means of a LP reasoner and afterwards asking for the probability of each grounding in the corresponding Bayesian Network. Queries with nonground query atoms can be used for Information Retrieval where the probabilities are used for ranking.

3.4 Our Example as BDLP

In our example scenario, we can represent uncertain information as follows. As mentioned in the introduction, there are two sources of uncertainty, the first is in the missing information about the topic of the papers in the external ontology. Using document classification techniques, we could for example determine the probability that the report is about logic and about probabilities with a certain degree of uncertainty. These facts can be represented as facts in a BDLP using the Boolean predicates aboutLogic and aboutProbability.

$$P(aboutLogic(BDLP) = true) \overset{0.8}{\leftarrow} \tag{24}$$

$$P(aboutProbability(BDLP) = true) \overset{0.9}{\leftarrow} \tag{25}$$

Further, the probability that a publication is about AI if it is about logic or probability, respectively, which we could determine using relational learning methods can be represented as rules of a BDLP. Below, we give the probability for the head being true given the states of the body atom in parantheses: the first number indicates the probability given the body is true and the second number indicates the probability given the body is false.

$$O_2 : aboutAI(x) \overset{(0.7,0.2)}{\leftarrow} O_2 : aboutLogic(x) \tag{26}$$

$$O_2 : aboutAI(x) \overset{(0.6,0.1)}{\leftarrow} O_2 : aboutProbability(x) \tag{27}$$

In a similar way, we can use BDLPs to represent mapping rules between the two ontologies in terms of probabilistic rules. The use of rules for representing mappings is quite a natural approach that has successfully been used in the database area. The use of probabilities for representing the degree of confidence in the correctness of the rules deserves a closer look. First of all, we can use probabilistic ontology matching tools such as Glue [18] or Caiman [19] that directly assign a probability to each mapping. In order to smoothly integrate the results of these tools into the system, we have to annotate the Boolean predicates that represent ontology elements with information about which of the models they belong to. In the example, we aim at querying O_1 and gather mappings from O_2 to O_1. The result could look as follows:

$$O_1 : Publication(x) \overset{(0.9,0.1)}{\leftarrow} O_2 : Publication(x) \tag{28}$$

$$O_1 : Publication(x) \overset{(0.75,0.2)}{\leftarrow} O_2 : Report(x) \tag{29}$$

Further, we can also assign subjective probabilities to manually created mappings. Then, the probability that a mapping is correct is judged by the person who created the mapping. Although it has been argued that people are often inconsistent in their judgement of probabilities, using subjective probabilities is still more accurate than forcing

[3] Each time a rule can derive a new head atom, the body atoms have been unified with ground atoms already identified as being in H and therefore for each of these ground body atoms, a node in the Bayesian Network already exists.

people to use a Boolean judgement. Thus we can also represent manually created mappings as rules in a Bayesian Logic program:

$$O_1 : keyword(x, AI) \overset{(0.9,0.01)}{\leftarrow} O_2 : aboutAI(x) \tag{30}$$

We assume that we also decide to add manually a rule $P(O_1 : author(x, y)\text{=true } | O_2 : author(x, y)\text{=true}) = 1.0$. In case, we do not want to think about an appropriate probability for this rule given the body atom is false, we can compute it by the formula $P(O_2 : author(x, y) = true) - P(O_1 : author(x, y) = true|O_2 : author(x, y) = true) = P(t : author(x, y) = true|O_2 : author(x, y) = false)$. Then, however, it has to hold that $O_1 : author(x, y)$ depends only on $O_2 : author(x, y)$ and we need to first asses the apriori probability $P(O_1 : author(x, y) = true)$ which can be done e.g. by counting within a sample. Let's assume that we yield in our example:

$$O_1 : author(x, y) \overset{(1.0,0.4)}{\leftarrow} O_2 : author(x, y) \tag{31}$$

4. PUTTING IT ALL TOGETHER: BAYESIAN DESCRIPTION LOGIC PROGRAMS

As we have argued above, BDLPs are a suitable knowledge representation framework for representing and reasoning about probabilistic information which is the result of automatic ontology mapping and document classification as it allows to reason across different ontologies and combine different sources of evidence in a sound formal framework. The step that is still missing is the integration with non-probabilistic information contained in the ontologies of the different sources.

In order to integrate deterministic ontological knowledge into the framework, we extend the corresponding DLP statements with probabilities basing on the assumption that everything which is true in the knowledge base is only true and not false as long as the knowledge base is consistent. Therefore, all facts in the knowledge base get a probability 1.0 for the state "true" and 0.0 for the state "false". For rules, it holds that if the body atoms are true, the rule will fire and the head atom becomes true as well. Thus, the head atom is true if all body atoms are true as well and it is false, if one of them is false. Note that this way to deal with rules is the same in all kinds of processing techniques typically used in the logic programming setting with languages without negation and without equality. E.g., the same holds here if we do not consider the forward-chaining processing technique but the backward-chaining one.

As Bayesian DLPs are a special case of BLPs, the reasoning algorithms devised for BLPs an be used to answer queries on BDLPs as well. I.e. by means of a Logic Programming reasoning algorithm, the least Herbrand Model can be computed and a Bayesian Network can be created by means of the applied rules and their conditional probability densities, the ground atoms of the least Herbrand Model and their a-priory probability density and the combining rules as described in section 3.3.

In figure 1, the Bayesian Network that corresponds to our example is shown. The dark blue arcs and nodes originate from O_2, the cyan coloured arcs and nodes have been learned additionally within O_2 and white nodes and black arcs originate from O_1. Red arcs represent mappings.

As first example for query answering, let's consider the query example mentioned in section 3.3. Within the Bayesian Network corresponding to our example setting this query would yield $P(O_1 : topic(AI) = \text{true } |O_2 : aboutLogic(BDLP) = \text{true}) \approx 0.83$. In contrast, the probability of occurence of the topic AI in our publication collection without any evidence amounts to $P(O_1 : topic(AI)\text{=true}) \approx 0.77$.

Our example query where it is asked for publications about AI, $?-O_1 : Publication(x)\text{=true}, O_1 : keyword(x, AI)\text{=true.}$, from section 1.1 is a non-ground query. It yields two valid groundings $\{x/BDLP\}$ and $\{x/BN\}$. Querying the Bayesian Network gives

$$\begin{aligned} &P(O_1 : Publication(BDLP)\text{=true}, O_1 : keyword(BDLP, AI)\text{=true}) \\ &\qquad = 0.55 \\ &P(O_1 : Publication(BN)\text{=true}, O_1 : keyword(BN, AI)\text{=true}) \\ &\qquad = 0.46 \end{aligned}$$

As mentioned above, queries with nonground query atoms can be used for Information Retrieval where the probabilities are used for ranking. In our example, the publication on BDLPs has a higher ranking, because it has been identified to be about both, Logic and Probabilities, with high probability, while the publication on BNs has been identified to be rather about probabilities and not about Logic.

5. RELATED WORK

There exists different kinds of probabilistic logics. However, most of them are not useful in the area of the Semantic Web as they do not provide a tight formal integration with a Semantic Web language or a subset thereof. Our intention with BDLPs is to present an elegant way of integrating the knowledge representation formalism of DLPs with probabilities. We chose the logic programming syntax of DLPs because corresponding reasoners allow much more efficient instance retrieval than Description Logics reasoners. BLPs lend themselves as a KR formalism that allows a probabilistic extension of the logic programming view of DLPs in a very straightforward and intuitive way. This is not possible with other kinds of probabilistic logics. A survey on probabilistic logics that focus on the logic programming and the relational logic paradigm can be found in [20].

In [21], pDatalog, a probabilistic extension of Datalog has been presented. One difference to our approach is that Datalog with stratified negation has been extended. In contrast, BDLPs do not have negation at all because the DLP fragment lying in the Description Logics and Logic Programming paradigm does not contain negation. Another difference is the probabilistic model behind pDatalog which does not extend Logic Programming in such a straightforward way like it is done in BDLPs. E.g. in pDatalog only facts have a probability attached and the probability of rules is defined procedurally instead of declaratively like in BLPs and BDLPs. The semantics of pDatalog considers a set of possible boolean worlds and a probability distribution on this set. In contrast, our approach views each ground fact as a (possibly non-boolean) random variable which can be either true or false and probabilistic queries are processed by reasoning algorithms for Bayesian Networks which al-

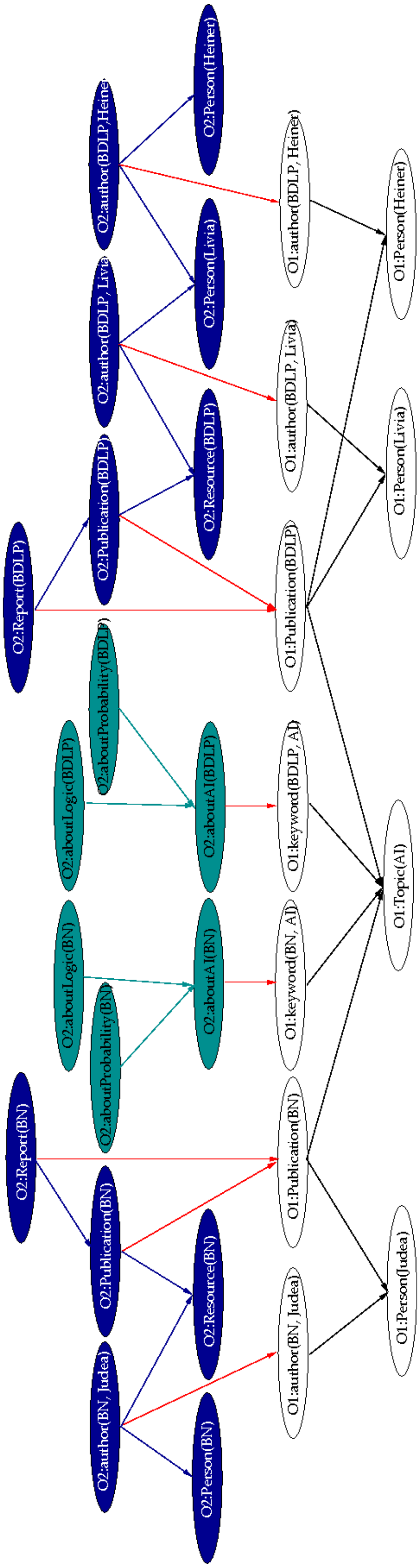

Figure 1: The Bayesian Network of our example.

low a compact and efficient representation of a probabilistic domain and thus can be expected to allow more efficient reaesoning facilities. Note that pDatalog is used in the tool HySpirit [22] for probabilistic Retrieval of documents.

Within the context of the Semantic Web, a couple of probabilistic formalisms that extend Semantic Web Languages or subsets thereof have been proposed. However, most of them do not aim at integration, i.e. neither at the integration of Logic Programming and Description Logics nor at the integration of different knowledge bases or ontologies. Examples of such formalisms are [23, 24, 25]. [25] proposes a probabilistic extension of a subset of Description Logic Programs Formalisms basing on pDatalog and is therefore very much related. However, the authors of this formalism do not aim at any kind of integration. The formalism is intended to provide a probabilistic extension of a subset of the DLP fragment of OWL in order to enable the representation of uncertain information.

Formalisms that aim at the integration of Description Logics and Logic Programming have been proposed recently as well. In [26, 27], a formalism has been proposed that consists of a description logics knowledge base corresponding to OWL-DL and a disjunctive datalog program with negation under stratified, well-founded and answer set semantics. The logic program has been probabilistically extended with independent choice logic. This formalism is much more expressive than ours and therefore, reasoning with it has a much higher complexity. Firstly, it contains a description logic knowledge base with the same expressivity as OWL-DL. Secondly, its logic programming component contains negation with different semantics and, thirdly, its query language is more expressive. For reasoning, a linear system has to be solved.

Formalisms that aim at probabilistically integrating knowledge bases on different peers have only been proposed for OWL and RDF ontologies as yet. An approach for Ontology Mediation by means of Bayesian Networks has been presented in [28]. However, the authors rather want to enhance existing mappings (cf. the name of the system **O**ntology **M**apping **EN**hancer). They do not provide an integrated framework on reasoning with the mappings and the ontologies. Furthermore, the language they use for capturing the mappings is very simple and similar to RDF Schema. A probabilistic framework for Information Integration and Retrieval on the Semantic Web does not exist at all yet. In [29], the oMap framework has been presented which aims at being such a framework. But up to now, the only substantial contribution to such a framework is a tool for learning mappings consisting of simple pDatalog rules between OWL ontologies. Ideas on how to reason with the ontologies and rules are completely missing.

6. CONCLUSIONS

In this paper, we have presented a framework for probabilistic Information Processing on the Semantic Web that is capable of representing ontologies (to a certain extent) as well as uncertain mappings and results of statistical instance classification and ontology learning. For this purpose, we use a formalism that is a probabilistic extension of Description Logic Programs [10]. The formalism is a subset of Bayesian Logic Programs (BLPs) [11] and can thus resort to algorithms developed for BLPs. Due to the sacrifice of some of the expressive power of both, existing ontology languages and of (Bayesian) logic programs, a simple formalism has been yielded that can easily be implemented. In future work, we will further explore the use of more expressive formalisms. In particular, we could allow the probabilistic part of the model to have the full complexity of the datalog fragment of BLPs or even more expressive formalisms.

We also showed in this paper how to perform Information Retrieval in our framework and extended the semantics of BDLPs (note that this extension can be applied for BLPs as well) to enable processing of queries that contain nonground bayesian atoms. In future work, we want to investigate information retrieval in our framework further and test different probabilistic retrieval models.

Even in the rather simple approach we took in this paper, the complexity of reasoning can be a serious problem. It is well known that reasoning in Bayesian networks is intractable in the general case (in particular it is in numberP which is the complexity class for counting solutions to decision problems lying in the complexity class NP). It can also happen that in general settings, cycles appear in mappings and thus the resulting Bayesian Network which makes exact inference intractable. Further, the size of the model of a BDLP program and thus the Bayesian network can be exponential in the size of the knowledge base. These factors force us to think about possible optimizations. The fact that parts of the network encode logical rather than probabilistic information can be exploited for this purpose. In particular, we will explore the following options

- Pruning techniques such as the one proposed in [30] might be used to tailor the inference process before the construction of the Bayesian Network to infer only the part relevant for answering a certain query and given certain evidence.
- Approximate inference algorithms have been shown to have good performance on networks even in the presence of cycles [31] and modifications thereof might prove themselves with BDLPs.
- We can try to reduce the computation time by distributing and parallelizing the computation amongst the different information sources involved.

In [32] it has been discussed how BLP programs can directly be learned from example data. This makes our approach a first step into integrating Semantic Web languages and deductive databases with machine learning techniques as we can use the techniques presented in [32] for directly learning parts of the model thus avoiding the step of transforming the results of other learning approaches into our framework.

7. REFERENCES

[1] Lassila, O., Swick, R.R.: Resource Description Framework (RDF) model and syntax specification. W3C Recommendation. URL: http://www.w3.org/TR/REC-rdf-syntax (1999)

[2] Manola, F., Miller, E.: RDF Primer. W3C Recommendation. URL: http://www.w3.org/TR/rdf-primer/ (2004)

[3] Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuiness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL Web Ontology Language Reference. W3C Recommendation. URL: http://www.w3.org/TR/owl-ref/ (2004)

[4] Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. URL: http://www.w3.org/Submission/SWRL/ (2004)

[5] Angele, J., Boley, H., de Bruijn, J., Fensel, D., Hitzler, P., Kifer, M., Krummenacher, R., Lausen, H., Polleres, A., Studer, R.: Web rule language. URL: http://www.w3.org/Submission/WRL/ (2005)

[6] Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys (2002)

[7] Maedche, A., Staab, S.: Ontology Learning. In: Handbook on Ontologies. Springer (2004)

[8] Euzenat, J., Shvaiko, P.: Ontology Matching. Springer (2007) To appear.

[9] Glimm, B., Horrocks, I., Lutz, C., Sattler, U.: Conjunctive query answering for the description logic SHIQ. In: Proc. of the 20th Joint Conf. on Artificial Intelligence. (2007)

[10] Grosof, B.N., Horrocks, I., Volz, R., Decker, S.: Description Logic Programs: combining logic programs with description logic. In: Proc. of the 12th international conference on World Wide Web. (2003)

[11] Kersting, K., Raedt, L.D.: Bayesian Logic Programs. Technical report, Albert-Ludwigs University, Freiburg (2001)

[12] Predoiu, L.: Information integration with bayesian description logic programs. In: Proc. of 3rd IIWeb Workshop for Information Integration on the Web. (2006)

[13] Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. In Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: Description Logic Handbook, Cambridge University Press (2003)

[14] Lloyd, J.W.: Foundations of Logic Programming, 2nd Edition. Springer (1987)

[15] Volz, R.: Web Ontology Reasoning with Logic Databases. PhD thesis, University of Karlsruhe (2004)

[16] Ullman, J.D.: Information integration using logical views. In Afrati, F.N., Kolaitis, P.G., eds.: Proc. of 6th International Conf. on Database Theory (ICDT. (1997)

[17] Jensen, F.V.: Introduction to Bayesian Networks. Springer-Verlag New York, Inc. (1996)

[18] Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Ontology Matching A Machine Learning Approach. In: Handbook on Ontologies in Information Systems. Springer (2004)

[19] Lacher, M.S., Groh, G.: Facilitating the exchange of explicit knowledge through ontology mappings. In: Proc. of the 14th International FLAIRS conference, Key West, Florida, USA (2001)

[20] Raedt, L.D., Kersting, K.: Probabilistic logic learning. SIGKDD Explorations (2003)

[21] Fuhr, N.: Probabilistic Datalog: Implementing Logical Information Retrieval for Advanced Applications. Journal of the American Society for Information

Science **51**(2) (2000)

[22] Röllecke, T., Lübeck, R., Kazai, G.: The hyspirit retrieval platform. In: Proc. of the 24th International ACM SIGIR Conf. on Research and Development in Information Retrieval. (2001)

[23] Ding, Z., Peng, Y.: A Probabilistic Extension to Ontology Language OWL. In: Proceedings of HICSS '04, IEEE Computer Society (2004)

[24] Udrea, O., Subrahmanian, V.S., Majkic, Z.: Probabilistic rdf. In: Proc. of the IEEE International Conference on Information Reuse and Integration. (2006) 172–177

[25] Nottelmann, H., Fuhr, N.: Adding Probabilities and Rules to OWL Lite Subsets based on Probabilistic Datalog. Uncertainty, Fuzziness and Knowledge-Based Systems **14**(1) (2006)

[26] Lukasiewicz, T.: Probabilistic Description Logic Programs. In: Proc. of the conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty. (2005)

[27] Lukasiewicz, T.: Stratified Probabilistic Description Logic Programs. In: Proc. of the ISWC Workshop on Uncertainty Reasoning for the Semantic Web. (2005)

[28] Mitra, P., Noy, N.F., Jaiswal, A.: OMEN: A Probabilistic Ontology Mapping Tool. In: International Semantic Web Conference. (2005)

[29] Straccia, U., Troncy, R.: Towards Distributed Information Retrieval in the Semantic Web: Query Reformulation Using the oMAP Framework. In: Proc. of the 3rd European Semantic Web Conference. (2006)

[30] Shachter, R.D.: Bayes-ball: The rational pastime. In: Proc. of 14th Conference of Uncertainty in Artificial Intelligence. (1998)

[31] Murphy, K.P., Weiss, Y., Jordan, M.I.: Loopy Belief Propagation for Approximate Inference: An Empirical Study. In: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence. (1999)

[32] Kersting, K., Raedt, L.D.: Bayesian Logic Programming:Theory and Tool. In: An Introduction to Statistical Relational Learning. Unpublished yet (2005) To appear.