Probabilistic Models for the Semantic Web – A Survey

Livia Predoiu,
Computer Science Institute, University of Mannheim,
A5, 6 68159 Mannheim, Germany
Phone: +49 621 181 2516, Fax: +49 621 181 2679
livia@informatik.uni-mannheim.de

Heiner Stuckenschmidt
Computer Science Institute, University of Mannheim,
A5, 6 68159 Mannheim, Germany
Phone: +49 621 181 2530, Fax: +49 621 181 2679
heiner@informatik.uni-mannheim.de

Probabilistic Models for the Semantic Web – A Survey

Abstract

Recently, there has been an increasing interest in formalisms for representing uncertain information on the Semantic Web. This interest is triggered by the observation that knowledge on the web is not always crisp and we have to be able to deal with incomplete, inconsistent and vague information. The treatment of this kind of information requires new approaches for knowledge representation and reasoning on the web as existing Semantic Web languages are based on classical logic which is known to be inadequate for representing uncertainty in many cases. While different general approaches for extending Semantic Web languages with the ability to represent uncertainty are explored, we focus our attention on probabilistic approaches. We survey existing proposals for extending semantic web languages or formalisms underlying Semantic Web languages in terms of their expressive power, reasoning capabilities as well as their suitability for supporting typical tasks associated with the Semantic Web.

INTRODUCTION

The Semantic Web is an extension of the World Wide Web that allows for expressing the semantics and not only the markup of data. By means of the representation of the semantics of data, new and not explictly stated information can be derived by means of reasoners. In this way, software agents can use and integrate information automatically. As common web languages like (X)HTML and XML are not enough for this purpose (Decker et al., 2000), Semantic Web languages have been standardised (RDF, RDF Schema and OWL), proposed (e.g. WRL, SWRL) and new ones are still being devised. However, most languages that are intended for usage on the Semantic Web are deterministic and cannot represent uncertainty. Currently, there is a growing interest in probabilistic extensions of Semantic Web languages. People start to realize that there is inherently probabilistic knowledge that needs to be represented on the Semantic Web. In the following, we briefly describe five areas where probabilistic information plays a role in the context of the Semantic Web.

Representing inherently uncertain Information: Not all of the information that needs to be represented on the Semantic Web is given in terms of definite statements. E.g. statistical information can provide insights to data to be shared on the Semantic Web. Ontological information attached with statistical values like the percentage of people in a population that are of a certain age can help answer queries about the correlation between this age and a certain chronical disease. There are many situations in which the use of this statistical information could be used to improve the behaviour of intelligent systems. An example would be a recommender System that points the user to certain information based on information about the age group.

Ontology Learning: The manual creation of ontologies has been identified as one of the main bottlenecks on the Semantic Web. In order to overcome this problem several researchers are investigating methods for automatically learning ontologies from texts. Existing approach normally use a combination of NLP and text mining techniques (Maedche & Staab, 2004). Typical tasks are the detection of synonyms and of subclass relations using clustering techniques and association rule mining. In both fields, the result of the mining process can be interpreted in terms of a probabilistic judgement of the correctness of the learned relation.

Document Classification: Document Classification can be seen as a special case of ontology learning called Ontology population. Today a major part of the information on the web is present in terms of documents (Web Pages, PDF Documents etc.). A common way of linking documents to knowledge encoded in ontologies is to assign individual documents to one or more concepts representing its content. Different machine learning techniques have been applied to this problem (Sebastiani, 2002). The most commonly used is the use of naïve Bayes classifiers that estimate the probability of a document belonging to a topic based on the occurrence of terms in sample documents.

Ontology Matching: Different sources often use different ontologies to organize their information. In the case of documents, these are often classified according to different topic hierarchies. In order to be able to access information across these different sources, semantic correspondences between the classes in the corresponding ontologies have to be determined and encoded in mappings that can be used to access information across the sources. Recently, a number of approaches for automatically determining such mappings have been proposed (Euzenat & Shvaiko, 2007). Some of the most successful ones use machine learning techniques to compute the probability that two classes represent the same information.

Ontology Mapping Usage for Information Integration: The usage of the mappings that have been found by matchers as explained in the paragraph above is currently mainly deterministic. Although the mappings are attached with a confidence that expresses how sure the matcher is that the mapping holds, the usage of those mappings consists of a preprocessing step: All mappings that have a confidence value above that threshold are considered deterministically true and all mappings that have a confidence value below that threshold are considered deterministically false. However, there is evidence that this kind of usage is error prone, especially when mappings are composed over several ontologies.

The five examples above clearly demonstrate the importance of probabilistic information in the context of the Semantic Web. E.g., in order to use the learned structures effectively, we need ways to represent and reason about the probabilities assigned to them. Most existing Semantic Web languages are mainly based on classical, deterministic logic and do not support this aspect. In the following, we review a number of general approaches for combining logical languages with probabilistic models and discuss existing proposals for extending semantic web languages with probabilistic information in more details.

Aim and Scope

In this paper, we review existing proposals to extend Semantic Web languages with the capability to handle uncertain information to better deal with the situations mentioned above. There are many ways of representing and dealing with uncertainty. In this paper, we restrict our attention to approaches that use probabilistic methods for representing uncertain information. In particular, we will not cover recent proposals for fuzzy-logic based extensions of semantic web languages. We will also not discuss nonmonotonic and non-standard logics for representing uncertainty unless they are based on a probabilistic semantics. We focus on these approaches, because we believe that probabilistic methods are a natural choice for representing the kinds of uncertainty we often find on the web. A strong motivation is the awareness that Semantic Web technology could greatly benefit from a tighter integration with machine learning and information retrieval techniques which are mostly based on probabilistic models. Probabilities have been criticised mostly due to the fact that people are very bad in providing correct judgements of the probability of events. We think that on the Semantic Web, this argument does not apply, because the aim here is not to use subjective judgements of probability but to provide mechanisms to represent inherently statistical information found on the web or produced by machine learning and matchers. The five examples above clearly demonstrate the importance of probabilistic information in the context of the Semantic Web. In this paper, we review a number of proposals for extending logical languages with probabilistic information in more details. We focus on

1. approaches that directly extend Semantic Web languages, in particular RDF and OWL
2. approaches that extend formalisms that have a very close connection to Semantic Web languages or that have explicitly designed to be used on the Semantic Web by the authors.

In the latter category, on the one hand, we cover probabilistic extensions of Description Logics which are commonly accepted as being the formal basis of OWL. Even though most approaches only cover logics that are much weaker than OWL, the methods proposed can direcly be applied to the corresponding subset of OWL. The second kind of languages we consider are rule languages. Although there is not yet an official rule language for the Semantic Web, it is clear, however that rule languages have an important role to play on the Semantic Web. As the area of rule languages is also very broad, we focussed on approaches that have been developed for the Semantic Web. Due to the fact that ontologies and thus Description Logics play a very important role in the Semantic Web, all approaches that extend are that combine rules and ontologies in some way. We restrict ourselves to probabilistic logics that allow combinations of rule and ontologies also with our application example in the area of ontology matching and ontology mapping usage in mind. This application example is presented below.

When talking about the different approaches, we will distinguish between the logical language which is used to describe knowledge and the probabilistic model used to assign probabilisties to certain assertions of the logical language. Based on this distinction, we discuss the following issues of the different approaches:

- the general probabilistic model used
- Expressiveness of the logical language
- kind of logical sentences that can be assigned a probability
- reasoning support and expected efficiency for large scale models

In order to evaluate the applicability of the respective approaches, we also consider an example scenario from the area of ontology matching and ontology mapping usage. This example illustrates also the inherent uncertainty of mappings and why this uncertainty needs to be taken into account for reasoning. Our example is based on two ontologies used in the Ontology Alignment Evaluation Challenge[1]. Assume a situation where a user is looking for publications about AI based on two ontologies O_1 and O_2.

[1] For the sake of simplicity we are considering here only a part of the ontologies. The complete ontologies can be found at http://oaei.ontologymatching.org/.

Let O_1 be specified by the following axiom which specify that for each publication there is a keayword which is a subject. Furtermore, there is a publication about the Semantic Web which has the keyword Artificial Intelligence.

(1) Publication $\sqsubseteq \forall$keyword.Subject
(2) (SW, AI): keyword
(3) SW: Publication

Let O_2 be specified by the following axioms which specify that reports are always publications and every concept in the knowledge base is about some topic. Furthermore, there is one report about Logic Programming and a publication about Description Logics. Both are about Logics.

(4) Report $\sqsubseteq$ Publication
(5) $\top \sqsubseteq \forall$about.Topic
(6) BN:Report
(7) DL: Publication
(8) (BN, Probability): about
(9) (DL, Logics): about

Without loss of generality, we can assume that O_1 is the local ontology, i.e. the ontology being queried explicitly by the user. In order to integrate the information which is stored in both ontologies, mappings are needed. With a probabilistic matcher like GLUE (Doan et al., 2003) mappings can be found which map the second ontology O_2 to our local ontology O_1.

(10) O1: Publication(x) $\leftarrow$ O2: Publication(x) with probability 0.8
(11) O1: Publication(x) $\leftarrow$ O2: Report(x) with probability 0.9
(12) O1: Subject(x) $\leftarrow$ O2: Topic(x) with probability 0.9
(13) O1: keyword(x, y) $\leftarrow$ O2: about(x, y) with probability 0.8

The mapping (10) basically says that all instances that are belonging to the concept Publication in O_2 are also belonging to the concept publication of O_1 with the probability 0,9. Due to the Kolmogorov axioms of probability theory, the probability that instances belonging to O_2:Publication do not belong to O_1:Publication is 0,1. For completeness, the probability that instances that do not belong to O_2:Publication belong to O_1:Publication needs to be derived by a matcher as well. GLUE can be modified such that it conforms to this requirement. Let's assume that for (10) this probability is 0.2, for (11) it is 0.4, for (12) it is 0.3 and for (13) it is 0.4.

If we pose a query, we want to get an answer that integrates the information of both ontologies. So, in our example, if we query our local ontology for all publications:

Publication(x) $\wedge$ keyword(x, AI)

we want to get also all relevant publications mentioned only in the second ontology. The answer here is

- the publication about Semantic Web with probability 1.0 because it is mentioned in the local ontology and no mapping has been used for deriving it
- the publication about Logic Programming in the second ontology which was derived by two mappings (10) and (11) and thus gets an accumulated probability of 0,75

- the publication about Description Logics which was derived by only one mapping (10) and has only the probability 0,44

The computation is based on the semantics of the Bayesian Description Logic Programming formalism and shows nicely the importance of the consideration of uncertainty in the area of information integration in the Semantic Web. Without the consideration of the uncertainty each mapping is associated with, all 3 answers would be needed to be treated in the same way although the publication about the Semantic Web is much more relevant than the one about Description Logics. Furthermore, if mapping composition is considered with mapping chains over several ontologies, mappings with rather low probabilities can contribute to an anwer with a rather high probability.
Another requirement for a mapping language is the possibility to express mappings between individuals. E.g. in our example, publications of O_2 that are about probabilities are less probable to be publications in O_1 that deal with AI than publications of O_1 that are about logics:

(14)	O1:keyword(x, AI) ← O2: about(x, Probability)	e.g. with probability 0.7
(15)	O1:keyword(x, AI) ← O2: about(x, Logics)	e.g. with probability 0.9

It is immediately clear that such mappings need to be expressed for a comprehensive handling of mappings in the Semantic Web area. Mappings that do not involve any variables might also be necessary to be expressed in certain scenarios. When we investige the different probabilistic extensions of Semantic Web languages, we also have a look at the applicability of the formalisms for the area of Information Integration as presented in this example.
This chapter is structured as follows. In the next section, we present an overview of current Semantic Web languages and related formalisms that are the basis for the logical languages used in the different approaches discussed later in the paper. We also provide a brief introduction to some basic probabilistic models that are used in the different approaches. Based on these basic methods, we discuss proposals for probabilistic languages for the Semantic Web, in the section "probabilistic extensions of Semantic Web languages" below in this chapter. We start with proposals for extending RDF and OWL. Afterwards, we discuss approaches for extending related formalisms woth notions of probability, namely Description Logics and different Rule Languages. We conclude the chapter with a critical review of the state of the art and an analysis of directions for future research.

PRELIMINARIES AND BACKGROUND

In this section, we introduce the reader to the state of the art in current Semantic Web languages and the background on the probabilistic models used in the probabilistic extensions surveyed below in the section "probabilistic extensions of Semantic Web languages" below in this chapter.

Current Semantic Web Languages

So far, the development of languages for the Semantic Web was dominated by traditional views on metadata models and logic-based knowledge representation. The major languages that have been developed are RDF/RDF Schema (Lassila & Swick, 1999; Manola & Miller, 2004) for representing metadata and the Web Ontology language OWL (Bechhofer et al., 2004) for representing terminological knowledge in terms of ontologies. The Web Ontology language OWL has its root in the formalism of Description Logics, a decidable subset of first-order logic that contains special constructs for defining classes in terms of necessary and sufficient conditions based on predicates representing binary relations between instances of different classes. More specifically, OWL corresponds to particular Description Logic variants (OWL Lite corresponds to $\mathcal{SHIF}$(D) and OWL DL corresponds to $\mathcal{SHOIN}$(D) (Horrocks et al., 2003)) in the sense that reasoning in OWL can be reduced to checking satisfiability in this logic (Horrocks & Patel-Schneider, 2004). Similarly, the semantics of RDF can be modelled with First-Order Logics, Description Logics and Datalog (Fikes & Guiness, 2001), (de Bruijn & Heymans, 2007).

Recently the need for rule languages on the Semantic Web has been recognized. Rule languages complement Description Logics as they allow to represent kinds of axioms not expressible in $\mathcal{SHIF}$ and $\mathcal{SHOIN}$ (e.g. property chaining (cf. e.g. (Horrocks, 2005)). Thus, several rule language proposals for the Semantic Web have emerged, examples being the Semantic Web Rule Language SWRL (Horrocks et al., 2005) and the Web Rule Language WRL (Angele et al., 2005) for describing domain-dependent inference rules. The Semantic Web Rule language allows the definition of conjunctive rules over the concepts and binary relations or roles, respectively, which are contained in an OWL ontology (Horrocks et al., 2005). Finally, similar to OWL, WRL is a layered language consisting of three languages, one being a superset of the other. WRL-Core which is the least subset of the WRL language family corresponds to a subset of OWL which lies in the language of Logic Programming (also known as the DLP fragment (Grosof et al., 2003)). WRL-Flight contains WRL-Core and is a Datalog-based rule language. WRL-Full contains WRL-Flight and is a rule language with function symbols and negation under the Well-Founded Semantics (Angele et al., 2005).

Description Logics which is represented by OWL in the Semantic Web and Logic Programming which is represented by a couple of W3C rule language proposals have both nice orthogonal properties and expressivity. Ways for combing both have been and still are investigated. Description Logics and Logic Programming have been found to have a common subset called Description Logic Programs (Grosof et al., 2003). Therefore, Description Logic Programs have a Logic Programming and a Description Logic syntax and wrappers can be used to translate them. Another subset of Description Logics and Logic Programming has been recently proposed that is called Horn-$\mathcal{SHIQ}$ (Hustadt et al., 2005) and is a strict superset of Description Logic Programs. Besides the investigation of the intersection of Description Logics and Logic Programming, a lot of research aims at a more comprehensive integration of both formalisms. Several approaches for enabling an interaction between logic programs and description logics exist. Usually, they consist of a Description Logics knowledge base and a Logic Program and the latter is equipped with special features for interacting with the Description Logics knowledge base. An example of such an approach where the Description Logic knowledge is an OWL Lite or OWL DL knowledge base is the formalism of Description Logic Programs under the answer set semantics by (Eiter et. al, 2004).

All of the languages mentioned above are logical languages with a classic model-theoretic semantics that makes a statement either true or false and have no means to represent uncertainty in any way.

Probabilistic Languages and Models

In the following, a short overview of the probabilistic models used for the languages in the sections below is presented. Those models are *Bayesian Networks*, *Bayesian Logic Programs, Independent Choice Logic*, *Probabilistic Datalog* and *Multi-Enity Bayesian Networks*. Some of these models are related to each other, e.g. Bayesian Networks can be considered as a subset of Bayesian Logic Programs because the latter provide a compact representation of the former in the same way like first-order logic does with sentential logic. Independent Choice Logic is a generalization and a superset of the formalism of Bayesian Logic Programs. The relationship of Bayesian Networks, Bayesian Logic Programs and Independent choice Logic with probabilistic Datalog is unclear. Multi Entity Bayesian Networks are comprising Bayesian Networks in the same way like Bayesian Logic Programs do. Multi Entity Bayesian Networks are more expressive than Bayesian Logic Programs, but it is unclear whether there is a semantical subset relationship. The relationship between Multi Entitity Relationship Programs and Independent Choice Logic has not been investigated yet either. Multi Entitity Relationship Programs differ from Probabilistic Datalog by the usage of negation. Probabilistic Datalog uses well-founded negation and the closed world assumption while Multi Entity Relationship Programs model probabilistic First-Order Logic knowledge bases and employ classical negation as well as the open world assumption.

Bayesian Networks (BNs) - One of the best understood models for representing the joint probability distribution of a domain of interest is the model of Bayesian Networks (BNs) (Jensen, 2001). A BN is a compact representation of the joint probability distribution among a set of random variables in a domain under consideration. More precisely, a BN is a directed, acyclic graph with the random variables as nodes and direct influence relationships as arcs. Several exact or approximate algorithms for reasoning in Bayesian Networks exist. Exact inference has been proven to be NP-complete in the maximal number of parents of nodes in the network. A considerable amount of research effort has been spent on different issues like learning of the conditional probability tables/distributions of the nodes in the BN, learning the structure of a BN, etc. (Castillo et al., 1997; Cowell et al., 1999; Jensen, 2001). A BN has been found to correspond to a probabilistic extension of sentential definite clauses.
In the area of the Semantic Web where the same or similar knowledge can happen to be represented on different and independent peers and integrated reasoning and information usage requires mappings, cycles in the complete representation may occur. Unfortunately, BNs are not allowed to have directed cycles. For reasoning with BNs, a huge amount of free and commercial software tools and implementations exist.

Bayesian Logic Programs (BLPs) - Bayesian Logic Programs (Kersting & De Raedt, 2001) are an extension of Bayesian Networks to first-order definite clause logic and a probabilistic extension of definite first-order logic at the same time. A BLP consists of a set of rules and facts, i.e. a definite clause logic program. Each fact is associated with an a-priori probability and each rule with a conditional probability where the probability of the head atom is conditioned on the states of the body atoms. Each ground atom of the Herbrand Model of the definite clause logic program corresponds to a node in a corresponding Bayesian Network. The arcs are defined through the rules. For each valid ground rule, an arc from each node representing a body atom to the node respresenting the head atom exists in the corresponding Bayesian Network. Additionally, combining rules are defined in order to enable the combination of conditional probabilities of different valid ground rules with the same head atom. BLPs are defined to be acyclic. Therfore, the corresponding Bayesian Networks are acyclic as well. Reasoning with BLPs corresponds to deriving the Herbrand Model or the part of it which is relevant to the query and building the corresponding Bayesian Network. For BLPs, no complexity results have been published, yet. Currently, only one tool for reasoning with BLPs exists: the Balios engine (Kersting & Dick, 2004).

Independent Choice Logic (ICL) - Independent Choice Logic (Poole, 1997) is a logic that is built upon a given base logic that conforms to some restrictions and determines truth in the possible worlds defined by choice spaces. Possible worlds are built by choosing propositions from sets of independent choice alternatives. As base logic, Poole suggests acyclic logic programs under the stable model semantics. However, as we will see later in the subsections on probabilistic (disjunctive) description logic programs below, the approach works for other base logics as well.

An independent choice logic theory on a base logic is a pair (C, F) where C is a so-called choice space and F is a knowledge base in the base logic. C is a set of sets of ground atomic formulae from the language of the base logic such that for two choices $c_1, c_2 \in$ C, if $c_1 \neq c_2$ then $c_1 \cap c_2 = \emptyset$. The elements of C are called alternatives and are basically random variables. The elements of an alternative c are called atomic choices and are basically possible values for the random variable c. The semantics of ICL is defined in terms of possible worlds. A possible world corresponds to the selection of one element from each alternative. Such a selection is called total choice. The atoms that follow using the consequence relation of the base logic from these selected atoms together with the knowledge base of the base logic are true in this possible world. Reasoners for ICL are conceivable but depend on the base logic used. Also, the complexity for deciding consistency and query answering depends on the base logic used.

Multi-Entity Bayesian Networks (MEBNs) - Multi-entity Bayesian Networks (Laskey & Costa, 2005) extend the Bayesian Network model to full First-Order logic. In this way, graphical models with repeated sub-structures can be represented and a probability distribution over models of any consistent, finitely axiomatizable first-order theory can be expressed. With MEBN logic, entities that have attributes and are related to other entities can be represented. Features of entities and relationships among entities are random variables. The knowledge about attributes and relationships is expressed as a collection of MEBN fragments (MFrags) organized into MEBN theories (MTheories). An MFrag represents a conditional probability distribution and an MTheory is a set of MFrags that collectively satisfies consistency constraints ensuring the existence of a unique joint probability distribution over instances of the random variables represented in the MTheory. Possible queries are queries for the degree of belief in specific random variables given evidence random variables. The response to a query is computed by constructing a so-called situation-specific Bayesian Network that can be processed by a usual tool for Bayesian Networks. We are not aware of the existence of general complexity results for reasoning with the MEBN formalism. There are proposals for reasoning algorithms (Laskey, 2006) but no direct implementation of a reasoner for MEBN logic. But there is a translation of a subset of the MEBN formalism into probabilistic relational models implemented in the Quiddity*Suite (cf. http://www.iet.com/quiddity).

Probabilistic Datalog (pDatalog) - Probabilistic Datalog (Fuhr, 2000) is Datalog where each fact and each rule is extended with a probability which states the certainty of it being true. An important underlying assumption is that each element of the probabilistic Datalog program (i.e. every fact and every rule) are probabilistically independent from the other elements. Probabilistic Datalog has been equipped with a well-founded semantics. According to Nottelmann (2005), the probability of a rule can be seen as a conditional probability like with

Bayesian Logic Programs. However, while Bayesian Logic Programs allow an arbitrary set of states for the ground atoms in the Herbrand Base, probabilistic Datalog envisions just boolean states for the atoms. Bayesian Logic Programs do not allow any negation while probabilistic Datalog allows negation under the well-founded semantics. As yet, it is unclear whether probabilistic Datalog programs can be represented as Bayesian Networks. Probabilistic Datalog has been implementented in the HySpirit system (Roellecke et al, 2001) and query answering and the computation of probabilities is a two step process. First, the answers to the Datalog component of the query are computed by means of bottom-up evaluation that employs magic sets. Afterwards, the inclusion-exclusion principle is used to compute the probability of the resulting expressions in Disjunctive Normal Form. (Fuhr, 2000) states "Practical experimentation with HySpirit has shown that the evaluation of about 10 or more conjuncts is not feasible". However, recently, in (De Raedt et. al, 2007) an algorithm has been proposed that is able to perform approximate probabilistic reasoning by combining iterative deepening with binary decision diagrams and is very efficient. (De Raedt et. al, 2007) claims that "one can deal with up to 100000 conjuncts".

PROBABILISTIC EXTENSIONS OF SEMANTIC WEB LANGUAGES

In this section, we survey probabilistic extensions of RDF, RDF Schema and OWL which are W3C recommendations and thus correspond to a standard. We also have a look at probabilistic extensions of subsets of the Description Logics corresponding to OWL, i.e. $\mathcal{SHIF}$(D) and $\mathcal{SHOIN}$(D), and RDF (Schema).

Extensions of RDF

RDF can be considered as the most widely accepted Semantic Web language as it provides the syntactic basis for other Semantic Web languages. A proof for its success is the huge amount of software for processing RDF data that has been implemented up to now. Quite naturally, also some approaches for combining probabilities with RDF have been proposed. Fukushige (2005) proposes an RDF vocabulary for representing Bayesian Networks. In (Udrea et al., 2006), a probabilistic extension of acyclic RDF statements with a model-theoretic semantics and a fixpoint semantics has been proposed. While the first work concentrates on representation issues, the second work can be considered as probabilistic logic on its own.

Representing probabilistic information in RDF

In (Fukushige, 2005), a vocabulary extension of RDF has been proposed that is capable of representing the different elements of a Bayesian Network and link them to regular RDF statements. The vocabulary consists of a set of classes (*prob:Partition*, *prob:ProbabilisticStatement*, *prob:Clause*, *prob:Probability,*) and a set of predicates (e.g. *prob:predicate*, *prob:condition*, *prob:case*, *prob:about*) that can represent a Bayesian Network. This vocabulary allows to link statements to their probabilities, express conditional probabilities and more complex probabilistic statements.

Expressiveness The vocabulary can solely represent Bayesian Networks and can basically be considered as a syntactical interchange format for Bayesian Networks. Thus, as with Bayesian Networks, cyclic probabilistic descriptions cannot be represented. We deem this as a clear disadvantage, because we think that cyclic descriptions cannot be forbidden or avoided in such an open and unstructured environment like the web.

Reasoning and Efficiency As yet no reasoning support has been implemented. However, after having implemented a parser and wrapper for this vocabulary, in principle any tool for reasoning with Bayesian Networks can be used for reasoning.

Applicability to Information Integration In principle, the vocabulary can be used for representing mappings between ontologies in a similar way as done with Bayesian Description Logic Programs (see a more detailed presentation on Bayesian Description Logic Programs in the subsection entitled likewise below in this chapter). A huge disadvantage, however, is that Bayesian Networks are not properly integrated with RDF on the meta level: the vocabulary for representing Bayesian Networks uses RDF for its syntax without a tight coupling to the logical model of RDF. Therefore, RDF ontologies cannot be integrated with mappings expressed in this vocabulary properly. Clearly, with OWL ontologies, it is not possible either.

pRDF

In contrast to the former formalism that is intended to just provide a vocabulary for representing Bayesian Networks, pRDF is a formal probabilistic extension of RDF which corresponds to a of probabilistic logic on its own.

Expressiveness pRDF is a probabilistic extension of a subset of RDF and consists of a pair (S, I) with S being a pRDF schema and I being a pRDF instance base. A pRDF schema S is defined as a finite set consisting of probabilistic quadruples extending the RDF Schema builtin predicate *rdfs:subClassOf* and non-probabilistic triples using the RDF Schema builtin predicates *rdfs:subPropertyOf*, *rdfs:range* and *rdfs:domain*. This means that in pRDF neither the subproperty relationship nor domain and range restrictions can be defined probabilistically. A pRDF instance base I is a finite set of quadruples extending the RDF builtin *rdf:type* and arbitrary properties $p \in P$.
More precisely, pRDF allows the following kinds of probabilistic definitions:

- A sequel of axioms: $C(x) \rightarrow D_1(x), ..., C(x) \rightarrow D_n(x)$ and a probability distribution over the axioms in the sequel where $C \neq D_1 \neq ... \neq D_n$.
- A sequel of axioms : $P(inst, inst_1), ..., P(inst, inst_n)$ and a probability distribution over the axioms in this sequel, where $inst \neq inst_1 \neq ... \neq inst_n$ and P being either the RDF builtin ***rdf:type*** or an arbitrary user-defined property.

Furthermore, the following deterministic expressions are allowed:

- $R(x, y) \rightarrow R_2(x, y)$,
- $R(x, y) \rightarrow C(x)$,
- $R(x, y) \rightarrow C(y)$.

A disadvantage of this approach is that only a very small subset of RDF/S is supported by pRDF yielding a very low expressivity. Furthermore, pRDF instances are required to be acyclic, which again can only be realized in small and closed environments, but not on the Web as it is.

Reasoning and Efficiency A model theoretic semantics and a fixpoint operator has been defined basing on a t-norm (Fagin, 1999). Furthermore, a reasoner has been implemented that evaluates the fixpoint operator until the least fixpoint has been reached. The properties of a t-norm allow certain pruning strategies that are employed in the reasoning algorithms. Queries to pRDF instances are atomic, i.e. conjunctions cannot be dealt with. A query is a quadruple (i, p, S, P) where i can be an instance, p can be a property, S can be a set of instances i is related to via p and P can be a probability distribution for this sequel of property axioms. At most one of the elements of the quadruple is allowed to be a variable.
Unfortunately, for pRDF schema no query answering facility has been defined yet. The reasoning engine supports only reasoning with pRDF instances.

Appicability to Information Integration This formalism can be used for information integration with mappings. Mappings that map classes from one ontology to classes of the other ontology can be expressed. Also, mappings that map instances from one ontology to instances of another ontology can be expressed. But no mappings can be expressed that capture partly uninstancialized axioms like the ones in (14) and (15). However, the uncertainty attached to each mapping can be used for integrated reasoning with the mappings and the ontologies. But, due to the limited RDF support, not only the mappings but especially also the RDF ontologies which are to be mapped have a very limited expressivity.

Extensions of OWL

Quite naturally a number of proposals for using probabilistic knowledge on the Semantic Web focus on the extension of the Web Ontology Language as the central mechanism of representing complex knowledge in semantic web applications. When looking at the existing proposals, we see two fundamentally different approaches for combining OWL with probabilistic information.
The first kind of approach implements a loose coupling of the underlying semantics of OWL and probabilistic models. In particular these approaches use OWL as a language for talking about probabilistic models. An example of this approach is the work of Yang and Calmet (2006) that propose a minimal OWL ontology for representing random variables and dependencies between random variables with the corresponding conditional probabilities (Yang & Calmet, 2006). This allows the user to write down probabilistic models that correspond to Bayesian networks as instances of the OntoBayes Ontology. The encoding of the model in OWL makes it possible to explicitly link random variables to elements of an OWL ontology, a tighter integration on the formal level, however, is missing. A similar approach is proposed by Costa and Laskey (2006). They propose the PR-OWL model which is an OWL ontology for describing first order probabilistic models (Costa & Laskey, 2006). More specifically, the corresponding ontology models Multi-Entity Bayesian networks (Laskey & Costa, 2005) that define probability distributions over first-order theories in a modular way. Similar to OntoBayes, there is no formal integration of the two representation paradigms as OWL is used for encoding the general structure of Multi-entity Bayesian networks on the meta-level.

The second kind of approaches actually aims at enriching OWL ontologies with probabilistic information to support uncertain reasoning inside OWL ontologies. These approaches are comparable with the work on probabilistic extensions of Description Logics also presented in this section. A survey of the existing work reveals, however, that approaches that directly address OWL as an ontology language are less ambitious with respect to combining logical and probabilistic semantics that the work in the DL area. An example is the work of Holi and Hyvönnen (2006) that describe a framework for representing uncertainty in simple classification hierarchies using Bayesian networks. A slightly more expressive approach called BayesOWL is proposed by Ding and others (Ding et. al, 2006). They also consider Boolean operators as well as disjointness and equivalence of OWL classes and present an approach for constructing a Bayesian network from class expressions over these constructs. An interesting feature of BayesOWL is some existing work on learning and representing uncertain mappings between different BayesOWL ontologies reported in (Pan et al., 2005) which is an interesting alternative to existing matching tools.

In the following, we discuss PR-OWL and BayesOWL which are the most interesting representatives of the two general approaches to combining OWL and probabilistic models in more details.

PR-OWL

Expressiveness As mentioned above PR-OWL is an OWL Ontology that describes Multi-Entity Bayesian Networks. OWL is mainly used as a basis for a Protégé plugin for modelling MEBNs and as a language for representing MEBNs and linking them to domain ontologies encoded in OWL. On the other hand, MEBNs can be translated into Bayesian networks. This means that PR-OWL can be used to link OWL ontologies to Bayesian networks through the MEBN formalism. The Question about the expressiveness of PR-OWL therefore boils down to an analysis of the expressiveness of MEBNs as the actual representation model for uncertainty provided by the approach. According to the authors, MEBNs are capable of representing and reasoning about probabilistic information about any sentence in first-order logic by compiling it into a Bayesian network[2] but they define some restrictions on the nature of the theory, especially on the use of quantifiers. MEBNs specify random variables representing terms and organize them in so-called fragments that describe a certain aspect of the world. Fragments have an interface that defines the terms covered by the fragment. Each fragment defines the joint distribution over the random variables in terms of conditional probabilities encoded as part of a Bayesian network. Variables in terms can be instantiated with multiple constants each instantiation leading to a unique node in the resulting network. Logical formulas are modelled by special fragments that encode the semantics of Boolean operators, quantifiers and instantiation. Fragments are linked via shared terms and additional constraints ensure that only wanted instantiations take place.

It is quite hard to say whether MEBNs are expressive enough to capture probabilistic information about OWL ontologies. In principle it should be possible to translate each OWL ontology into first order logic and assign probabilities to conditional probabilities of the resulting model by encoding it as an MEBN. So far, it has not been investigated whether the restrictions on the use of quantifiers in MEBNs affect the representation of Ontologies. The language should be expressive enough top represent mappings between terms from different ontologies that go beyond simple concept-to concept mappings because it allows to combine

[2] Due to the semi-decidability of First-order logic this can only be true if the translation allows for networks of infinite size.

terms from different ontologies using arbitrary logical operators as well as the conditional probability of one given the other. It is less clear whether the representation of the mappings can be integrated with the definitions in the semantically consistent way that goes beyond simple reference to parts of the ontologies. In the same way, we could also represent the result of ontology learning methods in terms of conditional probabilities between terms. As fragments in MEBN need input in terms of instantiations of the interface, probabilistic information about instances (e.g. the probability that a paper is about a certain topic) cannot directly be encoded in MEBNs, we could, however find a workaround by explicitly representing a Bayesian classifier as a fragment.

Reasoning and Efficiency Reasoning in MEBNs is performed by constructing a Bayesian network from the instantiations of fragments. Inside each fragment, a network fragment is created that includes random variables and conditional probabilities for all input objects based on the network pattern specified in the fragment. Here, the actual conditional probability values depend on the number of input objects. The independent network fragments are then combined into a so-called situation-specific network, a Bayesian network that is customized to the given situation in terms of fragments actually instantiated and input objects. The basic reasoning task supported by this network is to compute the probability of one or more random variables given some evidence in terms of instantiations of some input random variables. This means that we can ask for the probability that certain terms are true or false given some knowledge about the truth or falseness of some other terms.
The basic problem of MEBNs when it comes to efficiency is the complexity of the logical language supported. In particular, this has an impact on the size of the situation specific network created as this network represents probabilistic information about all instances simultaneously instead of re-evaluating a standard network multiple times. In information retrieval applications, we often assume that information objects are independent of each other and do not have to be treated in parallel. Although, the bottom-up creation of this network ensures that only the part of the network that is actually needed to answer the query is constructed, this network can still have an infinite size. It would be interesting to identify tractable subsets of MEBNs that correspond to more tractable fragments of first order logic.

Applicability to Information Integration Again, as with the formalism of Fukushige (2005) presented above, the vocabulary can be used for representing very expressive mappings between ontologies in the MEBN formalism. However, as PR-OWL does not provide a proper integration of the formalism of MEBN and the logical basis of OWL on the meta level, OWL ontologies cannot be integrated with mappings expressed in this vocabulary properly. More specifically, as the connection between a statement in PR-OWL and a statement in OWL is not formalized, it is unclear how to perform the integration of ontologies that contain statements of both formalisms.

BayesOWL

Expressiveness BayesOWL is an approach for representing probabilistic information about class membership within OWL ontologies. The approach can be seen as an extension of Holi & Hyvönnen (2006). Both approaches support the representation of degrees of overlap between classes in terms of conditional probabilities of the form P(C|D) where C and D are class names. These statements denote the probability that an instance that is a member of D is also a member of C. The main feature of BayesOWL is that it does not only support simple class hierarchies but also class definitions of the following form

- Equivalence: $C(x) \leftrightarrow D(x)$
- Complement: $C(x) \leftrightarrow \neg D(x)$
- Disjointness: $C(x) \rightarrow \neg D(x)$
- Intersection: $C(x) \leftrightarrow D(x) \wedge E(x)$
- Union: $C(x) \leftrightarrow D(x) \vee E(x)$

This means that BayesOWL is actually a probabilistic extension of propositional logic rather than more expressive description logics. This is a quite a strong restriction as it means that we cannot represent probabilistic information about any relations except the subsumption relation. This limits the applicability to scenarios where we are only interested in the classification of information objects and not in relations between them. This means that the approach is not suitable to support the reasoning about structured information which plays an important role in many semantic web applications.

Reasoning and Efficiency The basic reasoning task associated to BayesOWL is given some evidence for an object in terms of its classification to determine membership probabilities for all the classes in the ontology. For this purpose a Bayesian network is constructed from the definitions in the model. As in PR-OWL network nodes with a predefined conditional probability table are used to represent Boolean Operators. This computation is done using iterative proportional fitting, a special technique from statistics that selects a probability distribution that best fits the conditional probabilities given in the network. This approach is quite different from the other approaches presented in this survey as the inference is not guided by a specific query. This can be an advantage if many queries about different aspects of the model are issued; we can expect it to be unnecessarily complex if we are only interested in very specific aspects of the model as the method will also compute probabilities that do not have an influence on the variable. Despite this fact, the use of Bayesian networks for implanting probabilistic reasoning can be expected to be relatively efficient. A special feature of BayesOWL is that it allows including probabilistic mappings between different ontologies into the inference procedure (Pan et al., 2005). Mappings are represented in terms of conditional probability statements that include concepts from different ontologies. The probabilistic influence of these statements on the distributions is used to update the distribution in the mapped ontologies. The conditional probabilities used in the mappings can be created using statistical learning methods.
In summary, the approach is well suited for applications that use rather simple classifications of information items such as documents that are classified according to a topic hierarchy. It supports the representation and semi-automated mapping of such hierarchies. As soon as the application demands for more structural information such as document metadata, the approach reaches it limits in terms of the inability to represent information about relations.

Applicability to Information Integration This formalism provides an integration between Bayesian Networks and OWL and thus it can be used for expressing uncertain mappings between OWL ontologies and for using those mappings for integrating the information distributed over the ontologies. As only class definitions are supported, however, neither the mappings nor the ontologies themselves can contain instances which is a severe drawback of this approach. Also, the expressivity on the schema level is very low in general and thus only a very small subset of OWL can be used for expressing ontologies to be mapped (and mappings).

Extensions of Description Logics

There have been a number of approaches for extending description logics with probabilistic information in the earlier days of description logics. Heinsohn (Heinsohn, 1991) was one of the first to propose a probabilistic notion of subsumption for the logic ALC. Jaeger (Jaeger, 1994) investigated some general problems connected with the extension of T-Boxes and A-Boxes with objective and subjective probabilities and proposed a general method for reasoning with probabilistic information in terms of probability intervals attached to Description logic axioms. Recently, Giugno and Lukasiewicz proposed a probabilistic extension of the logic SHOQ along the lines sketched by Jäger (Giugno & Lukasiewicz, 2002). A major advantage of this approach is the integrated treatment of probabilistic information about Conceptual and Instance knowledge based on the use of nominals in terminological axioms that can be used to model uncertain information about instances and relations. An alternative way of combining description logics with probabilistic information has been proposed by Koller et al. (1997). In contrast to the approaches mentioned above, the P-CLASSIC approach is not based on probability intervals. Instead it uses a complete specification of the probability distribution in terms of a Bayesian network which nodes correspond to concept expressions in the CLASSIC description logic. Bayesian networks have also been used in connection with less expressive logics such as TDL (Yelland, 2000). The approaches for encoding probabilities in concept hierarchies using Bayesian networks described in the section “preliminaries and background” can be seen as a simple special case of these approaches.
We can see two general approaches for extending description logics with probabilistic information. The first is based on probability intervals describing the validity of concept inclusion axioms, the other one is based on the use of Bayesian networks for assessing and relating the probability of different features of the terminological model. In the following, we will restrict our discussion to representative approaches of these different strategies, namely P-SHOQ and P-CLASSIC.

P-$\mathcal{SHOQ}$(D)

Expressiveness P-$\mathcal{SHOQ}$(D) is based on the description logics $\mathcal{SHOQ}$(D) which is very close to the description logic which provides the semantics of OWL. The only feature of OWL that is not contained in the language is the use of inverse roles. In particlar, the language also supports datatypes in the same way as OWL does. Probabilistic information is represented by statements of the form (C|D)[l,u] where C and D are concept expressions in SHOQ(D) and l and u are the maximal and the minimal probability that an instance of D is also an instance of C. Using this general scheme, different kinds of knowledge can be represented, for instance:

1. The probability that C is subsumed by D P(C(x)|D(x))
2. The probability that a particular individual o is a member of a concept C P(C(o))
3. The probability that an individual o is related to an instance of a concept C P(R(o,x)|C(x))
4. The probability that two individuals o and o’ are related P(R(o,o’))

From a representational point of view, P-$\mathcal{SHOQ}$(D) offers a lot of possibilities for supporting the task mentioned in the motivation. For the case of overlapping ontologies uncertain mappings between concepts in different ontologies can be represented using probabilistic subsumption statements of the form P(i:C(x)| j:D(x)) where C is a concept from ontology i and D a concept from ontology j. Concerning the task of ontology learning, the language is expressive enough to capture typical information that is determined in the learning process such as the concept hierarchy. We can also represent uncertain information about the range of concepts. The lack of inverse relations in the language, however, makes it impossible to represent domain restrictions. The use of nominals allows us to represent the results of instance learning both for concept and relation instances using statement 3 and 4 mentioned above.

Reasoning and Efficiency Reasoning in P-$\mathcal{SHOQ}$ is based on a function μ that maps every instance of the interpretation domain Δ on a number in [0,1] such that the value of this function for all elements in Δ sum up to 1. The Probability Pr(C) of a concept expression C is defined as the sum of all μ values of the instances of C. Based on this semantics a number of reasoning task have been defined that can be solved using appropriate inference procedures. At the most basic level, the tasks supported by the language are to determine whether a given knowledge base is consistent and to compute the upper and lower bounds l and u of a conditional probability statement $P(C(x)|D(x)) \in [l,u]$. Computing these bounds in based on independent choice logic. Different choices are specified by the possible semantic relations that could hold between any pair of concepts. This definition of choices leads to two linear equation systems whose solutions are the upper and the lower bound of the probability. Solving the equation system involves reasoning in $\mathcal{SHOQ}$(D) for determining the possible choices.
Based on this general method for computing upper and lower bounds a number of reasoning tasks that generalize standard reasoning tasks in Description Logics can be defined. In particular, the approach supports the following tasks

- **Concept satisfiability:** in particular decide whether $P(\exists x{:}C(x)) \in [0,0]$ does not follow
- **Concept Subsumption:** given two concepts C and D compute l and u such that $P(C|D) \in [l,u]$ follows from the knowledge base
- **Concept Membership:** given an instance o and a concept C compute l and u such that $P(C(o)) \in [l,u]$ follows from the knowledge base.
- **Role Membership:** given two instances o and o' and a relation R compute l and u such that $P(R(o,o')) \in [l,u]$ follows from the knowledge base

These reasoning tasks provide a suitable basis for supporting tasks such as probabilistic data retrieval across different ontologies. In particular, we can formulate queries as concept expressions in $\mathcal{SHOQ}$(D) and compute the probabilities that certain instances are members of this query concept. Probabilitistic information originating from uncertain mappings and classifications provide background constraints for this reasoning task. A potential problem of the approach with respect to the retrieval scenario is the ability to use the probabilities as a basis for ranking. As the approach is based on intervals rather than exact probabilities, there is no total order on the results that could be used for this purpose. Another potential problem is the complexity of the approach which has not been investigated in detailed. It is clear however, that reasoning in $\mathcal{SHOQ}$(D) is likely to be highly intractable.

Applicability to Information Integration P-$\mathcal{SHOQ}$ can be used for expressing all the mappings mentioned in the introduction. The ontologies, however, are not allowed to contain inverse roles. Furthermore, RDF ontologies whose semantics cannot be described solely with the Description Logics paradigm cannot be integrated, because the Logic Programming paradigm which is needed for describing the RDF semantics as well, is not covered by P-$\mathcal{SHOQ}$.

P-CLASSIC

Expressiveness P-CLASSIC is a probabilistic extension of the CLASSIC Description Logics. Different from $\mathcal{SHOQ}$, the CLASSIC description logics is designed for efficiency of reasoning rather that for expressive power. In particular, CLASSIC does only contain conjunction, negation on atomic concepts, universal and number restrictions as well as role fillers. As a result, deciding subsumption in CLASSIC can be computed in polynomial time based on structural comparison of concept expressions. P-CLASSIC extends the language with probabilistic information about properties of typical instances in terms of a Bayesian network. The corresponding network contains random variables indicating the following information.

- Membership in atomic concepts A
- For each Property R
    - A distribution over possible fillers o in expressions of the form P(R(x,o))
    - A distribution over possible ranges C in expressions of the form $(R(x,y) \rightarrow C(y))$ where C is specified in terms of a separate Bayesian network.
    - A distribution over the number of fillers n in equations of the from $(\exists^{=n} y{:}R(x,y))$

Additionally, the network represents an efficient encoding of the joint probability over these random variables in terms of conditional probabilities between kinds of assertions mentioned above. This means that P-CLASSIC can be used to represent probabilistic information about terminological knowledge. In particular, we can represent probabilistic subsumption relations between atomic concepts that can be used to represent uncertain mappings and the results of learning subsumption relations. The other features of the language can also be used to represent the result of ontology learning especially distributions over property fillers and ranges are useful for this purpose.

Reasoning and Efficiency The basic reasoning service in P-CLASSIC is to compute the probability of a complex concept expression based on the definition of the joint probability distribution over atomic classes and features of relations. The inference algorithm given in (Koller et al., 1997) takes a concept expression and a P-CLASSIC knowledge base as input and returns the probability of the concept expression. This probability is computed by bottom-up construction of a Bayesian network that represents the concept and using it to infer the probability that an arbitrary object is member of this concept expression. This method can be used to implement probabilistic data retrieval by computing the probability of a class description using a Bayesian network that has been initialized with evidence that corresponds to the properties of the individual we want to test. The fact that P-CLASSIC is based on exact probabilities rather than probability intervals means that the probability defines a natural ranking function for answers.

The major advantage of P-CLASSIC is the fact that reasoning is relatively efficient compared to other formalisms. This is due to the fact that both, the logical and probabilistic formalism have been chosen with efficiency in mind. The algorithm for constructing the Bayesian Network of a class description is defined as a direct extension of the structural subsumption algorithm of P-CLASSIC that is known to be polynomial. Additional complexity is added by the need to evaluate the network. This problem is known to have an exponential complexity, but only in the maximal the number of parents of a node. Further, the reuse of results for certain class expressions improve the time needed for actually compute the probability. This means that P-CLASSIC has relatively nice properties with respect to the computational complexity.

Applicability to Information Integration When P-CLASSIC was devised, its application in the area of information integration was not intended. Mainly, it was intended to express and reason about the degree of overlap between concepts of an ontology. P-CLASSIC works with probabilistic formalizations of so-called p-classes each of which describes a certain class of individuals. Except of the expressibility of a the probability distribution over the role fillers of a role, the probabilistic expressions formalize concepts. The possibility to express a probability distribution over the role fillers of a role is not enough for the area of information integration. Therefore, this formalism is too restricted for being used in the area of information integration.

Extensions of Logic Programming Formalisms

Several approaches for extending Logic Programming formalisms with probabilities have been proposed. However, most of them have not been designed with the Semantic Web in mind. In the following, we discuss only those probabilistic logic programming approaches that have been designed for the Semantic Web and involve ideas about how to connect rule bases with ontologies represented in OWL or related formalisms. Two kinds of such approaches can be distinguished. The first kind integrates OWL with Logic Programming by allowing to specify a logic program and a description logics knowledge base at the same time and allowing them to interact in some way. In general, the logic program is used for querying both knowledge bases. For this purpose, the logic program can contain atoms that query the Description Logics knowledge base. We survey two approaches of this kind, (Lukasiewicz, 2005) (and a restricted version thereof by Lukasiewicz (2006)) and (Cali et al, 2008). The other kind of approaches base on a subset OWL and Logic Programming have in common and on a translation from OWL to Logic Programming formalisms that have been extended with probabilities. The subset of OWL and Logic Programming, that these approaches consider is Description Logic Programs (DLP) which is very close to Datalog (Grosof et al., 2003). (Predoiu, 2006; Predoiu & Stuckenschmidt, 2007) translates OWL ontologies that lie in the DLP fragment to a probabilistic Datalog formalism that is close to Bayesian Logic Programs (Kersting & De Raedt, 2001) while (Nottelmann & Fuhr, 2005) translate a slight extension of the DLP fragment, namely DLP with equality, to probabilistic Datalog (Fuhr, 2000).

In the following, we present a short overview on Description Logic Programs: As they are a subset of the Description Logics underlying OWL and the Logic Programming paradigm and thus have a Description Logics and a Logic Programming syntax. In the logic programming syntax, they correspond to pure Datalog without negation, equality and integrity constraints. I.e. as with Datalog, a Description Logic Program consists of facts and rules. Each rule has the form $H \leftarrow B_1, ..., B_n$, where H and the B_i are atomic formulae and $n \geq 1$. An atomic formula consists of a predicate symbol p followed by a bracketed n-tuple of terms ti, $p(t_1, . . . , t_n)$ 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, $B_1 \wedge . . . \wedge B_m$, 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. Semantically, Description Logic Programms in the logic programming syntax do not differ from them having been specified in the description logics syntax. As reasoning is concerned with syntactical manipulations, however, Description Logic Programs in the logic programming syntax are restricted to fact-form inference with logic programming reasoners, i.e. only facts can be derived and no axioms like with description logics reasoners that reason with the description logics syntax of Description Logic Programs.

In the following we compare two formalisms that are based on the Description Logic Programming fragment and 2 formalisms that are more expressive.

Bayesian Description Logic Programs

In (Predoiu, 2006), Description Logic Programs have been embedded into the Bayesian Logic Programming formalism (Kersting & De Raedt, 2001). In this approach, the probabilistic extension has the purpose of information integration and has been proposed in order to represent uncertain mappings between ontologies and rules. Also, a means to reason with the mappings and the ontologies and rules having been mapped in an integrated way has been proposed.

Expressiveness Bayesian Description Logic Programs (BDLPs) are a probabilistic extension of the logic programming syntax (and semantics) of Description Logic Programs (Grosof et al., 2003). In Bayesian Description Logic Programs, facts are attached with an apriori probability and rules are attached with a conditional probability where the states of the head atom are conditioned on the states of the body atoms. Like a Bayesian Logic Program, a Bayesian Description Logic Program encodes a Bayesian Network.

Reasoning and Efficiency The basic reasoning task associated with Bayesian Description Programs is querying for the probability density of a conjunction of ground atoms given a conjunction of ground evidence atoms. In (Predoiu & Stuckenschmidt, 2007), the semantics has been extended to allow non-ground query atoms in order to enable information retrieval by deriving all ground atoms that satisfy the query and rank them by means of their probabilities. There are no complexity results known yet for Bayesian Description Logic Programs and no inference engine is available yet. However, the inference engine for Bayesian Logic Programs, Balios (Kersting & Dick, 2004) which calls Sicstus Prolog for deriving the least Herbrand Model, can be used for reasoning with Bayesian Description Logic programs as well, because Bayesian Description Logic Programs are a subset of Bayesian Logic Programs.

Applicability for Information Integration Bayesian Description Logic Programs have been devised in order to enable Information Integration and they are able to cover all representational issues mentioned in the introduction. However, the ontologies to be mapped are restricted to the Description Logic Programming fragment and this is often a too severe expressivity restriction.

pOWL Lite$^-$ and pOWL Lite^{EQ}

(Nottelmann & Fuhr, 2005) have presented probabilistic extensions of two OWL Lite subsets. One of these subsets corresponds to Description Logic Programs and the other one to Description Logic Programs with equality. The probabilistic extensions are both based on probabilistic Datalog (c.f. the section on probabilistic models above in this chapter). OWL formulae that can be translated to Datalog can each be provided with probabilities and processed afterwards by a pDatalog system.

Expressiveness As mentioned above, two OWL Lite subsets have been extended with probabilities. One corresponds to Description Logic Programs, its probabilistic extension being called pOWL Lite$^-$.[3] The other one corresponds to Description Logic Programs extended with equality, its probabilistic extension being called pOWL Lite^{EQ}. A translation of OWL formulae in the Description Logic Programming fragment (possibly with equality) into the Logic Programming syntax is provided and these can be attached with probabilities in the way that pDatalog allows. These probabilistic Datalog rules are processed afterwards by a pDatalog system.
Possible pOWL Lite$^-$ expressions are listed below. Note that α ($\alpha \in [0, 1]$) which is written in front of each uncertain expression is the probability for the complete expression which is written behind it.

- **Class membership axioms:** α C(a) with $\alpha \in [0, 1]$
  This expression corresponds to the statement that a is an instance of class C with probability α
- **Complex class membership assertions:** α C(y) $\leftarrow$ R(a, y)
- **Role assertions:** α R(a, b)
- **class inclusions:** α_1 B_1(x) $\leftarrow$ A(x). and **…** and α_n B_n(x) $\leftarrow$ A(x). with $n \geq 1$. This expression corresponds to the OWL expression Class(A partial B_1 … B_n) and its probabilistic extension allows to express for each B_i a certainty with which A is a subclass of B_i.
- **Class inclusions with a restriction:** α B(y) $\leftarrow$ A(x), R(x, y). This expression corresponds to the OWL expression Class(A partial restriction(R allValuesFrom B)) and its probabilistic extension allows to express the probability for A being a subclass of the class of elements that have a relation with elements of B.
- **Role inclusions:** α R(x, y) $\leftarrow$ S(x, y).
- **Symmetric role axioms:** α R(x, y) $\leftarrow$ R(y, x).
- **Transitive role axioms:** α R(x, z) $\leftarrow$ R(x, y), R(y, z).
- **Domain restrictions:** α B(x) $\leftarrow$ R(x, y).
- **Range restrictions:** α B(y) $\leftarrow$ R(x, y).

Additionally, OWL Lite^{EQ} allows the expression of the following axioms:
- **Individual equivalence expressions:** α a = b $\leftarrow$ $\mathfrak{U}$(a), $\mathfrak{U}$(b).
- **Maximal Cardinality of 1 expressions:** α y = z $\leftarrow$ A(x), R(x, y), R(x, z).
- **Functional role axioms:** α y = z $\leftarrow$ R(x, y), R(x, z).
- **Inverse functional role axioms:** α x = y $\leftarrow$ R(x, z), R(y, z).

[3] Note that Description Logic Programs are called OWL Lite$^-$ in (Nottelmann & Fuhr, 2005). This is the reason for calling its probabilistic extension pOWL Lite$^-$.

$\mathfrak{U}$ is a predicate which contains all individuals that are available in the pOWL Lite^{-} or pOWL Lite^{EQ} knowledge base.
Additionally, in order to deal with pOWL Lite^{-}/pOWL Lite^{EQ} more easily, a language for stating probabilistic horn rules basing on the SWRL syntax has been added. For the purpose of reasoning, however, this language is translated to pDatalog as well. Clearly, with this addition, the expressivity goes beyond Description Logic Programs. Although the supported fragment of OWL is not extended, much more of the Logic Programming fragment is covered. It is unclear whether full pDatalog or only a subset is supported.

Reasoning and Efficiency In (Nottelmann & Fuhr, 2005), an implementation, i.e. a wrapper for a pDatalog reasoner like HySpirit, has not been provided. Efficiency for reasoning with pOWLLite^{-} and pOWLLite^{EQ} can be considered promising due to its limited expressivity. However, with the addition of the capability for stating horn rules basing of the SWRL syntax, one might end up with the full expressivity of pDatalog. Then, the general empirical complexity results of pDatalog mentioned in the section "probabilistic languages and models" above in this chapter is carried forward to pOWLLite^{-} and pOWLLite^{EQ} with the addition of probabilistic horn rules in the SWRL syntax.

Applicability for Information Integration This formalism is applicable for information integration and can express all kinds of mappings suggested in the introduction. But again, the restriction of the ontologies to the Description Logic Programming fragment is often too severe. Note that, although the formalism has been additionally equiped with horn rules basing on the SWRL syntax, the integration with the translation of the OWL ontologies in the DLP fragment has not been formalized explictly and thus cannot be considered concerning the expressivity of the ontologies.

Probabilistic Description Logic programs with special DL-atoms

In (Lukasiewicz, 2005) and (Lukasiewicz, 2006), probabilistic description logic programs[4] (pdl programs) are presented that base on a loose query-based coupling of a Logic Program and a Description Logic knowledge base. The non-probabilistic formalism that pdl programs are based on has been published in (Eiter et al., 2004) as a combination of answer set programming with Description Logics. This non-probabilistic formalism has been combined with independent choice logic yielding a probabilistic extension of the base formalism.

Expressiveness By means of the non-probabilistic base logic, a knowledge base KB = (*L*, *P*) can be specified. *L* corresponds to a classical $\mathcal{SHIF}$(D) or $\mathcal{SHOIN}$(D) knowledge base and *P* corresponds to a Logic Program which may contain queries to *L*. While *L* can be specified in the typical Description Logics syntax and has the typical Description Logics semantics, the Logic Program consists of a finite set of rules of the form

$$a \leftarrow b_1, ..., b_k, \text{not } b_{k+1}, ..., \text{not } b_m \quad \text{with } m \geq k \geq 0.$$

Here, a and the b_i are atomic formulae. An atomic formula consists of a predicate symbol *p* followed by a bracketed n-tuple of terms ti, $p(t_1, . . . , t_n)$ with $n \geq i \geq 0$. A term can be either a

[4] Note that although the formalism is called description logic programs like the formalism in (Grosof et al., 2003), it is a completely different language as it goes beyond the common subset of Description Logics and Logic Programming. In order to hint the difference, we are using lower case letters for this formalism while we call the formalism from Grosof et al. (2003) Description Logic Programs.

constant (i.e. an instance) or a variable (i.e. a placeholder for an instance). Two kinds of negated atoms are distinguished: classically negated atoms $\neg a$ and default-negated atoms *not a*. Furthermore, there are special kinds of atoms called dl-atoms that are allowed to be one of the $_{bi}$ with k ≥ i. I.e. the dl-atoms are only allowed to occur in the positive, unnegated part of the body . Such dl-atoms form a query to *L* with additional constraints that extend or shrink the instance set associated with concepts and roles occuring in L. The logic program *P* has been given a well-founded and an answer-set semantics in (Eiter et. al, 2004).

Basing on this formalism, in (Lukasiewicz, 2005) and (Lukasiewicz, 2006), a probabilistic extension has been proposed that combines this formalism with independent choice logic. A probabilistic description logic program is a knowledge base KB = (*L*, *P*, *C*, μ) where

- (*L*, *P*) is a dl program as explained above. Note that in (Lukasiewicz, 2005), a well-founded and an answer-set semantics have been defined for *P*.
- *C* is a choice space that corresponds to a set of sets whose union is a subset of the Herbrand Base HB_P of *P*. Alternatives, atomic choices and total choices are defined analogously to independent choice logic (c.f. the section "probabilistic languages and models" above in this chapter). No atomic choice is allowed to occur in the head of rule in *P*, but in anywhere in the body.
- μ is a probability distribution on the choice space C, i.e. $\mu: \cup C \rightarrow [0, 1]$ such that $\Sigma_{a \in A}\, \mu(a) = 1$ for all alternatives $A \in C$ and $\mu(B) = \Pi_{b \in B}\, \mu(b)$ for all total choices B of C. Note that the probability of total choices imposes probabilistic independence between the alternatives of C or, differently worded, the random variables specified by C.

Reasoning and Efficiency Probabilistic queries to a pdlp knowledge base as specified above can be either atomic or complex:

- an atomic probabilistic query queries for the probability of a formula ψ given another formula ϕ: $(\psi \mid \phi)[l, u]$. Here, l, u are placeholders for reals in the interval [0, 1] and stand for the lower bound and the upper bound of the probability. Formulas can be arbitrary contain of negation and conjunction.
- (complex) probabilistic queries F are inductively defined as follows: each atomic probabilistic query A (with l, u being instatiated, however) is a probabilistic query. If G and H are probabilistic queries, then so are $\neg G$ and $G \wedge H$.

The correct answer to a complex probabilistic query F is defined to be the set of all substitutions θ such that $F\theta$ is a consequence of the knowledge base. With the answer set semantics, it is distinguished between answer set consequences and tight answer set consequenes. For answer set consequences, every model of the knowledge base has to be a model of $F\theta$ as well. For tight answer set consequences, furthermore, *l* (resp. *u*) have to be the infimum (resp. supremum) of $Pr(\psi\theta \mid \phi\theta)$ subject to all models of KB given that $Pr(\phi\theta) > 0$.

With the well-founded semantics, $F\theta$ is a consequence of KB if $F\theta$ is true in the well-founded model. Again, a query $(\psi \mid \phi)[l, u]\theta$ is a tight well-founded anwer, is *l* (resp. *u*) are the infimum (resp. supremum) of $Pr(\psi\theta \mid \phi\theta)$ given that $pr(\phi\theta) > 0$. Note that $Pr(\psi\theta \mid \phi\theta)$ is a probabilistic interpretation either under the answer-set semantics or under the well-founded semantics as defined in (Lukasiewicz, 2005), depending on the context. More specifically, Pr is a probabilistic distribution over all models.

The computation of tight answers to queries $(\psi \mid \phi)[L, U]\theta$ under the answer-set semantics involves classical logical deduction (according to the semantics used) and solving two linear optimization problems. The complexity of solving these linear optimization problems has not been discussed, yet. However, deduction under the answer set semantics has a very high complexity. More specfically, for *L* being a $\mathcal{SHIF}$(D) knowledge base (resp. a $\mathcal{SHOIN}$(D) knowledge base) query answering is in the complexity class co-NEXP (resp. co-NP^{NEXP}) (Eiter et. al, 2004). Query answering under the well-founded semantics is for L being a $\mathcal{SHIF}$(D) knowledge base (resp. $\mathcal{SHOIN}$(D) knowledge base) complete for EXP (resp. P^{NEXP}) (Eiter et. al, 2004). In (Lukasiewicz, 2006), for the same syntax as shown above for both, knowledge bases and queries, a stratified semantics based on a (local) stratification of the knowledge base has been defined. Complexity for this semantics has not been considered at all. However, query answering in stratified logic programs in general, i.e. without integrating Description Logic knowledge bases, has a much lower complexity than in those that go beyond stratification and lie in the well-founded semantics, but is still intractable in the worst case.

Applicability to Information Integration This formalism is the first one mentioned in this chapter that is able to fully integrate full OWL and a huge part of RDF. Concerning the expressivity, this formalism is therefore very suitable for the representation of OWL (i.e. the OWL-Lite and OWL-DL fragments) and a huge part of RDF in the same syntax. However, as dl-atoms are not allowed to occur in the head of the rules, only a Logic Program can be the target of a mapping. Therefore, it cannot be used for information integration on the Semantic Web where OWL ontologies can be the target of mappings.

Probabilistic Disjunctive Description Logic Programs

In (Cali et al., 2008), a tighter integration of Logic Programs and the Description Logics underlying OWL has been combined with independent choice logic. This approach is called probabilistic disjunctive description logic programs (pddl programs) and differs from the formalism mentioned above in the fact that there are no special dl-atoms necessary for the flow of information between *L* and *P*. In fact, concepts and roles of L can occur as unary or binary predicates in P as well. Furthermore, the logic programming component *P* is allowed to have rules with disjunction in the head while with probabilistic description logic programs with special DL-atoms mentioned above, *P* was only allowed to consist of rules with a single, positive atom in the head[5]. Note also that classical negation is not allowed to occur in probabilistic disjunctive description logic programs in contrast to probabilistic description logic programs with special dl-atoms described above.

Expressiveness As before, in the section above, a non-probabilistic base logic is combined with independent choice logic yielding probabilistic disjunctive description logic programs. The non-probabilistic logic used is disjunctive description logic programs (Lukasiewicz, 2007). It allows to specify a knowledge base KB = (*L*, *P*) with *L* being either a $\mathcal{SHIQ}$(D) or a $\mathcal{SHOIN}$(D) knowledge base and *P* being a logic program. *P* is a finite set of disjunctive rules of the form

$$\alpha_1 \vee \ldots \vee \alpha_k \leftarrow \beta_1, \ldots, \beta_n, \text{not } \beta_{n+1}, \ldots, \text{not } \beta_{n+m}$$

with $\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_{n+m}$ being atoms built with the predicate, role and concept symbols of *P* an *L* in the usual way. The logic program *P* has been given an answer set semantics in (Lukasiewicz, 2007).

---

[5] Note that conjunction in the head is allowed with probabilistic description logic programs with special DL-atoms as well, because rules with conjunction in the head can be split to regular horn rules by means of the Lloyd-Topor-Transformation (Lloyd & Topor, 1984).

Basing on this formalism, in (Cali et. al, 2008), a probabilistic extension has been proposed that combines this formalism with independent choice logic. A pddl program is a knowledge base KB = (L, P, C, μ) where

- (L, P) is a ddl program as explained above
- C is a choice space that corresponds to a set of sets whose union of its elements A $\in$ C corresponds to a subset of the set $HB_P \backslash DL_P$. Here, HB_P is the herbrand base of P and DL_P is the subset of the herbrand base of P that is built with predicates that occur in L as concepts or roles, too. Alternatives, atomic choices and total choices are defined analogously to independent choice logic (c.f. the section "probabilistic languages and models" above in this chapter).
- μ is a probability distribution on the choice space C as defined in the section above.

Reasoning and Efficiency A probabilistic query to a pddl knowledge base has the form $\exists\ (c_1(\mathbf{x}) \vee \ldots \vee c_n(\mathbf{x}))[r, s]$ where $\mathbf{x}$, r, s are tuples of variables, $n \geq 1$, and each $c_i(\mathbf{x})$ is a conjunction of atoms constructed from predicate and constant symbols in P and variables in $\mathbf{x}$. Similarly to probabilistic description logic programs with special dl-atoms, it is distinguished between correct and tight answers to such a query. Given a probabilistic query $\exists\ (q(\mathbf{x}))[r, s]$, a formula $(q(\mathbf{x}))[l, u]$ with $l, u \in [0, 1]$ is a *correct consequence* of the knowledge base iff the probability of it lies always in the interval [0, 1] for every answer set of KB and every variable assignment σ. A formula $(q(\mathbf{x}))[l, u]$ with $l, u \in [0, 1]$ is a *tight consequence* of the knowledge base iff l (resp. u) is the infimum (resp. supremum) of the probability of the formula subject to all answer sets of the knowledge base and all variable assignments σ.
The consistency and the query processing problem are decidable in pddl programs. For a pddl knowledge base KB = (L, P, C, μ) with L being either a $\mathcal{SHIF}$(D) or a $\mathcal{SHOIN}$(D) knowledge base, deciding whether KB is consistent is complete for $NEXP^{NP}$ given that the size of C is bounded by a constant. For a pddl knowledge base KB = (L, P, C, μ) with L being either a $\mathcal{SHIF}$(D) or $\mathcal{SHOIN}$(D) knowledge base, deciding whether (q)[l, u] with q being a ground atom from HB_P and $l,u \in [0, 1]$ is a consequence of KB is complete for co-$NEXP^{NP}$.
In (Cali et al, 2008), a subset of pddl knowledge bases with strictly limited expressivity has been presented which allows for deciding consistency and query processing in polynomial time. However, for this purpose, the Description Logics part L must be in DL-Lite (Calvanese et. al, 2005) and the logic programming part P extended with additional rules modelling basic inclusion in L must be normal, i.e. only one non-negated atom in the head is allowed, and locally stratified.

Applicability to Information Integration This formalism is capable of representing full OWL (i.e. full OWL-Lite and OWL-DL ontologies) and a huge part of RDF in the same syntax and is therefore capable for integrated query answering and reasoning with both formalisms. Furthermore, as predicates representing concepts and roles in the ontology can occur freely in the rule, i.e. also in the head, mappings can be represented with the formalism straightforwardly. Furthermore, as disjunction in the head is allowed, inconstistent mappings can be dealt with more easily that with pure horn rules that allow only one atom in the head of a rule. The representation of mappings with this formalism has been investigated and described in detail by Cali & Lukasiewicz (2007).

DISCUSSION AND CONCLUSIONS

We conclude the chapter with a discussion of the benefits and drawbacks of the different approaches for extending Semantic Web languages with probabilistic information that we have surveyed above. It turns out that there exist *two different kinds of probabilistic extensions*. The first kind of extensions is a rather loose coupling between an existing Semantic Web language and a probabilistic model. There, the Semantic Web Language is just used syntactically as a vocabulary for exchanging knowledge bases specified in the probabilistic model. The second kind of extensions provides a tight integration on the formal level between a Semantic Web Language or a subset of it and a probabilistic model. The second kind of extensions encompasses as well the formalisms that integrate a Semantic Web language with logic programming and combine the resulting formalisms with a probabilistic model. These extensions provide also a tight formal integration of a Semantic Web language which usually is OWL-Lite/OWL-DL or the Description Logic which underlies these OWL fragments with a logic programming formalism and a probabilistic model.

Extensions of the first kind that are mentioned in this survey are the approaches of

- (Fukushige, 2005) which proposes a vocabulary for encoding Bayesian Netwoks with RDF,
- (Yang & Calmet, 2006) which proposes a vocabulary for encoding Bayesian Networks with OWL and
- (Costa & Laskey, 2006) which proposes a vocabulary for encoding Multi-Entity Bayesian Networks with OWL.

These approaches are rather unsatifying because they do not consider the semantics of Semantic Web languages but rather focus at a special kind of probabilistic model, i.e. Bayesian Networks or Multi-Entity Bayesian Networks, and provide a Semantic Web based syntactical interchange format for these probabilistic models and their semantics. By means of these approaches uncertainty can only be represented on the Semantic Web but no Semantic Web statement is extended by some kind of uncertainty. Thus, from the five areas mentioned in the introduction where a consideration of uncertainty is needed on the Semantic Web, only the needs of the first area are met. I.e. only the requirements for representing statistical information are met. The area of the Semantic Web itself does not benefit substantially from these extensions. It is even arguable whether the probabilistic models represented benefit from using a vocabulary basing on a Semantic Web language without any formal integration. Note that currently no reasoning support for these vocabularies has been implemented yet, i.e. no wrappers exist that is able to parse the Semantic Web language vocabulary defined for the particular probabilistic models and feed it to a reasoner that is capable to deal with them. However, for PR-OWL, a reasoner implementation effort has recently been started.

Extensions of the second kind naturally fullfil the requirements for representing statistical information. Additionally, because of the much tighter integration on the formal level, they are also much more appropriate for Ontology matching and aligning and also for ontology learning by means of bayesian machine learning methods. The same holds for ontology population or document classification, respectively. E.g. (Straccia & Troncy, 2006) have proposed methods for learning probabilistic mappings between OWL ontologies that are represented as very simple pDatalog rules. These methods have been implemented in the oMAP framework. The pDatalog rules that can be learned in the oMAP framework are contained in pOWLLite$^-$ as well. Thus, those mappings are very much related to POWLLite$^-$ and pOWLLite^{EQ}. Probabilistic disjunctive description logic programming as described above has also been proposed for usage in the area of the usage of ontology mappings and information integration. These considerations have been theoretical and no implementation has been provided, yet, but is considered as future work. In (Predoiu, 2006), Bayesian Description Logic Programs have been proposed solely for the representation of mappings and the uncertainty inherently associated with any automatically discovered mapping. An implementation, however, is not yet provided, but under development. The only further formalism for which a mapping scenario has been considered is BayesOWL. As each BayesOWL ontology corresponds to a Bayesian Network, in the mapping scenario, Bayesian Networks are mapped to each other. Hence, this scenario is computationally very expensive. The formalism which has been identified as being the most appropriate for information integration is probabilistic disjunctive description logic programming because of its expressivity concerning the ontologies to be mapped and the mappings and the possibility to deal with inconsistencies introduced by mappings to a certain extent which needs to be further investigated. For the other probabilistic extensions surveyed in this paper, no mapping scenario has been considered. Most of them have been proposed without the area of ontology mapping and information integration in mind and therefore they all have drawbacks concerning their usage in this area. Furthermore, no research on learning or using mappings has been performed yet in any of the formalisms except of pOWLLite^{-}.

The probabilistic extensions that integrate Semantic Web languages or subsets thereof tightly with a probabilistic model, can be destinguished as follows:

- Extensions that consider not only the semantics but also the syntax of established Semantic Web languages, examples being pRDF and BayesOWL. Both support only a small subset of the languages they extend probabilistically. pRDF extends basically only the three RDF built-in predicates for specifying subclass relations, instance and role membership with probabilities. Furthermore, RDF built-in predicates around properties (the subproperty relation, the definition of the range and the domain of properties) are allowed to be used classically in deterministic triples. BayesOWL has an even more limited expressivity than pRDF because it does not even allow to express uncertainty of properties and instances.
- Extensions that consider subsets of the Description Logics underlying OWL, examples being P-$\mathcal{SHOQ}$(D) and P-CLASSIC. P-CLASSIC has a rather limited expressivity as it combines the description logic CLASSIC that has been designed for efficiency of reasoning and suffers thus of a limited expressivity with the probabilistic model of Bayesian Networks. CLASSIC is a very small subset of $\mathcal{SHOQ}$(D). For P-CLASSIC no reasoning tools have been devised. P-$\mathcal{SHOQ}$(D) has the full expressivity of $\mathcal{SHOQ}$(D) and is very near to OWL-DL which corresponds to $\mathcal{SHOIN}$(D). The only difference is that inverse roles cannot be specified. However, for P-$\mathcal{SHOQ}$(D) no reasoning tools exist either. Furthermore, the proposed reasoning algorithm can be expected to have a very high complexity because it involves silving a linear equation system.
- Extensions that consider integrations of a Logic Programming variant and a Description Logic underlying OWL. Such extensions are Bayesian Description Logic Programs, pOWLLite^{-} and pOWLLite^{EQ}, probabilistic Description Logic Programs and probabilistic Disjunctive Description Logic Programs. We think that probabilistic extensions of integration formalisms that integrate Description Logics and Logic Programs are very important also because Logic Programming is a very important paradigm especially present in the database area. Furthermore, as shown by the Rule-Interchange-Format working group at the W3C[6] that intends to carry over the Logic

[6] http://www.w3.org/2005/rules

Programming paradigm into the Semantic Web, there is a huge interest in representing rules on the Web. In the next paragraph we will shortly summarize a comparison of the form of integration between DL and LP, the expressivity of the formalisms and the tightness of the combination between the deterministic logical model and the probabilistic model.

Two of the probabilistic approaches that integrate Logic Programming with Description Logics, integrate only a subset of OWL. These approaches are Bayesian Description Logic Programs and pOWLLite^{-}/pOWLLite^{EQ}. Bayesian Description Logic Programs combine pure Description Logic Programs, i.e. Datalog without equality and negation, a common subset that is shared by the Description Logics underlying OWL and the Logic programming paradigm, with Bayesian Logic Programs. The integration of the deterministic and the probabilistic model is very tight and yields even a subset of the probabilistic model. pOWLLite^{-} and pOWLLite^{EQ} are intended to be a probabilistic extension of Description Logic Programs as well (the latter extends them also with equality). Besides a probabilistic extension of Description Logic Programs (possibly extended with equality) also probabilistic Horn rules are supported that increase the expressivity and it is unclear whether the expressivity ends up in full pDatalog. However, as negation is allowed and also equality, pOWLLite$^{-/EQ}$ seems to support a larger expressivity of the deterministic model. The probabilistic models used in Bayesian Description Logic Programs and pOWLLite^{-}/^{EQ} differ as well. Bayesian Logic Programs do not support negation and are a compact representation of a Bayesian Network. pDatalog supports negation under the well-founded semantics and until now no relation to Bayesian Networks has been found.

Differently from Bayesian Logic Programs and pOWLLite^{-}/^{EQ}, probabilistic Description Logic Programs and probabilistic Disjunctive Description Logic Programs support full OWL-Lite and OWL-DL and integrate them with stratified logic programms, logic programs under the well-founded and under the answer set semantics. These approaches have a strong theoretical basis and all of them combine the deterministic model with independent choice logic as probabilistic model. The query language supports differently form Bayesian Logic Programs and pOWLLite^{-}/pOWLLite^{EQ} queries for probabilistic intervals. The query language is very expressive and reasoning is very complex because it involves solving a linear equation system like with P-$\mathcal{SHOQ}$. However, for a restricted subset of probabilistic Disjunctive Description Logic programs, a polynomial complexity has been shown. This subset consists of a Description Logics knowledge base lying in a subset of the Description Logic programming fragment and of a Logic Program that corresponds to Datalog with negation that is locally stratified.

Most of the approaches that probabilistically integrate the Logic Programming paradigm with the Description Logics paradigm, provide own reasoners. For Bayesian Description Logic Programs, the reasoner Balios (Kerstin & Dick, 2004) that has been implemented for its probabilistic model which is a superset of itself can be used. For pOWLLite$^{-/EQ}$, HySpirit or Pire which are reasoners for full pDatalog which is their underlying probabilistic model can be used. In fact, an implementation for pOWLLite$^{-/EQ}$ basing on PIRE exists. For probabilistic Description Logic Programs and probabilistic Disjunctive Description Logic Programs no reasoners exist yet.

FUTURE RESEARCH DIRECTIONS

As overall conclusion, we can summarize that until recently, research has not paid much attention to uncertainty in the area of the Semantic Web. However, it gains more and more interest and new approaches considering uncertainty tend to emerge. Still, many of these approaches are rather half-baked and a lot of things are missing:

- **Reflections on gathering probabilities.** *Where do the probabilities used in the web come from? What kinds of probabilities exist?* Cali & Lukasiewicz (2007) make the first proposal to distinguish between mapping trust, mapping error or plain mapping probabilities. However, we think that this is just a very first step and might be a beginning for new insights into the types and usages of probability usage, depending on the event space and intended semantics. *How can those probabilities be gathered?* (Straccia & Troncy, 2006) make proposals for learning very simple pDatalog rules. Investigations of methods for learning more complex structures of different probabilistic models would enable the Semantic Web community to anticipate in which forms a Semantic Web where automatic information integration would be possible.
- **Reflections on which probabilistic models are suitable for which subareas of the Semantic Web.** I.e. investigations of the applicabilty and usefulness of probabilistic extensions of Semantic Web languages in the different areas that need to consider uncertainty have to be done. E.g. it has to be seen whether a probabilistic Logic Programming approach is better suited for discovering and representing mappings than a purely probabilistic Description Logic one when only OWL ontologies and no rules are involved. This requirement is interweaved with the requirement above because the investigations on the different kinds of probabilities might lead to usefulness results. Furthermore, investigations on methods for learning those different probabilistic Semantic Web extensions, might naturally lead to further insights of the usability of the different formalisms in the different areas by means of complexity results and learnability results.
- **Reflections on cyclic probabilistic representations:** None of the above mentioned probabilistic extensions of Semantic Web languages can deal with cyclic representations. We deem this as a severe drawback because of the open and free nature of the Semantic Web. If ontologies, logic programs and mappings between them are considered as a whole, cyclic descriptions are very likely to occur and are not avoidable. Only in small toy worlds, cycles can be avoided. It has to be investigated in which ways cyclic probabilistic respresentations can be dealt with.
- **Reasoning methods and implementations:** Reasoning tools in general are not provided for the languages themselves, only for related logical formalisms which can used by means of wrappers but are not optimized for the languages at hand. If there are reasoning tools that are specialized for the languages themselves, then they support only a part of the language like in the case of pRDF. Research needs to focus on the development of optimized reasoning methods and reasoning tools need to be implemented in order to enable the usage of uncertain statements in the Semantic Web and in order to make reasoning feasible facing the huge amount of ontologies and data that can be expected to be present in the future of the Semantic Web. For example research on approximate and distributed reasoning would enable feasible query answering with large-scale knowledge bases and instance bases like imposed by the Semantic Web. None of the approaches above employ or consider currently any form of approximate or distributed reasoning.

REFERENCES

Angele, J., Boley, H., de Bruijn, J., Fensel, D., Hitzler, P., Kifer, M., Krummenacher, R., Lausen, H., Polleres, A. & Studer, R. (2005). *Web Rule language (WRL).* W3C Member Submission, from http://www.w3.org/Submission/WRL/

Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F. & Stein, L. A. (2004). *OWL Web Ontology Language Reference*. W3C Recommendation, from http://www.w3.org/TR/2004/REC-owl-ref-20040210

Cali, A., Lukasiewicz, T., Predoiu, L. & Stuckenschmidt, H. (2008). *Tightly Integrated Probabilistic Description Logic Programs for Representing Ontology Mappings.*. In Proceedings of the International Symposium on Foundations of Information and Knowledge Systems (FOIKS).

Cali, A. & Lukasiewicz, T. (2007). *Tightly Integrated Probabilistic Description Logic Programs*. Technical Report, Institut für Informationssysteme, TU Wien

Calvanese, D., de Giacomo, G., Lembo, D., Lenzerini, M., & Rosati, R. (2005). *DL-Lite: Tractable description logics for ontologies*. In Proceedings 20th AAAI conference on Artificial Intelligence.

Castillo, E., Gutierrez, J. M. & Hadi. A. S. (1997). *Expert systems and probabilistic network models*. Springer-Verlag

Costa, P. C. G. & Laskey, K. B. (2006). *PR-OWL: A Framework for Probabilistic Ontologies*. In Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS).

Cowell, R. G., Dawid, A. P., Lauritzen, S. L. & Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems*. Springer-Verlag.

de Bruijn & J., Heymans, S. (2007). *Logical Foundations of (e)RDF(S): Complexity and Reasoning*. In Proceedings of the International Semantic Web Conference (ISWC).

De Raedt, L., Kimmig, A. & Toivonen, H. (2007). *ProbLog: A Probabilistic Prolog and Its Application in Link Discovery*. In Proceedings of the 20th International Joint Conference on Artificial Intelligence.

Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., et al. (2000). *The Semantic Web: The roles of XML and RDF*. IEEE Internet Computing, 4(5), 63-67.

Ding, L., Kolari, P., Ding, Z. & Avancha, S. (2006). *BayesOWL: Uncertainty Modeling in Semantic Web Ontologies*, Soft Computing in Ontologies and Semantic Web, Springer Verlag..

Doan, A., Madhavan, J., Domingos, P. & Halevy, A. (2003). *Ontology Matching: A Machine Learning Approach.* Handbook on Ontologies in Information Systems (pp 397-416), Springer-Verlag.

Eiter, T., Lukasiewicz, T., Schindlauer, R. & Tompits, H. (2004). *Combining answer set programming with description logics for the Semantic Web*. In Proceedings of the 9th international conference on the Principles of Knowledge Representation and Reasoning (KR-2004).

Euzenat, J. & Shvaiko, P. (2007). *Ontology Matching*. Springer Verlag.

Fagin, R. (1999). *Combining fuzzy information from multiple systems*. Journal of Computer and Systems Sciences, 58:83-99.

Fikes, R. & McGuinness, D. (2001). *An Axiomatic Semantics for RDF, RDF-S and DAML+OIL*. W3C Note, from http://www.w3.org/TR/daml+oil-axioms

Fuhr, N. (2000). *Probabilistic Datalog: Implementing Logical Information Retrieval for Advanced Applications*. Journal of the American Society for Information Science, 51(2): 95-110.

Fukushige, Y. (2005). *Representing Probabilistic Knowledge in the Semantic Web,* from http://www.w3.org/2004/09/13-Yoshio/PositionPaper.html

Fukushige, Y. (2005). *Representing Probabilistic Relations in RDF*. In Proceedings of Workshop on Uncertainty Reasoning for the Semantic Web (URSW).

Giugno, R. & Lukasiewicz, T. (2002). *P-$\mathcal{SHOQ}$(D): A Probabilistic Extension of $\mathcal{SHOQ}$(D) for Probabilistic Ontologies in the Semantic Web*. In Proceedings Logics in Artificial Intelligence, European Conference, JELIA.

Grosof, B., Horrocks, I., Volz, R. & Decker, S. (2003). *Description Logic Programs: Combining Logic Programs with Description Logic*. In Proceedings of 12th International Conference on the World Wide Web.

Heinsohn, J. (1991). *A Hybrid Approach for Modeling Uncertainty in Terminological Logics*. In Proceedings of the European Conference on Symbolic and Qualitative Approaches to Reasoning with Uncertainty.

Holi, M. & Hyvönen, E. (2006). *Modeling Uncertainty in Semantic Web Taxonomies*. In Z. Ma (Ed.), Soft Computing in Ontologies and Semantic Web, Springer-Verlag.

Horrocks, I. (2005). *OWL Rules, OK ?* In Proceedings of W3C Workshop on Rule Languages for Interoperability.

Horrocks, I. & Patel-Schneider, P. F. (2004). *Reducing OWL entailment to description logic satisfiability*. Journal of Web Semantics, 1(4):345-357.

Horrocks, I. & Patel-Schneider, P. F., van Harmelen, F. (2003). *From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language.* Journal of Web Semantics, 1(1):7-26.

Horrocks, I., Patel-Schneider, P. F., Bechhofer, S. & Tsarkov, D. (2005). *OWL rules: A proposal and prototype implementation.* Journal of Web Semantics, 3(1):23-40.

Hustadt, U., Motik, B. & Sattler, U. (2005). *Data complexity of reasoning in very expressive description logics*. In Proceedings of the 19th International Joint Conference on Artificial Intelligence.

Jaeger, M. (1994). *Probabilistic Reasoning in Terminological Logics*. In Proceedings of the 4th international Conference on Principles of Knowledge Representation and Reasoning.

Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. Springer-Verlag.

Kersting, K. & De Raedt, L. (2001). *Bayesian Logic Programs*. Technical Report No. 151, Institute for Computer Science, University of Freiburg, Germany.

Kersting, K. & Dick, U. (2004). *Balios – The Engine for Bayesian Logic Programs*. In Proceedings of Knowledge Discovery in Databases (PKDD).

Koller, D., Levy, A. & Pfeffer, A. (1997). *P-CLASSIC: A tractable probabilistic description logic*. In Proceedings of the 14th AAAI Conference on Artificial Intelligence (AAAI-97).

Laskey, K. B. (2006). *MEBN: A Logic for Open-World Probabilistic Reasoning.* Technical Report C4I06-01, George Mason University, USA.

Laskey, K. B & Costa, P. C. G. (2005). *Of Klingons and Starships: Bayesian Logic for the 23rd Century*. In Proceedings of the 21st Conference of Uncertainty in AI (UAI).

Lassila, O. & Swick, R. (1999). *Resource Description Framework (RDF) Model and Syntax Specification*. W3C Recommendation, from http://www.w3.org/TR/REC-rdf-syntax/

Lloyd, J. W. & Topor, R. W. (1984). *Making Prolog more Expressive*. Journal of Logic Programming, 3:225-240.

Lukasiewicz, T. (2005a). *Probabilistic Description Logic Programs*. In Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty.

Lukasiewicz, T. (2005b). *Stratified Probabilistic Description Logic Programs*. In Proceedings of the ISWC-2005 Workshop on Uncertainty Reasoning for the Semantic Web (URSW).

Lukasiewicz, T. (2007). *A Novel Combination of Answer Set Programming with Description Logics for the Semantic Web* In Proceedings of the 4th European Semantic Web Conference (ESWC 2007).

Maedche, A. & Staab, S. (2004). *Ontology Learning*. In Staab, S. & Studer, R. (Eds.) *Handbook on Ontologies*. Springer 2004.

Manola, F. & Miller, E. (2004). *RDF Primer*. W3C Recommendation, from http://www.w3.org/TR/rdf-primer/

Nottelmann, H. (2005). *Inside PIRE: An extensible, open-source IR engine based on probabilistic logics.* Technical Report, University of Duisburg-Essen, Germany.

Nottelmann, H. & Fuhr, N. (2006). *Adding Probabilities and Rules to OWL Lite Subsets based on Probabilistic Datalog.* In International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 14(1):17-41.

Pan, R., Ding, Z., Yu, Y. & Pen, Y. A. (2005). *A Bayesian Network Approach to Ontology Mapping*, In Proceedings of the Fourth International Semantic Web Conference (ISWC).

Poole, D. (1997). *The independent choice logic for modelling multiple agents under uncertainty*. Artificial Intelligence 94(1-2):7-56.

Predoiu, L. (2006). *Information Integration with Bayesian Description Logic Programs*. In 3rd IIWeb Interdisciplinary Workshop for Information Integration on the Web.

Predoiu, L. & Stuckenschmidt, H. (2007). *A probabilistic Framework for Information Integration and Retrieval on the Semantic Web*. In Proceedings of the 3rd International Workshop on Database interoperability (InderDB).

Roelleke, T., Lübeck, R., Kazai, G. (2001). *The HySpirit Retrieval Platform.* In Proc. of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR.

Sebastiani, F. (2002). *Machine learning in automated text categorization.* ACM Computing Surveys, 34(1):1-47.

Straccia, U. & Troncy, R. (2006). *Towards Distributed Information Retrieval in the Semantic Web: Query Reformulation Using the oMAP Framework.* In Proceedings of the 3rd European Semantic Web Conference (ESWC).

Udrea, O., Subrahmanian, V. S., & Maijkic, Z. (2006). *Probabilistic RDF*. In Proceedings of the Conference on Information reuse and integration.

Yang, Y. & Calmet, J. (2006). *OntoBayes: An Ontology-Driven Uncertainty Model.* In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06).

Yelland, P.M. (2000). *An Alternative Combination of Bayesian Networks and Description Logics*. In Proceedings of of the 7th international Conference on Knowledge Representation (KR).

ADDITIONAL READINGS

General Logic

Enderton, H. B. (2002). *A mathematical Introduction to Logic*, 2nd edition, Academic Press.

Schoening, U. (1994). *Logic for Computer Scientists*. Birkhaeuser Verlag

Ebbinghaus, H.-D., Flum, J. & Thomas, W. (2007) *Mathematical Logic*. Springer-Verlag

Ben-Ari, M. (2001). *Mathematical Logic for Computer Scientists*. Springer-Verlag.

Logic Programming

Lloyd, J. W. (1987). *Foundations of Logic Programming*. Springer-Verlag.

Ullman, J. D. (1988). *Principles of Database and Knowledge-Base Systems, Volume I*. Computer Science Press.

Ullman, J. D. (1989). *Principles of Database and Knowledge-Base Systems, Volume II*. Computer Science Press.

Abiteboul, S., Hull, R. & Vianu, V. (1995) *Foundations of Databases: The Logical Level*. Addison Wesley.

Zaniolo, C., Ceri, S., Faloutsos, C., Snodgrass, R. T., Subrahmanian, V. S. & Zicari, R. (1997). *Advanced Database Systems*. Morgan Kaufmann.

Baral, C. (2003). *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press.

General Probability Theory

Georgii, H.-O. (2008). *Stochastics*. de Gruyter Verlag

Ross, S. M. (2005). *A first course in Probability*. Prentice Hall.

Bayesian Networks and Graphical Models

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Castillo, E., Gutierrez, J. M. & Hadi. A. S. (1997). *Expert systems and probabilistic network models*. Springer-Verlag

Cowell, R. G., Dawid, A. P., Lauritzen, S. L. & Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems*. Springer-Verlag.

Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. Springer-Verlag.

Edwards, D. (2000). "Introduction to Graphical Modelling", 2nd ed. Springer-Verlag.

Bayesian Logic Programming

Kersting, K. & De Raedt, L. (2001). *Bayesian Logic Programs*. Technical Report No. 151, Institute for Computer Science, University of Freiburg, Germany.

Kersting, K. & De Raedt, L. (2007). *Bayesian Logic Programs: Theory and Tool*. In Getoor, L. & Taskar, B. (Ed.), Introduction to Statistical Relational Learning. MIT Press.

Kersting, K. (2006). *An Inductive Logic Programming Approach to Statistical Relational Learning*. Frontiers in Artificial Intelligence, Volume 148, IOS Press, Amsterdam, The Netherlands.

Independent Choice Logic

Poole, D. (1997). *The independent choice logic for modelling multiple agents under uncertainty*. Artificial Intelligence 94(1-2):7-56.

Multi-Entity Bayesian Networks

Laskey, K. B. (2006). *MEBN: A Logic for Open-World Probabilistic Reasoning.* Technical Report C4I06-01, George Mason University, USA.

Laskey, K. B & Costa, P. C. G. (2005). *Of Klingons and Starships: Bayesian Logic for the 23rd Century*. In Proceedings of the 21st Conference of Uncertainty in AI (UAI).

Probabilistic Datalog

Fuhr, N. (2000). *Probabilistic Datalog: Implementing Logical Information Retrieval for Advanced Applications*. Journal of the American Society for Information Science, 51(2): 95-110.

RDF and RDF Schema

A selection of documents on RDF and RDF Schema (Specification, Use Cases, Recommended Readings, Tools, Related Technologies, etc.) can be found at this url: http://www.w3.org/RDF/

de Bruijn & J., Heymans, S. (2007). *Logical Foundations of (e)RDF(S): Complexity and Reasoning*. In Proceedings of the International Semantic Web Conference (ISWC).

OWL

Smith, M. K., Welty, C. & McGuiness, D. L. (2004). *OWL Web Ontology Language Guide*. W3C Recommendation, from http://www.w3.org/TR/2004/REC-owl-ref-20040210/

Horrocks, I. & Patel-Schneider, P. F., van Harmelen, F. (2003). *From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language*. Journal of Web Semantics, 1(1):7-26.