CASAM:

Computer-Aided Semantic Annotation of Multimedia

European Seventh Framework Programme

Project Acronym: **CASAM**

Project Title: **Computer-Aided Semantic Annotation of Multimedia**

Contract Number: **FP7-217061**

Deliverable Number: **D3.1**

Title of the Deliverable: **Formalisms Supporting First-order Probabilistic Structures**

Task/WP related to the Deliverable: **WP3**

Type: **Deliverable**

Distribution:

Author(s): **TUHH**

Partner(s) Contributing: **TUHH**

Versioning and Contribution History			
Version	**Date**	**Modification reason**	**Modified by**
01	09/25/08	First Completed Version	TUHH
02	10/07/08	Final Version	TUHH
03	10/15/08	Final Version After Internal Review	TUHH

Executive Summary

This report describes major approaches for first-order probabilistic knowledge representation and reasoning. It focuses on approaches with an underlying formal semantics. We analyse the pros and cons of each approach and derive requirements for a representational formalism to be developed in order to meet the knowledge representation and reasoning issues in CASAM.

Contents

1 Introduction

Uncertainty plays a central role in many applications. For instance, classification tasks in general require the systematic management of evidence leading to certain conclusions. Classic fields where this is important are medical and technical diagnosis, forensic science, as well as criminology. We argue that the same systematic uncertainty management is also relevant for media classification or interpretation tasks. The goal of this report is first to provide an overview about the state of the art in the development of expressive modeling formalisms that can appropriately deal with uncertainty, and second, to derive reasoning capabilities that are suited to the requirements for image interpretation as met in the project CASAM (Computer-Aided Semantic Annotation of Multimedia).

To deal with uncertainty, the main epistemological commitment [Russell and Norvig, 2003, p. 242] is that one should be able to express some kind of *degree of belief in a certain statement* (i.e., the degree of belief that a formula holds or does not hold). Note that approaches using degrees of belief are indeed compatible with classical logic. In both approaches each statement is considered to be either true or false. For instance, if an agent believes the formula $Bird(tweety)$ is true with degree 0.9, the individual *tweety* is an element of the set of birds in 90 percent of all worlds the agent can imagine. Nevertheless, for a particular world *tweety* is either a member of $Bird$ or not.[1]

As in classical logic, also in probabilistic logics, it is the goal to compute conclusions (entailments, satisfiability results, etc.). In the case of probabilistic logics, the degree of belief in the explicit statements should systematically influence the degree of belief the formalism associates with the conclusions (implicit statements). If, for instance, media retrieval is formalized by computing conclusions (a deduction problem), the degree of belief associated with the conclusion should somehow model the relevance of a media document for a given information retrieval query. This also holds for induction or abduction problems.

Degrees of belief can be represented using probability theory, which is a well-established research field. One has to keep in mind that there are many probabilistic formalisms, each of which has its pros and cons. In particular, it is important to understand that most approaches do not support relational structures but just provide so-called random variables that can take either boolean values or values from a discrete or continuous domain.[2] Simple probabilistic formalisms involving boolean or discrete domains have a strong correspondence with propositional logic. So-called events are described using propositional formulae (called propositions for short). Propositions can be either true or false. Relational structures, as supported by first-order logic (predicate logic), however, are based on richer ontological commitments [Russell and Norvig, 2003, p. 242]. The world is assumed to consist of objects which can be set into relation to each other. Objects can be denoted using names (or terms). Relations are denoted by predicate symbols. Nonetheless, also in first-order logic, statements about objects (formulae) can be either true or false.

A combination of first-order logic and a probabilistic formalism for representing uncertain information about relational structures and for reasoning about uncertain information in this context is advantageous in many applications. In non-trivial applications propositional logic usually suffers from the variable explosion problem. In addition, degrees of beliefs concerning the *relation of objects* cannot appropriately be represented in propositional probabilistic formalisms. For this reason, so-called first-order probabilistic approaches have been devised. In the last ten years, research on probabilistic first-order formalisms has gained substantial interest in the research community as well as in industry.

In this report we focus on three major approaches: the Bayesian and the Markovian style of modeling (see, e.g., [Pearl, 1988, Chapter 3] or [Koller et al., 2007]) as well as the so-called P-$\mathcal{SHOQ}(D)$ formalism [Giugno and Lukasiewicz, 2002a]. The so-called Bayesian style of modeling is specified with acyclic graphs representing conditional independence assumptions that make

[1] In so-called fuzzy logics, the view is that *tweety* is to some extent a member of $Bird$. In these formalisms there is a *degree of truth for a statement* and not a degree of belief that the statement is true (or false). While in probabilistic logics, the membership function (characteristic function) just returns 0 and 1, in fuzzy logics, membership is characterized using values from the interval $[0, 1]$. Fuzzy logics are better suited for modeling vagueness rather than uncertainty.

[2] Obviously, the booleans are a special discrete domain. For the CASAM context, only boolean or discrete domains are considered as relevant.

probabilistic reasoning practical. Due to the graphical structure, the formalism is usually referred to as Bayesian networks. Bayesian netword are, for instance, explained in [Russell and Norvig, 2003, Chapter 14] and will also be shortly described in Chapter 3 of this report for the sake of completeness. Probabilistic Relational Models (PRMs) [Getoor et al., 2001] augment this formalism with the possibility to handle assertions about objects, their attributes, and relations to other objects. Although PRMs are influential, the logical formalization of PRMs is weak (they are based on object-oriented programming). Since we focus on first-order probabilistic structures in order to support more structured logic-based domain modeling in Chapter 3 we also introduce well-founded first-order Bayesian network approaches based on description logics, although these formalisms are less prominently discussed in the literature.

Concerning the Markovian style of modeling, which is specified with so-called Markov networks, currently, there are two main approaches: Relational Markov networks (RMNs) [Taskar et al., 2007] and Markov logic networks (MLNs) [Domingos and Richardson, 2007]. The former formalism is also specified with relational structures but with less well-developed formal logical foundation. The latter, however, is based on first-order logic and Markovian probabilistic structures, and seems highly relevant for modeling probabilistic first-order structures. This approach is discussed in detail in Chapter 4.

In Chapter 5 the P-$\mathcal{SHOQ}(D)$ formalism is described. As we will see, P-$\mathcal{SHOQ}$ represents yet another approach to formalize probabilistic reasoning about first-order structures. In order to investigate the expressivity of P-$\mathcal{SHOQ}(D)$ we analyse application problems with P-$\mathcal{SHOQ}$.

We argue that the formalism of Markov logic networks seems suitable for CASAM, but point out some deficiencies (in particular problems concerning ontology engineering w.r.t. predicate logic). To overcome these problems, we propose a combination of description logics and Horn logics with Markov networks to enhance ontology engineering and to guarantee the decidability of decision problems in a yet expressive probabilistic formalism.

2 Preliminaries

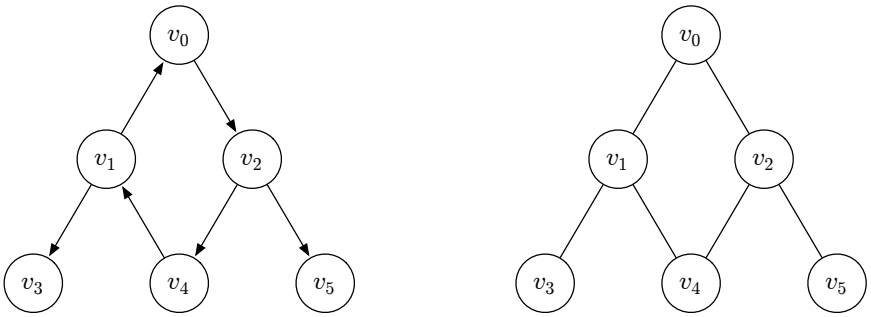
The modeling style presented in this report have several notions and definitions in common. We introduce the section in order to harmonize the presentation of different probabilistic representation formalisms.

- **Graph:**
  A graph $G = (V, E)$ is composed of a finite set of vertices V and a finite set of edges $E \subseteq V \times V$. Thus, $E = \{(v_i, v_j) \mid v_i, v_j \in V\}$ representing the fact that there are edges from v_i to v_j, respectively. G is said to be **directed** (Figure 1(a)), if for each v_i, v_j it holds that $(v_i, v_j) \in E$ implies $(v_j, v_i) \notin E$ and, analogously, G is said to be **undirected** (Figure 1(b)), if for each v_i, v_j it holds that $(v_i, v_j) \in E$ implies $(v_j, v_i) \in E$. Directed edges are depicted with arrows and undirected edges with simple lines. Figure 1(a) shows an example of a directed graph and its undirected counterpart. $V = \{v_0, \ldots, v_5\}$, $E = \{(v_0, v_2), (v_1, v_0), (v_1, v_3), (v_2, v_4), (v_2, v_5), (v_4, v_1)\}$ for the directed graph, and the set of edges for the corresponding undirected graph results by adding all tuples to E needed in order to ensure symmetry).

- **Path:**
  A path P in a directed graph G is a sequence of incident vertices where the v_i are all distinct. An example for a path P in Figure 1(a) is $v_0 \rightsquigarrow v_2 \rightsquigarrow v_4 \rightsquigarrow v_1 \rightsquigarrow v_3$.

- **Cyclic Graph:**
  A cycle C is a path $P = v_i \rightsquigarrow v_j \rightsquigarrow \ldots \rightsquigarrow v_i$ which begins and ends with the same vertex. In Figure 1 in both graphs there is the cycle $C_1 = v_0 \rightsquigarrow v_2 \rightsquigarrow v_4 \rightsquigarrow v_1 \rightsquigarrow v_0$. In the undirected graph there is the additional cycle $C_2 = v_0 \rightsquigarrow v_1 \rightsquigarrow v_4 \rightsquigarrow v_2 \rightsquigarrow v_0$. A graph G is called cyclic if it contains at least one cycle, otherwise it is called acyclic.

- **Parents:**
  A function $Parents : V \rightarrow U$ maps a vertex v_i to a set U where $U \subset V$. In Figure 1(a), $Parents(v_2) = \{v_0\}$.



(a) Directed graph (b) Undirected graph

Figure 1: Example of a directed and an undirected graph

In the following, we define the basic probability notations based on the notation used in [Russell and Norvig, 2003]. Examples are given from the environment domain used in the CASAM project.

- **Random variable:**
  A random variable X is a variable whose value depends on the result of a random experiment.

  There are two types of random variables: discrete and continuous ones. Unlike a continuous random variable, a disctere random variable takes only finite distinct values. In this report, we consider only discrete random variables. Note that a probability distribution is assigned to a discrete random variable. Assume random variable X denotes the number of rainy days in three successive days, where R indicates a rainy day and S a sunny day. Table 1 depicts all possible events E and the associated number of rainy days:

E	RRR	RRS	RSR	SRR	RSS	SRS	SSR	SSS
X	3	2	2	2	1	1	1	0

Table 1: Example of a random variable

- **Domain of a random variable:**
  The domain of a random variable $dom(X)$ is a set of possible values that a random variable can take. A random variable X is called binary if $dom(X) = \{true, false\}$. Otherwise, it is called a multi-valued random variable. For instance, the domain of a discrete random variable $WaterPollution$ has the following values which indicate the water pollution rate:

$$dom(WaterPollution) = \{low, medium, high\} \tag{1}$$

  In Table 1, $dom(X) = \{0, 1, 2, 3\}$.

- **Event:**
  An atomic event $X = x_i$ where $x_i \in dom(X)$ is a result of a random experiment. For example, an atomic event from the above example is $WaterPollution = low$.
  An Event consists of multiple atomic events.

- **Prior probability:**
  A prior probability or unconditional probability $P(A = true)$ is the probability or likelihood that proposition A is true. We use the prior probability if no other information about proposition A is given. Otherwise we use the conditional probability which is described later. A shorthand for $P(A = true)$ is $P(a)$ and for $P(A = false)$ it is $P(\neg a)$. For example, $P(OilPollution = true) = 0.1$ indicates that the probability of $OilPollution = true$ is 0.1.

- **Probability distribution:**
  A probability distribution is a vector of probabilities assigned to the possible values in the domain such that their sum is set to one. Assume the probability distribution:

$$\mathbf{P}(WaterPollution) = \langle 0.5, 0.3, 0.2 \rangle \tag{2}$$

This means, the associated probabilities to the events are, respectively:

$$\begin{aligned} P(WaterPollution = low) &= 0.5 \\ P(WaterPollution = medium) &= 0.3 \\ P(WaterPollution = high) &= 0.2 \end{aligned} \tag{3}$$

- **Full joint probability distribution:**
  Assume two random variables X and Y. The joint probability distribution of X and Y, indicated by $P(X = x_i, Y = y_j)$ or $P(X = x_i \wedge Y = y_j)$, describes their probability of occurring together. For example, the probability of $AirPollution = true$ and $Rain = true$ is indicated by:

  $$P(AirPollution = true \wedge Rain = true)$$

  In the full joint probability distribution, we consider all random variables involved in the experiment. Assume n random variables $X_1, \ldots, X_n$. Consequently $P(X_1 = x_1, \ldots, X_n = x_n)$ is a full joint probability distribution.

- **Conditional probability:**
  A conditional probability or posterior probability $P(A = true|B = true)$ is the probability of event $A = true$ under the condition that event $B = true$ is given. This is defined as follows:

  $$P(A = true|B = true) = \frac{P(A = true \wedge B = true)}{P(B = true)} \tag{4}$$

  The above equation holds if $P(B = true) > 0$.
  For instance, the probability of $Flood = true$ given $Rain = true$ is indicated by:

  $$P(Flood = true|Rain = true)$$

  where $Flood$ and $Rain$ are binary random variables. This probability is determined by

  $$P(Flood = true|Rain = true) = \frac{P(Flood = true \wedge Rain = true)}{P(Rain = true)}. \tag{5}$$

3 Bayesian style of modeling

Bayesian networks [Pearl, 1988] are one of the most inferential frameworks for representing and reasoning with probabilistic models. They are used in many real-world applications including diagnosis, forecasting, automated vision, sensor fusion and manufacturing control. In the next sections, syntax and semantics of Bayesian networks are discussed.

Syntax:

A Bayesian network $BN = (G, \gamma)$ consists of a directed acyclic graph $G = (V, E)$ and a function $\gamma : V \rightarrow T$ which maps a vertex v to a conditional probability distribution T. Note that a vertex v indicates a random variable and an edge $e_{ij} = (v_i, v_j)$ shows the direct influence of parent vertex v_i to a child vertex v_j. A conditional probability distribution T_i has the form $\mathbf{P}(X_i|Parents(X_i))$. If $Parents(X_i) = \emptyset$, then T_i specifies a prior probability.

Semantics:

The semantics of a Bayesian network BN can be seen in two different ways [Russell and Norvig, 2003]:

1. The structure of a Bayesian network is determined by the insertion order of the vertices. During the network construction, vertices are added to the network individually. After adding a vertex, conditional dependencies of the new vertex to the vertices of the current network are checked. If there are dependencies, incoming edges to the new vertex are added. Each incoming edge comes from a previously added vertex which has conditional dependency to the new vertex. Consequently, the first semantics of a Bayesian network indicates that the structure of a Bayesian network shows the conditional independence relationships which hold among the variables in the domain.

2. Bayesian networks are representions of full joint probability distributions of the domain variables $P(X_1 = x_1, \ldots, X_n = x_n)$. Equation 6 indicates the solution to the full joint probability distribution of a set of graph vertices:

$$\mathbf{P}(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} P(X_i = x_i | Parents(X_i)) \tag{6}$$

   It means, the joint probability distribution of a set of variables is a product of their conditional probability distributions. If $Parents(X_i) = \emptyset$, then a prior probability is inserted.

3.1 Example of a Bayesian network

In this example we consider two events which influence the human health namely air pollution and noise pollution. One of the factors which affects air- and noise pollution is traffic jam. These relationships are modelled by the Bayesian network graph in Figure 2:

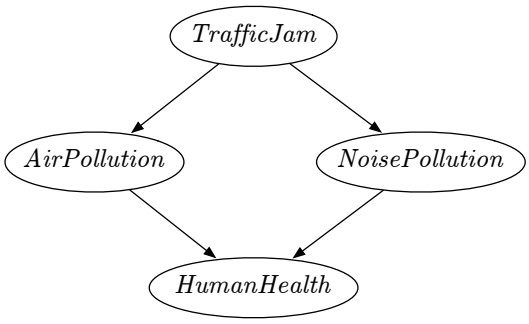


Figure 2: Example of a Bayesian network graph

TrafficJam and *HumanHealth* are conditionally independent given *AirPollution* and *NoisePollution* therefore there is no link between them. Similarly *AirPollution* and *NoisePollution* are conditionally independent given *TrafficJam*. Figure 3 depicts the above Bayesian network with conditional probability distributions. The variables *TJ*, *AP*, *NP* and *HH* stand for *TrafficJam*, *AirPollution*, *NoisePollution* and *HumanHealth*, respectively. All random variables are binary.

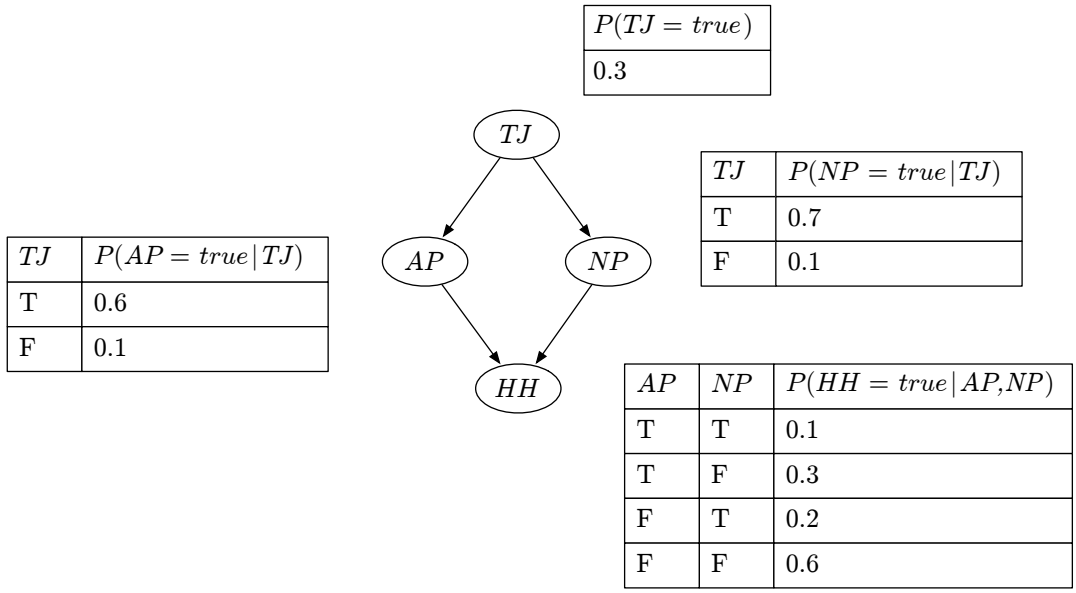


Figure 3: Example of a Bayesian network

In the Bayesian networks, conditional probability distributions T are given in the form of tables which are assigned to vertices. Since TJ has no parents, only prior probabilities are assigned to them. The probability of TJ=*true* is 0.3. Consequently, the probability of TJ=*false* is 0.7. Since AP has only one parent, its conditional probability distribution has two rows. For example, the first row means that the probability of AP=*true* is 0.6, if TJ is true. HH has two parents, consequently its conditonal probability table has four rows. The first row means that the probability of HH=*true* is 0.1 if AP and NP are both *true*.

An example for a full joint distribution is $P(\neg hh, ap, np, tj)$ which is computed based on Equation 6 as follows:

$$\begin{aligned} P(\neg hh, ap, np, tj) &= P(tj)P(ap|tj)P(np|tj)P(\neg hh|ap, np) && (7) \\ &= 0.3 \times 0.6 \times 0.7 \times 0.9 \\ &= 0.1134 && (8) \end{aligned}$$

3.2 Inference in Bayesian networks

In this section, we discuss the decision problem in Bayesian networks which is called *inference*. The objective of inference is the computation of a posterior probability $\mathbf{P}(X|\mathbf{E} = \mathbf{e})$ for a query variable X given a set of evidence variables $\mathbf{E} = \{E_1, E_2, ...\}$ where $\mathbf{e}$ is a tuple of particular observed evidence. In addition to X and $\mathbf{E}$, there is a set of nonevidence variables $\mathbf{Y} = \{Y_1, Y_2, ...\}$ which are considered in the solution of the inference problem. There are two main solutions for the inference problem, namely exact inference and approximate inference.

3.2.1 Exact inference

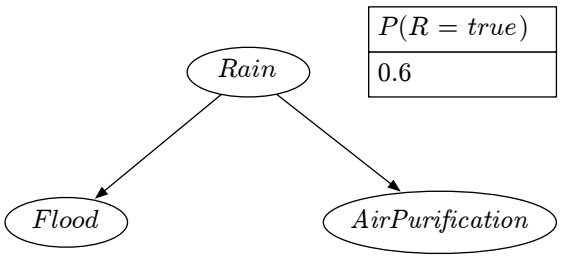
The exact inference solves the inference problem by the full joint distribution:

$$\mathbf{P}(X|\mathbf{E} = \mathbf{e}) = \alpha\, \mathbf{P}(X, \mathbf{E} = \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{E} = \mathbf{e}, \mathbf{y}) \quad (9)$$

where the summation is over all nonevidence (hidden) variables $\mathbf{Y}$. In the above equation, α denotes a normalization constant. With this notation, the above equation can be written as a

full joint distribution. The complexity of exact inference for a Bayesian network with n boolean variables is $O(n2^n)$. It shows that the complexity of exact inference for large networks is very high. In the following, an example for exact inference is given.

The next figure depicts a Bayesian network where the relationships between rain, flood and air purrification are given. Since *Flood* and *AirPurrification* are conditionally independent, there is no edge between them. In the conditional probability distributions R, F and AP respectively stand for *Rain*, *Flood* and *AirPurification* which are all binary variables:



$P(R = true)$
0.6

R	$P(F = true \mid R)$
T	0.6
F	0.1

R	$P(AP = true \mid R)$
T	0.8
F	0.4

Figure 4: Example of a Bayesian network

The next table depicts the full joint distribution for the three boolean random variables:

	$flood$		$\neg flood$	
	$airPurification$	$\neg airPurification$	$airPurification$	$\neg airPurification$
$rain$	0.24	0.06	0.16	0.04
$\neg rain$	0.01	0.03	0.09	0.27

Table 2: A full joint distribution for *Rain*, *Flood* and *Airpurrification* world

— In Equation 10 we compute the probability of $Rain = true$ given $Flood = true$, where *Rain* is a query variable and *Flood* is an evidence variable:

$$P(rain|flood) = \frac{P(rain \wedge flood)}{P(flood)} = \frac{0.24 + 0.06}{0.24 + 0.06 + 0.01 + 0.03} = 0.88 \tag{10}$$

Similarly, we compute the probability of $Rain = false$ given $Flood = true$:

$$P(\neg rain|flood) = \frac{P(\neg rain \wedge flood)}{P(flood)} = \frac{0.01 + 0.03}{0.24 + 0.06 + 0.01 + 0.03} = 0.12 \tag{11}$$

The term $1/P(Flood = true)$ in Equations 10 and 11 is a normalization constant, which causes the sum of the above probabilities to be set to one. The nonevidence variable in the above inference is *AirPurrification*. If we use probability distributions, the above equations can be written in a single equation:

$$\begin{aligned} \mathbf{P}(Rain|flood) &= \alpha\, \mathbf{P}(Rain, flood) \\ &= \alpha\, [\mathbf{P}(Rain, flood, airPurification) + \mathbf{P}(Rain, flood, \neg airPurification)] \\ &= \alpha\, [\langle 0.24, 0.01\rangle + \langle 0.06, 0.03\rangle] \\ &= \alpha\, [\langle 0.30, 0.04\rangle] = \langle 0.88, 0.12\rangle \end{aligned} \tag{12}$$

3.2.2 Approximate inference

As it was discussed in the previous section, the complexity of exact inference for large networks is very high. Therefore approximate inference methods have been developed. The functionality of these methods is the generation of samples for the considered random variables. The accuracy of sampling methods depends on the number of samples. It means, generating more samples leads to higher accuracy and consequently the result converges to the result of exact inference. In the next section, we introduce one of the sampling methods and its functionality.

Direct sampling method In this section, we describe direct sampling which is based on the generation of samples. To generate these samples, the creation of random numbers in the interval $[0, 1]$ is needed. In a Bayesian network, the sampling process is performed for all random variables (the nodes) of the network, no matter whether they are query-, evidence- or nonevidence variables of the inference problem. The order, in which the sampling takes place, is geared to the topological appearance of the nodes. The required probability distributions of the nodes are known and are given by the conditional probability tables. Sampling a Bayesian network leads to the generation of events, which are sets of boolean assignments for all the variables of the network.

Consider the example depicted in Figure 4. *Rain* is given as a parent node of *Flood* as well as of $AirPurification$, hence the sampling order is $[Rain, Flood, AirPurification]$. The probability distribution from which the values are sampled depends on the values that were assigned to the variable's parents. An examplary creation of an event includes the following three steps:

1. Sample from $\mathbf{P}(Rain) = \langle 0.5, 0.5 \rangle$; suppose this returns *true*.
2. Sample from $\mathbf{P}(Flood|Rain = true) = \langle 0.6, 0.4 \rangle$; suppose this returns *true*.
3. Sample from $\mathbf{P}(AirPurification|Rain = true) = \langle 0.8, 0.2 \rangle$; suppose this returns *true*.

The resulting event would be $[true, true, true]$.

Because each step in the sampling process depends only on the parent values, the probability that a specific event $S(x_1, \ldots, x_n)$ is generated can be expressed by

$$S(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|Parents(X_i)). \tag{13}$$

This probability is equivalent to the full joint distribution of the Bayesian network for the given event as shown in Equation 6. Any sampling algorithm is based on counting the generated samples. If a total number of N samples is generated and the frequency of the specific event $x_1, \ldots, x_n$ is $N(x_1, \ldots, x_n)$, the limit of the fraction

$$\lim_{N \to \infty} \frac{N(x_1, ..., x_n)}{N} \tag{14}$$

is expected to converge to its sampling probability $S_{PS}(x_1, \ldots, x_n)$ and accordingly also to the expected probability $P(x_1, \ldots, x_n)$. For the given example, the sampling propability is

$$S_{PS}(true, true, true) = 0.5 \times 0.6 \times 0.8 = 0.24, \tag{15}$$

so for large N, 24% of the samples are expected to be of this event.

Rejection sampling in Bayesian networks Rejection sampling is an approximate inference method which is used for the computation of inference problems. It is based on an estimated probability function $\hat{\mathbf{P}}(X|\mathbf{E} = \mathbf{e})$. The idea of this method is the generation of samples for the considered evidence variables $\mathbf{E}$. Afterwards, the samples which do not match $\mathbf{E} = \mathbf{e}$ are rejected. The remaining samples show the appearance frequency of X. By using sampling, the estimated probability distribution can be written as:

$$\hat{\mathbf{P}}(X|\mathbf{E}=\mathbf{e}) = \frac{\mathbf{N}(X, \mathbf{E}=\mathbf{e})}{N(\mathbf{E}=\mathbf{e})} \tag{16}$$

where N indicates the number of samples and $\mathbf{N}$ is a vector of counts over values of X. For example, consider the conditional probability $\mathbf{P}(Rain|flood)$. Assume we have 100 samples. Of the produced 100 samples, suppose that 54 have $flood = false$ and are rejected, whereas 46 have $flood = true$. From the 46 samples, 37 have $Rain = true$ and 9 have $Rain = false$. Consequently,

$$\hat{\mathbf{P}}(Rain|flood) \approx \alpha \langle 37, 9 \rangle = \langle 0.80, 0.20 \rangle \tag{17}$$

The correct answer is $\langle 0.88, 0.12 \rangle$. By generating more samples, the result converges to the correct answer. The disadvantage of the rejection algorithm is that it takes a long time to collect correct samples since this algorithm drops many samples which do not have the prerequirements.

In the next section, the advantages and disadvantages of Bayesian networks are discussed.

3.3 Advantages and disadvantages of Bayesian networks

The advantages of Bayesian networks are as follows:

- Bayesian networks are one of the best-understood models for representing the joint probability distribution of a domain.
- It has a good graphical representation which shows the local influences among the random variables.
- Since Bayesian networks are used for modeling causality, we can understand the conditional probability distributions.

Despite these interesting properties, Bayesian networks have the following disadvantages:

- In Bayesian networks, it is not possible to refer to objects and their relations. For example, we can not refer to the *TrafficJam* in a particular city like Hamburg or a *Flood* in Berlin.
- modeling of the Bayesian network with cycles is not allowed. The Bayesian network graph in Figure 5 represents an environmental relation. We insert the nodes in the following order from top to bottom: *Condensation*, *Precipitation* and *Evaporation*. Afterwards, to determine the edges in each step, we check the dependency of each node to the previously inserted nodes. This means that the node *Precipitation* has no outgoing edge to *Condensation*. Similarly, the node *Evaporation* has no outgoing edge to other nodes:

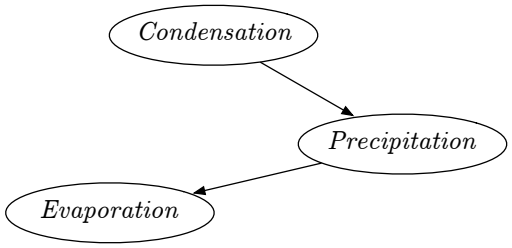


Figure 5: Bayesian network graph representing an environmental relation

As it can be seen in the above Bayesian network, a water cycle is not represented in a cyclic form.

- The complexity of decision problems in Bayesian networks is NP-hard.

3.4 First-order logic

Unlike propositional logic, where it is only possible to represent information on simple statements/propositions $A, B, C, ...$ with boolean operators $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, etc. resulting in formulas such as e.g. $((A \rightarrow B) \leftrightarrow ((A \wedge \neg B) \vee C))$, in first-order logic (FOL) [Fitting, 1996] there is the additional possibility to handle assertions about **objects** and their **relations**. Atomic formulas (i.e. formulas without any operator) in FOL have a complex structure. They are represented with $R(t_1, ..., t_n)$ for a predicate R denoting a relation of the chosen domain and terms $t_1, ..., t_n$, each term t_i being either a constant denoting an object of the domain, a variable or (inductively) a function of terms. The second enlargement of expressivity are quantified formulas $\forall x\,[F]$ (for all x, F holds) and $\exists x\,[F]$ (there is an x such that F holds). It is possible to substitute variables appearing in terms of first-order formulas with constants. Since resulting terms consist of explicit constants denoting real world objects (and explicit function terms assigning to real world objects), these formulas (resp. terms) are referred to as being **grounded** and corresponding substitutions are also called **groundings** in the following.

3.5 Probabilities and first-order logic

Modelling uncertainty in the context of description logics has been a topic of research for many years. An overview of such extensions to classical description logics is presented in [Baader et al., 2003c]. The research is oriented to the work of modelling uncertain knowledge on the basis of first-order structures [Nilsson, 1986, Bacchus, 1990, Halpern, 1990]. The fundamental view of the approaches based on description logics is such that it should also be possible to represent the degree of overlap between concepts (and not only subsumption or disjunction) through probabilities. Furthermore it should also be possible to formulate uncertainty about the structure of objects. Initial approaches considered primarily probabilistic knowledge at the conceptual level, this means, at the level of the TBox [Heinsohn, 1994]. Also knowledge representation for single objects and their relations from a probabilistic view were studied [Jaeger, 1994], such that structural uncertainty could potentially be modeled. Along with early research results about decidability of very expressive logics (e.g. OWL DL), proposals for the modelling of uncertain knowledge were given.

It is important to observe that the semantics used in the different approaches do not differ much (for example w.r.t. [Jaeger, 1994] and [Giugno and Lukasiewicz, 2002b]). An approach for the modelling of uncertain structures for a less expressive language is presented in [Dürig and Studer, 2005]. However, no specific inference algorithms are known for this approach. An important step for the practical use of description logics with probabilities occurred with the integration of Bayesian networks in P-CLASSIC [Koller et al., 1997], nevertheless very strong disadvantages were obtained: for number restrictions the supremum limits must be known and separate Bayesian networks are necessary to consider role fillers. Along with this problem, the probabilistic dependencies between instances must also be modeled. This problem was overcome in [Koller and Pfeffer, 1998] - however not in the context of description logics but with a frame-based approach, in which the treatment of default values is given without formal semantics. The main idea in [Koller and Pfeffer, 1998] is the view of considering role fillers as nodes in Bayesian networks which have CPTs (conditional probability tables) associated to them as generalized number restrictions in the sense of description logics. Related studies followed in [Pfeffer et al., 1999].

Complementary to the P-CLASSIC approach, another approach called PTDL [Yelland, 2000] was developed for probabilistic modelling with the use of first-order structures. In this approach the Bayesian network theory is considered as basis reference for further extensions, instead of (classical) description logics. The Bayesian network nodes represent function values and an individual is associated to other nodes through these function values. The approach in [Yelland, 2000] avoids some disadvantages of P-CLASSIC, but it offers minimal expressivity on the side of description logics. In context with very expressive description logics another approach [Ding and Peng, 2004, Ding et al., 2005] was presented for the integration of Bayes networks. Algorithms for deduction over probabilistic first-order structures were developed by Poole [Poole, 2003]. Poole observes, that the existing approaches (e.g. [Koller and Pfeffer, 1998, Pfeffer et al., 1999]) only consider individuals that are explicitly named. Qualitative probabilistic matching with hierarchical descriptions was studied [Smyth and Poole, 2004]. It allows for a variation of the level of abstraction.

Previous studies have investigated the combination of Datalog and description logics (so-called description logic programs) [Nottelmann and Fuhr, 2004, Lukasiewicz, 2005a, Lukasiewicz, 2005b, Nottelmann and Fuhr, 2006]. Approaches for information retrieval with probabilistic Datalog are presented in [Fuhr, 2000, Fuhr, 1995]. In this area, work on learning from Datalog-predicates with uncertainty is also relevant [Nottelmann and Fuhr, 2001].

While [Hobbs et al., 1993, Shanahan, 2005] use first-order logic for text and image/video interpretation, with description logics, we argue to use a decidable knowledge representation formalism with well-tested implementations that are known to be efficient for many typical-case inputs (see, e.g., [Espinosa et al., 2007, Espinosa et al., 2008, Castano et al., 2008]). The use of logical rules and backward chaining for implementing an abduction algorithm is also investigated in the area of logic programming [Kakas et al., 1992, Poole, 1993a, Poole, 1992, Kakas and Denecker, 2002, Flach and Kakas, 2000]. In our approach, however, predicate names in rules are defined w.r.t. ontologies represented as description logic Tboxes, and thus we use another expressive fragment of first-order logic. In the context of information retrieval, user queries can be answered regarding user-specified Tboxes. In the previous sections, we have argued that probabilistic reasoning would really add to the application scenario of information retrieval we have used in this chapter. In [Sebastiani, 1994] an proposal is made for using probabilistic description logics for information retrieval. No system implementation has been developed, though.

In the previous section we have discussed related work for integrating probabilistic and description logic reasoning. Only recently, however, abduction has been investigated in the context of description logics [Colucci et al., 2004]. However, in this work, abduction is considered for concepts, not Aboxes and queries. Due to the best of our knowledge, abduction has not yet been considered in the context of probabilistic description logics. Interesting input to this research is provided by abduction in probabilistic logic programming [Charniak and Goldman, 1991, Poole, 1993b].

4 Markovian style of modeling

In this chapter, the formalism of Markov logic networks [Domingos and Richardson, 2007] is introduced, which emerged from Markov networks and first-order logic. First, in Section 4.1 it is shown that Markov networks allow for modeling dependencies of a set of random variables with undirected and cyclic graphs in order to determine degrees of beliefs of the values of these random variables. After that, in Section 4.2 it is explained that knowledge representation with first-order logic provides a means to handle assertions about objects and their relations. Finally, in Section 4.3 the formalism of Markov logic networks is presented, which results from the combination of Markov networks and the expressivity of first-order logic. The chapter finishes with advantages and disadvantages of this approach.

4.1 Markov networks

Like Bayesian networks, Markov networks allow for modeling joint distributions of a set of random variables $X = \{X_1, ..., X_n\}$ [Pearl, 1988, Chapter 3] [Koller et al., 2007]. A Markov network $MN = (G, \Phi)$ is composed of a graph $G = (V, E)$ and a set of potential functions Φ. G consists of a set of nodes $V = \{v_1, ..., v_n\}$ (each node v_i represents the random variable X_i) and a set of edges E between nodes. The graph is assumed to be irreflexive, i.e. $E \subseteq \{(v_i, v_j) \mid v_i, v_j \in V \wedge i \neq j\}$. In contrast to graphs of Bayesian networks, G is undirected and possibly cyclic. These properties are necessary to immediately model interrelations between random variables. For example, referring to the environmental domain of the CASAM project, consider the natural water cycle: Ascending air condensates (i.e. changes its physical state of aggregation from gaseous to liquid), resulting water precipitates (i.e. is transferred from the atmosphere to earth) and then evaporates (i.e. becomes gaseous again). An undirected graph of this cycle and some of its consequences could be visualized as in Figure 6.

Each edge in this graph represents a dependency between corresponding random variables. The dependencies between *Condensation*, *Precipitation* and *Evaporation* are cyclic and therefore cannot be modeled with Bayesian networks directly. The edges between these nodes, though, are intended to be directed. This is not the case for the edge between *HumanProliferation* and

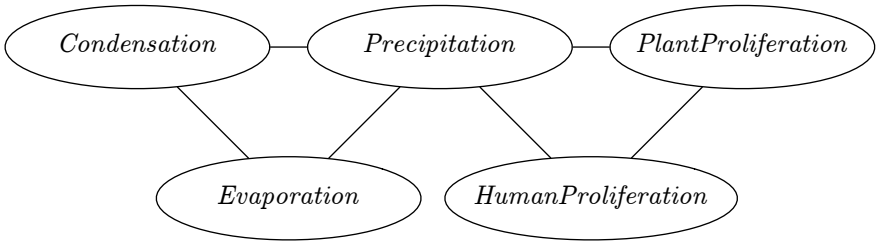


Figure 6: Markov network graph modeling the water cycle and some of its consequences

PlantProliferation: *PlantProliferation* has influence on *HumanProliferation* and vice versa (since humans settled down thousands of years ago).

If a node X_i is not directly connected to another node X_j, it is **conditionally independent** of X_j given the set Z of nodes directly connected to X_i. For example, since *Condensation* is not directly connected to both *HumanProliferation* and *PlantProliferation*, it is possible to compute the conditional probability of *Condensation* given only *Precipitation* and *Evaporation*.

A **clique** is a maximal subgraph of G whose nodes are connected to each other. Concerning the example, there are two cliques, each composed of three nodes: The cliques *Condensation* − *Precipitation* − *Evaporation* as well as *Precipitation* − *PlantProliferation* − *HumanProliferation*. In other scenarios, there are also cliques consisting of more or less than three nodes (and therefore not necessarily of cycles). The set of all cliques of a graph G is $C(G) = \{c_1, ..., c_m\}$.

Possible worlds $\vec{x} = <x_1, ..., x_n>$ are instantiations of all random variables $\vec{X} = <X_1, ..., X_n>$ of a Markov network MN. If they are observed as tuples $(x_1, ..., x_n)$, the set of all these instantiations is $\Gamma = \{\vec{x_1}, ..., \vec{x_r}\}$. In order to quantify the edges, **potential functions** $\phi_c \in \Phi$ have to be defined for each clique $c \in C(G)$ in the graph, depending on a world $\vec{x}_c$ for which only random variables appearing in c have to be considered, i.e. the clique defines the arity of ϕ_c (note the advantage of this locality). The **full joint probability distribution of Markov networks** is given by

$$P(\vec{X} = \vec{x}) = \frac{1}{Z} \prod_{c \in C(G)} \phi_c(\vec{x}_c) \tag{18}$$

i.e. the probability of an event $\vec{X} = \vec{x}$ (a world $\vec{x}$) is computed with the product of all clique-potentials divided by the normalization scalar Z, the sum of potential function products for all possible worlds $\vec{x} \in \Gamma$:

$$Z = \sum_{\vec{x} \in \Gamma} \prod_{c \in C(G)} \phi_c(\vec{x}_c)$$

Dividing by Z guarantees that the sum of the probabilites of all possible worlds $\vec{x}$ is 1.

Consider a Markov network graph only representing the water cycle (Figure 7).

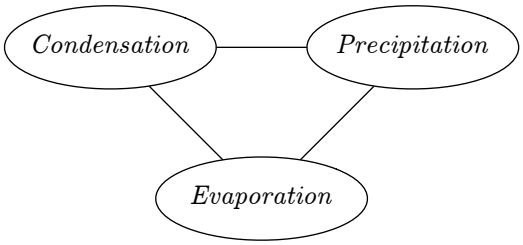


Figure 7: Markov network graph representing the water cycle

If occuring variables *Condensation*, *Precipitation* and *Evaporation* are intended to be binary (i.e., their values are either *true* or *false*), then the assignment *Condensation* = *true* can be abbreviated with c and *Condensation* = *false* with $\neg c$ (analogously for the other variables) and there

are eight possible combinations of the assignments of the three random variables constituting eight possible worlds:

$$\begin{array}{llll} \vec{x_1} & = & <c, p, e> & \vec{x_5} & = & <c, \neg p, \neg e> \\ \vec{x_2} & = & <c, p, \neg e> & \vec{x_6} & = & <\neg c, p, \neg e> \\ \vec{x_3} & = & <c, \neg p, e> & \vec{x_7} & = & <\neg c, \neg p, e> \\ \vec{x_4} & = & <\neg c, p, e> & \vec{x_8} & = & <\neg c, \neg p, \neg e> \end{array}$$

If the potential function ϕ_{water} for the water cycle clique is defined, the probabilities of these eight worlds can be computed: Given the arbitrary potential function values

$$\phi_{water}(\vec{x_{i_{water}}}) = \begin{cases} 1500 \text{ , if } i = 1 \text{ or } i = 8 \\ 120 \text{ , else} \end{cases}$$

(equal values of the three variables are considered more likely) the probability of e.g. $\vec{x_1}$ is

$$\frac{1500}{Z} = \frac{1500}{1500 + 120 + 120 + 120 + 120 + 120 + 120 + 1500} \approx 0,4.$$

By observing random variables as propositions, there is a strong connection to propositional logic: Concerning the modeling of the water cycle, a corresponding formula in this logic is $F = Condensation \wedge Precipitation \wedge Evaporation$ consisting of a conjunction of three propositions. Since the semantics of propositional logic is defined with interpretation functions $\mathcal{I}$ assigning truth values (*true* resp. *false*) to each proposition, each of the eight worlds $\vec{x_i}$ presented above can be expressed, if corresponding interpretations for F are considered. For example, the world $\vec{x_2} = <c, p, \neg e>$ can also be expressed as $c \wedge p \wedge \neg e$, since $\vec{x_2}$ corresponds to the interpretation $\mathcal{I}(Condensation) = true, \mathcal{I}(Precipitation) = true$ and $\mathcal{I}(Evaporation) = false$.

The representation of the full joint probability distribution with potential functions often is replaced by so-called **log-linear modeling**. In this representation, the exponent of an e-function is the sum of all real number weighted (usually binary) features f_c of the world $\vec{x}$:

$$P(\vec{X} = \vec{x}) = \frac{1}{Z} exp\,(\sum_{c \in C(G)} w_c f_c(\vec{x})) \tag{19}$$

Its advantage is that potential functions ϕ_c are broken down into components which are easier to handle and easier to understand. If these components are chosen properly, then probabilites of arbitrary worlds do not differ from probabilites computed with the potential function representation. The scalar Z is defined analogously with the sum of these e-functions over all possible worlds.

4.2 First-order logic in Markovian apporaches

It is possible to substitute variables appearing in terms of first-order formulas with constants. Since resulting terms consist of explicit constants denoting real world objects (and explicit function terms assigning to real world objects), these formulas (resp. terms) are referred to as being **grounded** and corresponding substitutions are also called **groundings** in the following.

To model dependencies between random variables, FOL formulas can be represented in a FOL **knowledge base**. Concerning the water cycle example presented at the beginning of this chapter (cf. Figure 6 and surrounding explanations), dependencies in this scenario could be specified with the following FOL knowledge base KB_1:

1. $\forall x\,[RegionWithHighCondensation(x) \rightarrow RegionWithHighPrecipitation(x)]$
2. $\forall x\,[RegionWithHighPrecipitation(x) \rightarrow RegionWithHighEvaporation(x)]$
3. $\forall x\,[RegionWithHighEvaporation(x) \rightarrow RegionWithHighCondensation(x)]$
4. $\forall x\,[RegionWithHighPrecipitation(x) \rightarrow RegionWithHighHumanProliferation(x)]$
5. $\forall x\,[RegionWithHighPrecipitation(x) \rightarrow RegionWithHighPlantProliferation(x)]$
6. $\forall x\,[RegionWithHighHumanProliferation(x) \leftrightarrow RegionWithHighPlantProliferation(x)]$

The variable x could be substituted by a constant denoting a specific local region (e.g. Northern Germany). However, KB_1 does not represent relations of the domain. To use this kind of expressivity, consider another knowledge base KB_2 representing the water cycle w.r.t. possibly adjacent regions:

1. $\forall x\,[RegionWithHighCondensation(x) \rightarrow RegionWithHighPrecipitation(x)]$
2. $\forall x\,[RegionWithHighPrecipitation(x) \rightarrow RegionWithHighEvaporation(x)]$
3. $\forall x\,[RegionWithHighEvaporation(x) \rightarrow RegionWithHighCondensation(x)]$
4. $\forall x\,[RegionWithHighPrecipitation(x) \rightarrow \exists y\,[adjacent(x,y) \wedge RegionWithHighPrecipitation(y)]]$

The fourth formula of this knowledge base represents the constraint that for each region with high precipitation there has to be at least one adjacent region also with high precipitation.

Clearly, the formulas of KB_1 and KB_2 do not represent the real world correctly. They are not always true (e.g. it is possible that there is a high condensation in region r_1 with its according high precipitation in another region r_2). Therefore, [Domingos and Richardson, 2007] apply Markov networks to FOL in order to be able to specify degrees of beliefs of formulas. The resulting formalism will be explained in the following section.

4.3 Markov logic networks

The formalism of Markov logic networks [Domingos and Richardson, 2007] provides a means to combine the expressivity of first-order logic with the formalism of Markov networks. A Markov logic network $MLN = (\mathcal{F}, \mathcal{W})$ consists of a set of first-order formulas $\mathcal{F} = \{F_1, ..., F_m\}$ and a set of real number weights $\mathcal{W} = \{w_1, ..., w_m\}$ associated to these formulas.

In contrast to conventional formulas assumed to be valid, weighted formulas F_i need not always be true. For example, as indicated above, the all-quantified formula

$$\forall x\,[RegionWithHighCondensation(x) \rightarrow RegionWithHighPrecipitation(x)]$$

might be true for a lot of regions, but also might be false for other regions. By assigning a reasonable weight to this formula, it becomes a soft constraint representing the environmental domain more appropriate. When a world $\vec{x}$ violates this weighted formula (worlds including regions with high condensation, but without high precipitation) it is less probable rather than impossible [Domingos and Richardson, 2007].

Consider the Markov logic network MLN_1 which consists of the all-quantified formulas of KB_1 prefixed with (initially arbitrary) real number weights:

2.5 $\forall x\,[RegionWithHighCondensation(x) \rightarrow RegionWithHighPrecipitation(x)]$
2.5 $\forall x\,[RegionWithHighPrecipitation(x) \rightarrow RegionWithHighEvaporation(x)]$
2.5 $\forall x\,[RegionWithHighEvaporation(x) \rightarrow RegionWithHighCondensation(x)]$
1.1 $\forall x\,[RegionWithHighPrecipitation(x) \rightarrow RegionWithHighHumanProliferation(x)]$
2.5 $\forall x\,[RegionWithHighPrecipitation(x) \rightarrow RegionWithHighPlantProliferation(x)]$
1.1 $\forall x\,[RegionWithHighHumanProliferation(x) \leftrightarrow RegionWithHighPlantProliferation(x)]$

The weights of the fourth and sixth formula of MLN_1 have been chosen lower than the other weights, because there are many regions with high precipitation but without high human proliferation respectively many regions with high plant proliferation but without high human proliferation. Usually the weights are not given. They have to be computed given evidence. This computation is called **learning** (see e.g. [Kok and Domingos, 2005]). In *MLN* [Domingos and Richardson, 2007], evidence is a relational database consisting only of atomic facts (grounded atomic formulas). Facts not specified are assumed to be false (this is the Closed-World-Assumption).

For each Markov logic network *MLN* there is an appropriate Markov network $MN = (G, \Phi)$. To create such a network, the graph G of this network contains

1. one node for each possible grounding of each predicate appearing in *MLN*
2. one edge between two nodes if and only if the corresponding ground predicates appear together in a grounding of a formula F_i in *MLN*

As can be concluded from this conditions, the approach of [Domingos and Richardson, 2007] treats cliques c as influences in grounded first-order logic formulas F_i. For example, according to the representation of the water cycle and some of its consequences with KB_1 in Section 4.2, a similar Markov network as presented in Figure 6 can be created: Suppose that it is assumed that there is only one constant ng in the domain denoting the local region Northern Germany. To state this, the **domain closure axiom** $\forall x\,[x \doteq ng]$ has to be considered which is required to hold in all possible worlds (see [Brachman and Levesque, 2004] for an explanation of this kind of axioms). The nodes of the graph of the corresponding Markov network then are given as shown in Figure 8 (with $Condensation(ng)$ as an abbreviation of $RegionWithHighCondensation(ng)$, analogously for the other nodes).

Figure 8: Nodes originating from grounding example first-order formulas

To instantiate the edges of this graph, it is needed to apply all possible groundings to the formulas of KB_1. Since there is only one region, the resulting Markov network graph G_1 is the one presented in Figure 9.

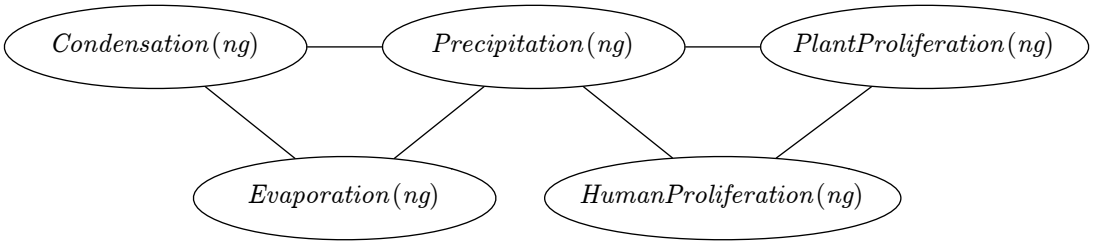


Figure 9: Markov logic network graph of the water cycle example

According to Equation 19, the log-linear representation of the full joint probability distribution of a Markov logic network *MLN* is given, if for each possible world $\vec{x}$ the value of $f_c(\vec{x})$ is determined by the number of true groundings $n_i(\vec{x})$ of formulas F_i in $\vec{x}$, $i = 1, ..., m$:

$$P(\vec{X} = \vec{x}) \;=\; \frac{1}{Z}\, exp\,(\sum_{i=1}^{m} w_i n_i(\vec{x})) \tag{20}$$

Since the *exp*-term is another representation of the product of potential functions $\phi_c \in \Phi$, the corresponding Markov network $MN = (G, \Phi)$ of a given Markov logic network $MLN = (\mathcal{F}, \mathcal{W})$ now is defined completely.

By means of an example, in the following it is shown that a world is less probable than another, if it violates more (higher weighted) formulas: With respect to MLN_1, the probability of $\vec{x_1} = <\neg C(ng), Pr(ng), E(ng), H(ng), \neg Pl(ng)>$ (where e.g. $Pr(ng)$ is an abbreviation for $RegionWithHighPrecipitation(ng) = true$),

$$P(\vec{X} = \vec{x}_1) \;=\; \frac{1}{Z}\, exp(2,5 \cdot 1 + 2,5 \cdot 1 + 2,5 \cdot 0 + 1,1 \cdot 1 + 2,5 \cdot 0 + 1,1 \cdot 0) \;\approx\; \frac{446}{Z},$$

is much lower than that of $\vec{x_2} = <C(ng), Pr(ng), E(ng), \neg H(ng), Pl(ng)>$:

$$P(\vec{X} = \vec{x}_2) \;=\; \frac{1}{Z}\, exp(2,5 \cdot 1 + 2,5 \cdot 1 + 2,5 \cdot 1 + 1,1 \cdot 0 + 2,5 \cdot 1 + 1,1 \cdot 0) \;\approx\; \frac{22026}{Z}$$

Due to its complexity (2^5 possible worlds), the computation of Z is not presented here.

By assigning the first, second and third formula of MLN_1 to the water cycle clique and the fourth, fifth and sixth to the clique of the consequences of precipitation, the probabilities of the worlds $\vec{x_1}$ and $\vec{x_2}$ can further be computed with corresponding potential functions ϕ_{water} and ϕ_{cons}, if

$$\begin{array}{lcllcll}
\phi_{water}(\vec{x_{1}}_{water}) & = & \phi_{water}(<\neg C(ng), Pr(ng), E(ng)>) & \approx & 148,41 & \approx & e^{2,5} \cdot e^{2,5} \cdot e^{0}, \\
\phi_{cons}(\vec{x_{1}}_{cons}) & = & \phi_{cons}(<Pr(ng), H(ng), \neg Pl(ng)>) & \approx & 3,004 & \approx & e^{1,1} \cdot e^{0} \cdot e^{0}, \\
\phi_{water}(\vec{x_{2}}_{water}) & = & \phi_{water}(<C(ng), Pr(ng), E(ng)>) & \approx & 1808,04 & \approx & e^{2,5} \cdot e^{2,5} \cdot e^{2,5}, \\
\phi_{cons}(\vec{x_{2}}_{cons}) & = & \phi_{cons}(<Pr(ng), \neg H(ng), \neg Pl(ng)>) & \approx & 12,1825 & \approx & e^{0} \cdot e^{2,5} \cdot e^{0},
\end{array}$$

since $148,41 \cdot 3,004 \approx 446$ and $1808,04 \cdot 12,1825 \approx 22026$.

In order to show the expressivity and complexity of Markov logic networks, another example is discussed in the following. By (initially arbitrary) assigning prefixed real number weights to the formulas of KB_2, the Markov logic network MLN_2 is defined by

2.5 $\forall x\,[RegionWithHighCondensation(x) \rightarrow RegionWithHighPrecipitation(x)]$

2.5 $\forall x\,[RegionWithHighPrecipitation(x) \rightarrow RegionWithHighEvaporation(x)]$

2.5 $\forall x\,[RegionWithHighEvaporation(x) \rightarrow RegionWithHighCondensation(x)]$

1.6 $\forall x\,[RegionWithHighPrecipitation(x) \rightarrow \exists y\,[adjacent(x,y) \wedge RegionWithHighPrecipitation(y)]]$.

Since in this Markov logic network there is a predicate relating to adjacent regions, it is reasonable that there are at least two constants denoting two regions of the domain. We assume that there are only the constants ng (denoting Northern Germany) and d (denoting Denmark) by considering the domain closure axiom $\forall x\,[x \doteq ng \vee x \doteq d]$. Possible groundings of the fourth formula then are

$$\begin{array}{lll}
RegionWithHighPrecipitation(ng) & \rightarrow adjacent(ng,ng) & \wedge\ RegionWithHighPrecipitation(ng) \\
RegionWithHighPrecipitation(ng) & \rightarrow adjacent(ng,d) & \wedge\ RegionWithHighPrecipitation(d) \\
RegionWithHighPrecipitation(d) & \rightarrow adjacent(d,ng) & \wedge\ RegionWithHighPrecipitation(ng) \\
RegionWithHighPrecipitation(d) & \rightarrow adjacent(d,d) & \wedge\ RegionWithHighPrecipitation(d).
\end{array}$$

Together with all possible groundings of the (simpler) first three formulas, they induce the Markov network graph G_2 of MLN_2 visualized in Figure 10.

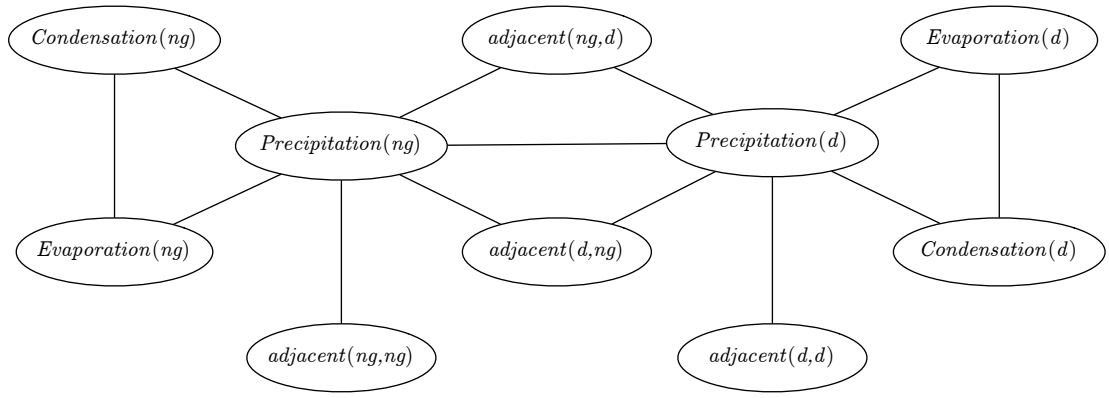


Figure 10: Markov logic network graph also representing relations

Thus, compared to G_1, in presence of additonal constants and relations, the size of a Markov network graph increases heavily, even if there are less underlying formulas.

Analogous to the previous example, probabilities of possible worlds can be computed by applying 20. With respect to MLN_2, the probability of $\vec{x_1} = <C(ng), Pr(ng), \neg E(ng), C(d), \neg Pr(d), \neg E(d), adj(ng,d), adj(d,ng), \neg adj(ng,ng), \neg adj(d,d)>$ (with e.g. $adj(ng,d)$ as an abbreviation for $adjacent(ng,d) = true$),

$$P(\vec{X} = \vec{x}_1) \;=\; \frac{1}{Z}\, exp(2,5 \cdot 1 + 2,5 \cdot 1 + 2,5 \cdot 2 + 1,6 \cdot 1) \;\approx\; \frac{109098}{Z},$$

is much lower than that of $\vec{x_2} = <C(ng), Pr(ng), E(ng), \neg C(d), \neg Pr(d), \neg E(d), \neg adj(ng,d), \neg adj(d,ng), \neg adj(ng,ng), \neg adj(d,d)>$:[3]

$$P(\vec{X} = \vec{x}_2) \;=\; \frac{1}{Z}\, exp(2,5 \cdot 2 + 2,5 \cdot 2 + 2,5 \cdot 2 + 1,6 \cdot 1) \;\approx\; \frac{16191549}{Z}$$

As can be seen from this example, there is an additional advantage of the log-linear representation of Markov networks. Since there are six cliques in G_2, a representation with potential functions is more expensive.

4.3.1 Decision problems - Querying Markov logic networks

To compute the **conditional probability** for a query X based on given evidence $\mathbf{e}$,

$$\mathbf{P}(X \mid \mathbf{e}) = \frac{\mathbf{P}(X, \mathbf{e})}{\mathbf{P}(\mathbf{e})} = \alpha\, \mathbf{P}(X, \mathbf{e}),$$

performing exact inference with the necessity of summing out all hidden variables (cf. Chapter 2) is significantly intractable: it is ♯P-complete [Roth, 1996]. Therefore, like in Bayesian Networks, sampling inference methods have to be applied. The most used ones are Markov Chain Monte Carlo algorithms (MCMC) [Gilks et al., 1996], which have been implemented in the Alchemy open source software [Kok et al., 2005]. The number of possible worlds, however, is exponential in the size of introduced constants independent of the inference method. To reduce this number, in these implementations it is possible to specify domains for predicates.

4.3.2 Advantages and disadvantages of Markov logic networks

A great advantage of Markov logic networks is the possibility to specify interrelations of random variables. It means that the edges of corresponding Markov network graphs are undirected. In addition, Markov logic networks allow the graphical representation of cyclic dependencies. Since Markov logic applies first-order logic as modeling language, it is possible to handle expressive assertions about objects and their relations.

Disadvantages evolve by considering ontology engineering and the complexity of computing inference: In the formalism of Markov logic networks consistency checks are not possible and MCMC algorithms applied to more expressive *MLN*s probably are exponential in time or even will not terminate. Another disadvantage is that formula weights are counterintuitive. Their influence is not obvious. To get a slightly understanding of them, they have to be related to all other formula weights.

5 The P-$\mathcal{SHOQ}(D)$ style of modeling

Description Logics (DLs) [Baader et al., 2003b] in most cases are decidable fragments of first-order logic. Some of these languages are very expressive, though, and – in contrast to first-order logic – DLs provide well understood means to establish ontologies. Therefore they are also used as representation languages for the Semantic Web [Baader et al., 2005].

Many formalisms extending DLs with probabilistic knowledge representation already have been investigated (see [Predoiu, 2008] for an overwiev). In this chapter, the P-$\mathcal{SHOQ}(D)$ [Giugno and Lukasiewicz, 2002a] formalism is introduced, which allows for a combination of the very expressive description logic $\mathcal{SHOQ}(D)$ and probabilities. Later this formalism was adapted to $\mathcal{SHIF}(D)$ and $\mathcal{SHOIN}(D)$ in [Lukasiewicz, 2008] and implemented for $\mathcal{SHIQ}(D)$ in [Näth, 2007] [Näth and Möller, 2008] [Klinov, 2008] [Klinov and Parsia, 2008a]. Summarized in a nutshell the formalism extends the description logic syntax by a construct called conditional constraint, its semantics by a probabilistic interpretation on possible worlds and decision problems are solved by finding solutions

[3] The worlds $\vec{x_1}$ and $\vec{x_2}$ have been chosen according to the intended irreflexivity and symmetry of the relation denoted by *adjacent*.

to linear programs and standard satisfiability tests with respect to the underlying description logic. They allow for representation of terminological and assertional probabilistic knowledge as well as default knowledge on concepts as a special case.

5.1 Syntax

Since the P-$\mathcal{SHOQ}(D)$ formalism relies on the description logic $\mathcal{SHOQ}(D)$, the syntax of this logic and DLs in general is introduced first.

The vocabulary of description logic languages consists of **concepts**, **roles** and **constants**. Concepts denote sets of objects, roles binary relationship between objects and constants a specific individual. **Atomic concepts** (atomic roles) are distinguished from **complex concepts** or concept descriptions (resp. complex roles), which are composed of atomic concepts and concept constructors (resp. atomic roles and role constructors). Every string of a concept description itself being a concept is called a subconcept.

In order to understand the syntax of $\mathcal{SHOQ}(D)$, the well known propositionally closed description logic $\mathcal{ALC}$ [Schmidt-Schauß and Smolka, 1991] is considered first. Let A be an atomic concept and R an atomic role. Then, the set of $\mathcal{ALC}$ concepts denoted by C or D is inductively defined with

$$C, D \longrightarrow \top \mid \bot \mid A \mid \neg A \mid C \sqcap D \mid C \sqcup D \mid \neg C \mid \forall R.C \mid \exists R.C,$$

i.e. besides atomic concepts, $\mathcal{ALC}$ concepts are composed of the logical constants $\top$ and $\bot$, concept conjunction ($C \sqcap D$), concept disjunction ($C \sqcup D$), concept negation ($\neg C$), value restrictions ($\forall R.C$) and existential restrictions ($\exists R.C$). For better reading, (sub-)concepts can be written in parantheses. A **DL knowledge base** $KB = (\mathcal{T}, \mathcal{A})$ consists of a (generalized) terminology $\mathcal{T}$ called TBox and and an assertional component $\mathcal{A}$ called ABox. $\mathcal{T}$ is a set of axioms of the form $C \sqsubseteq D$ referred to as generalized concept inclusions (GCIs) and if a and b are constants denoting objects of the domain, $\mathcal{A}$ is a set of assertions $C(a)$ or $R(a, b)$ (called concept assertions resp. role assertions).

The very expressive DL $\mathcal{SHOQ}(D)$ enlarges $\mathcal{ALC}$-TBoxes with transitive roles $Trans(R)$ (resulting in the DL denoted with $\mathcal{S}$) and with role hierarchies $R_1 \sqsubseteq R_2$ (denoted with $\mathcal{H}$). Further, in this language it is possible to specify concepts with nominals $\{a\}$ (denoted with $\mathcal{O}$)[4], qualified number restrictions ($\geq nR.C$), ($\leq nR.C$) (denoted with $\mathcal{Q}$) and concrete domains like strings and integers (denoted with D).

Besides the syntactic constructs used in DLs, the P-$\mathcal{SHOQ}(D)$ sytle of formalisms introduce conditional constraints which have the form $(D|C)[l, u]$. D, C are defined as DL concept expressions and l, u are reals from the closed interval $[0, 1]$ with $l \leq u$. To gain the ability to store such statements in a knowledge base it has to be extended to a probabilistic knowledge base $\mathcal{PKB}$. With the extension the individuals o in Δ are separated into two disjoint sets I_C and I_P where I_C is the set of *classical individuals* and I_P the set of *probabilistic individuals*. I_P has to be finite. Now the parts of a $\mathcal{PKB}$ are defined. A PTBox $\mathcal{PT}$ is DL knowledge base $\mathcal{T}$ and a finite set of conditional constraints $\mathcal{P}$. Note that the conditional constraints in $\mathcal{P}$ represent the terminological probabilistic knowledge. A PABox $\mathcal{P}_o$ is a set of conditional constraints associated with an individual o from the set I_P. Conditional constraints in a PABox have the restricted form $(D|\top)[l, u]$. There is one PABox $\mathcal{P}_o$ for each individual $o \in I_P$. Note that these sets of conditional constraints $\mathcal{P}_o$ represent the assertional probabilistic knowledge. In DLs with nominals the use of probabilists, indiP-$\mathcal{SHOQ}(D)$ sytle of oncepts is disallowed.

5.2 Semantics

In order to understand the semantics of $\mathcal{SHOQ}(D)$ see [Giugno and Lukasiewicz, 2002a]. Informally, the sematics of conditonal constraints $(D|C)[l, u]$ in a PTBox can be described as "generally, if $o : C$ holds, then $o : D$ holds with a probability between between l and u for every randomly chosen individual o" [Lukasiewicz, 2008]. Whereas the conditional constraints $(D|\top)[l, u]$ in a PABox are interpreted as a concrete "individual $o \in I_P$ is an instance of the concept D with a probability

[4]Note that with this kind of expressivity constants a can also be specified in $\mathcal{T}$

in [l,u]" [Giugno and Lukasiewicz, 2002a]. The semantics of the formalisms are defined in terms of a possible worlds with respect to the DL concept vocabulary Φ used in the conditional constraints. A *world* W is a set of concepts taken from Φ such that $\{a : C|C \in W\} \cup \{a : \neg C|C \in \Phi \setminus W\}$ is satisfiable for a new individual a. The set of all possible worlds relative to Φ is called W_Φ. A world W models a DL axiom T from $\mathcal{T}$ iff $T \cup \{a : C|C \in W\} \cup \{a : \neg C|C \in \Phi \setminus W\}$ is satisfiable for a new individual a. Furthermore a world W is model of a DL knowledge base $\mathcal{T}$ if all of its DL axioms T are satisfiable in the previous manner. In [Lukasiewicz, 2008] compatibility with standard DL semantics is proven.

Now a probabilistic interpretation $\mathcal{P}r$ is defined on the possible worlds W_Φ as a probability function: $\mathcal{P}r : W_\Phi \rightarrow [0, 1]$ and $\sum_{W \in W_\Phi} \mathcal{P}r(W) = 1$. With the probabilistic interpretation $\mathcal{P}r$ at hand the probability of a concept C, represented by $\mathcal{P}r(C)$, is defined as sum of all $\mathcal{P}r(W)$ where $W \models C$. The probabilistic interpretation of a conditional probability $\mathcal{P}r(D|C)$ is given as $\frac{\mathcal{P}r(C \sqcap D)}{\mathcal{P}r(C)}$ where $\mathcal{P}r(C) > 0$.

A conditional constraint $(D|C)[l, u]$ is *satisfied* by $\mathcal{P}r$ or $\mathcal{P}r$ *models* $(D|C)[l, u]$ if and only if $\mathcal{P}r(D|C) \in [l, u]$ or $\mathcal{P}r(C) = 0$. We will write this as $\mathcal{P}r \models (D|C)[l, u]$. A set $\mathcal{F}$ consisting of DL axioms and conditional constraints, where F denotes the elements of $\mathcal{F}$, is satisfied or modeled by $\mathcal{P}r$ if and only if $\mathcal{P}r \models F$ for all $F \in \mathcal{F}$.

The *verification* of a conditional constraint $(D|C)[l, u]$ is defined as $\mathcal{P}r(C) = 1$ and $\mathcal{P}r$ has to be a model of $(D|C)[l, u]$. We also may say $\mathcal{P}r$ *verifies* the conditional constraint $(D|C)[l, u]$. On the contrary the *falsification* of a conditional constraint $(D|C)[l, u]$ is given if and only if $\mathcal{P}r(C) = 1$ and $\mathcal{P}r$ does **not** satisfy $(D|C)[l, u]$. It is also said that $\mathcal{P}r$ *falsifies* $(D|C)[l, u]$.

Further a conditional constraint F is said to be *tolerated* under a DL knowledge base $\mathcal{T}$ and a set of conditional constraints $\mathcal{D}$ if and only if a probabilistic interpretation $\mathcal{P}r$ can be found that verifies F and $\mathcal{P}r \models \mathcal{T} \cup \mathcal{D}$.

With all these definitions at hand we are now prepared to define the *z-partition* of a set of conditional constraints $\mathcal{P}$. The z-partition is build as ordered partition $(\mathcal{P}_0, \ldots, \mathcal{P}_k)$ of $\mathcal{P}$, where each part $\mathcal{P}_i$ with $i \in \{0, \ldots, k\}$ is the set of all conditional constraints $F \in \mathcal{P} \setminus (\mathcal{P}_0 \cup \cdots \cup \mathcal{P}_{i-1})$, that are tolerated under the DL knowledge base $\mathcal{T}$ and $\mathcal{P} \setminus (\mathcal{P}_0 \cup \cdots \cup \mathcal{P}_{i-1})$.

If the z-partition can be build from a PTBox $\mathcal{PT} = (\mathcal{T}, \mathcal{P})$, it is said to be *consistent*. A probabilistic knowledge base $\mathcal{PKB} = (\mathcal{PT}, (\mathcal{P}_o)_{o \in I_p})$ is *consistent* if and only if $\mathcal{PT}$ is consistent and $\mathcal{P}r \models \mathcal{T} \cup \mathcal{P}_o$ for all $o \in I_p$. We use the z-partition for the definition of the lexicographic order on the probabilistic interpretations $\mathcal{P}r$ as follows:

A probabilistic interpretation $\mathcal{P}r$ is called *lexicographical preferred* to a probabilistic interpretation $\mathcal{P}r'$ if and only if some $i \in \{0, \ldots, k\}$ can be found, that $|\{F \in \mathcal{P}_i \mid \mathcal{P}r \models F\}| > |\{F \in \mathcal{P}_i \mid \mathcal{P}r' \models F\}|$ and $|\{F \in \mathcal{P}_j \mid \mathcal{P}r \models F\}| = |\{F \in \mathcal{P}_j \mid \mathcal{P}r' \models F\}|$ for all $i < j \leq k$.

We say a probabilistic interpretation $\mathcal{P}r$ of a set $\mathcal{F}$ of DL axioms and conditional constraints is a *lexicographically minimal model* of $\mathcal{F}$ if and only if no probabilistic interpretation $\mathcal{P}r'$ is lexicographical preferred to $\mathcal{P}r$.

By now the meaning of *lexicographic entailment* for conditional constraints from a set $\mathcal{F}$ of DL axioms and conditional constraints under a PTBox $\mathcal{PT}$ is given as:

A conditional constraint $(D|C)[l, u]$ is a *lexicographic consequence* of a set $\mathcal{F}$ of DL axioms and conditional constraints under a PTBox $\mathcal{PT}$, written as $\mathcal{F} \mid\!\sim (D|C)[l, u]$ under $\mathcal{PT}$, if and only if $\mathcal{P}r(D) \in [l, u]$ for every lexicographically minimal model $\mathcal{P}r$ of $\mathcal{F} \cup \{(C|\top)[1, 1]\}$. *Tight lexicographic consequence* of $\mathcal{F}$ under $\mathcal{PT}$ is defined as $\mathcal{F} \mid\!\sim_{tight} (D|C)[l, u]$ if and only if l is the infimum and u is the supremum of $\mathcal{P}r(D)$. We define $l = 1$ and $u = 0$ if **no** such probabilistic interpretation $\mathcal{P}r$ exists.

Finally we define lexicographic entailment using a probabilistic knowledge base $\mathcal{PKB}$ for terminological and assertional conditional constraints F.

If F is a terminological conditional constraint, then it is said to be a lexicographic consequence of $\mathcal{PKB}$, written $\mathcal{PKB} \mid\!\sim F$ if and only if $\emptyset \mid\!\sim F$ under $\mathcal{PT}$ and a tight lexicographic consequence of $\mathcal{PKB}$, written $\mathcal{PKB} \mid\!\sim_{tight} F$ if and only if $\emptyset \mid\!\sim_{tight} F$ under $\mathcal{PT}$.

If F is an assertional conditional constraint for $o \in I_P$, then it is said to be a lexicographic consequence of $\mathcal{PKB}$, written $\mathcal{PKB} \mid\!\sim F$, if and only if $\mathcal{P}_o \mid\!\sim F$ under $\mathcal{PT}$ and a tight lexicographic consequence of $\mathcal{PKB}$, written $\mathcal{PKB} \mid\!\sim_{tight} F$ if and only if $\mathcal{P}_o \mid\!\sim_{tight} F$ under $\mathcal{PT}$.

$$\sum_{(W\in W_\Phi),W\models\neg D\sqcap C} -l y_W + \sum_{(W\in W_\Phi),r\models D\sqcap C} (1-l)y_W \geq 0 \quad \text{(for all } (D|C)[l,u]\in\mathcal{F}) \tag{21a}$$

$$\sum_{(W\in W_\Phi),W\models\neg D\sqcap C} u y_W + \sum_{(W\in W_\Phi),W\models D\sqcap C} (u-1)y_W \geq 0 \quad \text{(for all } (D|C)[l,u]\in\mathcal{F}) \tag{21b}$$

$$\sum_{(W\in W_\Phi)} y_W = 1 \tag{21c}$$

$$y_W \geq 0 \quad \text{(for all } W\in W_\Phi) \tag{21d}$$

Figure 11: Constraints of the linear program

5.3 Decision problems

For these formalisms the following interesting decision problems have been introduced:

- **Probabilistic TBox consistency**: Given a PTBox $\mathcal{PT} = (\mathcal{T}, \mathcal{P})$ decide if it is consistent. This involves checking the consistency of $\mathcal{T}$ and if a z-partition can be build form $\mathcal{P}$ with respect to $\mathcal{T}$.
- **Probabilistic KB consistency**: Given a $\mathcal{PKB} = (\mathcal{PT}, (\mathcal{P}_o)_{o\in I_p})$ decide if it is consistent. This involves checking the consistency of $\mathcal{PT}$ and for each PABox $\mathcal{P}_o$ its satisfiablility with respect to $\mathcal{T}$.
- **Probabilistic lexicographic entailment**: Given $\mathcal{PT}$ and a finite set of conditional constraints $\mathcal{F}$ and a conditional constraint $(D|C)[?,?]$ with unknown bounds determine the tightest lexicographic bounds. This set $\mathcal{F}$ is the empty set $\emptyset$ in case of terminological queries or it contains the relevant PAbox $\mathcal{P}_o$ for a specific $o \in I_P$ in case of assertional queries.

The stated decision problems can be broken down into two subproblems probabilisitic satisfiability and tight logical entailment. These are then solved by satisfiability test against the underlying DL and linear programming.

5.4 Inference

In order to decide probabilistic satisfiability the first objective is to build a set of all possible worlds W_Φ. It contains the worlds W, which we obtain by a mapping r of the conditional constraints $F_i = (D_i|C_i)[l_i, u_i]$, elements of a set of conditional constraints $\mathcal{F}$, onto one of the following terms $D_i \sqcap C_i$, $\neg D_i \sqcap C_i$ or $\neg C_i$ under the condition, that the intersection of our terms is not equal to the bottom concept given a consistent DL KB $\mathcal{T}$, written $\mathcal{T} \not\models r(F_1) \sqcap \cdots \sqcap r(F_n) \sqsubseteq \bot$. In the following we will write W instead of $r(F_1) \sqcap \cdots \sqcap r(F_n)$ as an abbreviation.

With W_Φ at hand we are able to set up linear programs to decide the satisfiability of the DL KB $\mathcal{T}$ and a finite set of conditional constraints $\mathcal{F}$. The constraints of the linear program are displayed in Figure 11. We say that $\mathcal{T} \cup \mathcal{F}$ is satisfiable if and only if the linear program with the constraints 21a-d is solvable for variables y_W, where $W \in W_\Phi$. This means that in the objective function all coefficients preceding the variables y_W are set to 1. We further need to introduce the meaning of $W \models C$ which is used as index of the summation in 21a and 21b. It is an abbreviation for $\emptyset \models W \sqsubseteq C$. So the coefficient preceding the variables y_W is set in linear constraints 21a and 21b if either $W \models \neg D \sqcap C$ or $W \models D \sqcap C$ may be proven.

Why is the creation of linear programs reasonable? Consider the following: By definition a conditional constraint is satisfied if $u \geq \mathcal{P}r(D|C) \geq l \Leftrightarrow u\mathcal{P}r(C) \geq \mathcal{P}r(D \sqcap C) \geq l\mathcal{P}r(C)$. This may lead us to linear constraints 21a and 21b. Lets focus on the upper bound, whose derivation is displayed in Figure 12. The derivation for the lower bound 21a follows analogously. The linear constraints 21c and 21d reflect that all $\mathcal{P}r(W)$ have to sum up to 1 and all $\mathcal{P}r(W) \in [0, 1]$

With the tool at hand to decide satisfiability, we may also decide if a conditional constraint may be tolerated by a set of conditional constraints $\mathcal{F}$. To verify a constraint we add a conditional

$$u \sum_{W \in W_\Phi, W \models C} y_W \geq \sum_{W \in W_\Phi, W \models D \sqcap C} y_W \Leftrightarrow \quad (22a)$$

$$u \sum_{W \in W_\Phi, W \models (\neg D \sqcap C) \sqcup (D \sqcap C)} y_W \geq \sum_{W \in W_\Phi, W \models D \sqcap C} y_W \Leftrightarrow \quad (22b)$$

$$u \sum_{W \in W_\Phi, W \models \neg D \sqcap C} y_W + u \sum_{W \in W_\Phi, W \models D \sqcap C} y_W \geq \sum_{W \in W_\Phi, W \models D \sqcap C} y_W \Leftrightarrow \quad (22c)$$

$$\sum_{W \in W_\Phi, W \models \neg D \sqcap C} u y_W + \sum_{W \in W_\Phi, W \models D \sqcap C} (u-1) y_W \geq 0 \quad (22d)$$

Figure 12: Upper bound derivation

constraint $(C|\top)[1,1]$. With the extended set the linear program is generated and solved. If an unfeasible solution is computed the conditional constraint is conflicting. If an optimal solution is found, the conditional constraint is tolerated. Now the z-partition of a set of conditional constraints is computable.

How to compute tight probability bounds for given evidence C and conclusion D in respect to a set of conditional constraints $\mathcal{F}$ under a DL KB $\mathcal{T}$? The task is named *tight logical entailment* and denoted $\mathcal{T} \cup \mathcal{F} \models_{tight} (D|C)[l,u]$. Given that $\mathcal{T} \cup \mathcal{F}$ is satisfiable, a linear program is set up for $\mathcal{F} \cup (C|\top)[1,1] \cup (D|\top)[0,1]$. The objective function is set to $\sum_{W \in W_\Phi, W \models D} y_W$. So the coefficient in front of the variables y_W are set 1 if $W \models D$. The tight logical entailed lower bound l is computed by minimising, respectively the upper bound u by maximising the linear program.

In order to compute tight probabilistic lexicographic entailment for given evidence C and conclusion D under a $\mathcal{PKB}$ the following steps have to be taken:

1. Compute the z-partition of $\mathcal{P}$ in order to be able to generate a lexicographic ordering
2. Compute lexicographic minimal sets $\mathcal{P}'$ of conditional constraints of $\mathcal{P}$ as elements of $\overline{\mathcal{P}}$.
3. Compute the tight logical entailment $\mathcal{T} \cup \mathcal{F} \cup \mathcal{P}' \models_{tight} (D|C)[l,u]$ for all $\mathcal{P}' \in \overline{\mathcal{P}}$.
4. Select the minimum of all computed lower bounds and the maximum of all upper bounds.

The 2. step needs some explanation since a new task "compute lexicographic minimal sets" is introduced. In order to define a lexicographic minimal set $\mathcal{D}'$, a preparatory definition is required. A set $\mathcal{P}' \subset \mathcal{P}$ lexicographic preferable to $\mathcal{P}'' \subset \mathcal{P}$ if and only if some $i \in \{0, \ldots, k\}$ exists such that $|\mathcal{P}' \cap \mathcal{P}_i| > |\mathcal{P}'' \cap \mathcal{P}_i|$ and $|\mathcal{P}' \cap \mathcal{P}_i| > |\mathcal{P}'' \cap \mathcal{P}_i|$ for all $i < j \leq k$. With the lexicographic order introduced onto the sets $\mathcal{P}'$ the definition of lexicographic minimal is given as: $\mathcal{P}'$ is lexicographic minimal in $\overline{\mathcal{P}} \subseteq \{\mathcal{P}' | \mathcal{P}' \subseteq \mathcal{P}\}$ if and only if $\mathcal{P}' \in \overline{\mathcal{P}}$ and no $\mathcal{P}'' \in \overline{\mathcal{P}}$ is lexicographic preferable to $\mathcal{P}'$.

5.5 Example

Lets have a look at an example to get an intuition how the formalisms work. The small scenario describes the knowledge which we have on penguins and birds as a small part of the environmental domain. We know for sure that *Penguins* are subset of *Birds*. Additionally we know with atleast probability 0.95 that birds have *Wings*, *Birds* can *Fly* with high probability [0.9, 0.95] and *Penguins* can *Fly* with low probability $[0, 0.01]$. Furthermore we know of the individual $p1$ which we believe to be a penguin within the interval $[0.7, 0.8]$. This knowledge is captured in the $\mathcal{PKB}$ in Figure 13. The first decision problem to look at is checking probabilistic TBox consistency. With the $\mathcal{T}$ part consistent. The next task is building a z-partion of the 3 conditional constraints. In order to do this the possible worlds W_Φ have to be generated. They are shown in Figure 14. Each of these worlds is associated with the vari-

$$
\begin{aligned}
\mathcal{PKB} = (&\{Penguin \sqsubset Bird\},\\
&\{(\forall has.Wings|Bird)[0.95, 1]\\
&(\forall can.Fly|Bird)[0.9, 0.95],\\
&(\forall can.Fly|Penguin)[0, 0.01]\}\\
&\{(Penguin|\top)[0.7, 0.8]\}_{p1})
\end{aligned}
$$

Figure 13: Small scenario

$$
\begin{aligned}
W_\Phi = \{&\forall has.Wings \sqcap \forall can.Fly \sqcap Bird \sqcap Penguin,\\
&\neg\forall has.Wings \sqcap \forall can.Fly \sqcap Bird \sqcap Penguin,\\
&\forall has.Wings \sqcap \neg\forall can.Fly \sqcap Bird \sqcap Penguin,\\
&\neg\forall has.Wings \sqcap \neg\forall can.Fly \sqcap Bird \sqcap Penguin,\\
&\forall has.Wings \sqcap \forall can.Fly \sqcap Bird \sqcap \neg Penguin,\\
&\neg\forall has.Wings \sqcap \forall can.Fly \sqcap Bird \sqcap \neg Penguin,\\
&\forall has.Wings \sqcap \neg\forall can.Fly \sqcap Bird \sqcap \neg Penguin,\\
&\neg\forall has.Wings \sqcap \neg\forall can.Fly \sqcap Bird \sqcap \neg Penguin,\\
&\neg Bird \sqcap \neg Penguin\}
\end{aligned}
$$

Figure 14: Worlds

ables within the linear programs which have to be solved to compute the z-partion. The resulting z-partion has two Parts ($\mathcal{P}_0 = \{(\forall has.Wings|Bird)[0.95, 1], (\forall can.Fly|Bird)[0.9, 0.95]\}, \mathcal{P}_1 = \{(\forall can.Fly|Penguin)[0, 0.01]\}$). Thus the PTBox is consistent. In order to handle the second decision problem additonally the probabilistic satisfiability of $\{(Penguin|\top)[0.7, 0.8]\}_{p1}$ with respect to $\mathcal{T}$ has to be determined. This is the case therefore the $\mathcal{PKB}$ is consistent.

Lets have a look at some queries for probabilistic lexicographic entailment. With the $\mathcal{PKB}$ above the following interesting queries can be answered as follows:

$$
\begin{aligned}
\{\} &\mid\!\sim (\forall has.Wings \sqcap \forall can.Fly|Bird)[0.85, 0.95]\\
\{\} &\mid\!\sim (\forall has.Wings \sqcap \forall can.Fly|Penguin)[0.0, 0.1]\\
\{\} &\mid\!\sim (Penguin|Bird)[0.0, 0.1]\\
\{\} &\mid\!\sim (Bird|Penguin)[1, 1]\\
\{(Penguin|\top)[0.7, 0.8]\} &\mid\!\sim (\forall can.Fly|\top)[0.0, 0.307]\\
\{(Penguin|\top)[0.7, 0.8]\} &\mid\!\sim (\forall has.Wings|\top)[0.665, 1]
\end{aligned}
$$

5.6 Advantages and disadvantages of P-$\mathcal{SHOQ}(D)$ style of modeling

These formalisms are build on top of well known description logics. This has the advantage that reasoning tools can be developed on top of a DL Reasoner treating it as a Black Box. Furthermore this might also allow for an easy extension of already developed KB as it might seem. Some guidelines on how to develop a $\mathcal{PKB}$ are described in [Klinov and Parsia, 2008b]. Despite all the advantages these formalisms come with some limitations which are highlighted here. The introduction of "cyclic" conditional constraints, e.g. two constraints of the form $(D|C)[l, u]$, $(C|D)[l, u]$ with $0 < l \leq u < 1$, render a probabilistic knowledge base inconsistent. One can not decide which of the two constraints is lexicographic preferred therefore a z-partition can not be build. Furthermore in these formalisms it is possible to specify believes about the individuals in I_P however there are no relations possible between individuals from I_P. This means that probabilistic individuals

can only specify islands of believe as opposed to believes on a relational structure of individuals.

6 Conclusion and Outlook

In this deliverable we have introduced three important probabilistic approaches with formal semantics, namely Bayesian networks, Markov logic networks and the P-$\mathcal{SHOQ}(D)$ formalism.

We outlined that graphs of Bayesian networks do not consist of cycles or undirected arcs and, thus, modeling of mutual dependencies in Bayesian networks might result in conditional probability tables which are hard to understand by humans. Thus, machine learning techniques would have to be used right from the beginning.

With Markovian formalisms, uncertainty information is specified as weights for formulas. Weights are considered to be rather intuitive for humans. The weights are automatically transformed into internal values (potentials), which are used for probabilistic reasoning. The representation of relational structures is given in terms of predicate logics.

With P-$\mathcal{SHOQ}(D)$ another approach for representing first-order structures is given. However, it is not possible to sufficient specify uncertainty with respect to relational structures among individuals. Since Markov logic networks are not affected by these limitations, we conclude that this formalism is more suitable for CASAM, although the Markovian approaches found in the literature either use Horn logic (inexpressive) or do not care about decidability issues at all. To overcome some of these disadvantages, the approach in CASAM should be to combine Description Logics and the Markov network formalism.

References

[Baader et al., 2003a] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. F., editors (2003a). *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.

[Baader et al., 2003b] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F. (January 2003b). *The Description Logic Handbook: Theory, Implementation and Application*. Cambridge UP: Cambridge, NY.

[Baader et al., 2005] Baader, F., Horrocks, I., and Sattler, U. (2005). *Description logics as ontology languages for the Semantic Web Mechanizing*, pages 228–248. LNAI 2605, Springer. http://www.cs.man.ac.uk/~horrocks/Publications/download/2003/BaHS03.pdf.

[Baader et al., 2003c] Baader, F., Küsters, R., and Wolter, F. (2003c). Extensions to description logics. In [Baader et al., 2003a], chapter 6, pages 219–261.

[Bacchus, 1990] Bacchus, F. (1990). *Representing and reasoning with probabilistic knowledge: A logical approach to probabilities*. The MIT Press, Cambridge.

[Brachman and Levesque, 2004] Brachman, R. J. and Levesque, H. J. (2004). *Knowledge Representation and Reasoning*. San Francisco, CA: Morgan Kaufmann.

[Castano et al., 2008] Castano, S., Espinosa, S., Ferrara, A., Karkaletsis, V., Kaya, A., Möller, R., Montanelli, S., Petasis, G., and Wessel, M. (2008). Multimedia interpretation for dynamic ontology evolution. In *Journal of Logic and Computation*. Oxford University Press.

[Charniak and Goldman, 1991] Charniak, E. and Goldman, R. (1991). Probabilistic abduction for plan recognition. Technical report, Brown University, Providence, RI, USA.

[Colucci et al., 2004] Colucci, S., Noia, T. D., Sciascio, E. D., Mongiello, M., and Donini, F. M. (2004). Concept abduction and contraction for semantic-based discovery of matches and negotiation spaces in an e-marketplace. In *ICEC '04: Proceedings of the 6th international conference on Electronic commerce*, pages 41–50, New York, NY, USA. ACM Press.

[Ding and Peng, 2004] Ding, Z. and Peng, Y. (2004). A probabilistic extension to ontology language OWL. In *Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS)*.

[Ding et al., 2005] Ding, Z., Peng, Y., and Pan, R. (2005). BayesOWL: Uncertainty modeling in semantic web ontologies. In *Soft Computing in Ontologies and Semantic Web*. Springer.

[Domingos and Richardson, 2007] Domingos, P. and Richardson, M. (2007). *Markov Logic: A Unifying Framework for Statistical Relational Learning*, pages 339–371. Cambridge, MA: MIT Press.

[Dürig and Studer, 2005] Dürig, M. and Studer, T. (2005). Probabilistic abox reasoning: Preliminary results. In *Proc. Int. Description Logics Workshop 2005*, pages 104–111.

[Espinosa et al., 2008] Espinosa, S., Kaya, A., Melzer, S., and Möller, R. (2008). On ontology based abduction for text interpretation. In Gelbukh, A., editor, *Proc. of 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2008)*, number 4919 in LNCS, pages 194–205. Springer.

[Espinosa et al., 2007] Espinosa, S., Kaya, A., Melzer, S., Möller, R., and Wessel, M. (2007). Towards a media interpretation framework for the semantic web. In *Proc. of the 2007 IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, pages 374–380. IEEE Computer Society.

[Fitting, 1996] Fitting, M. (1996). *First-order logic and automated theorem proving*. New York: Springer-Verlag.

[Flach and Kakas, 2000] Flach, P. and Kakas, A., editors (2000). *Abduction and Induction: Essays on their relation and integration*. Kluwer Academic Publishers.

[Fuhr, 1995] Fuhr, N. (1995). Probabilistic Datalog: a logic for powerful retrieval methods. In *Proceedings of SIGIR-95: 18th ACM International Conference on Research and Development in Information Retrieval*, pages 282–290.

[Fuhr, 2000] Fuhr, N. (2000). Probabilistic datalog: Implementing logical information retrieval for advanced applications. *Journal of the American Society of Information Science*, 51(2):95–110.

[Getoor et al., 2001] Getoor, L., Friedman, N., Koller, D., and Pfeffer, A. (2001). Learning probabilisitc relational models. In Dzeroski, S. and Lavrac, N., editors, *Relational Data Mining*. Springer.

[Gilks et al., 1996] Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London, UK.

[Giugno and Lukasiewicz, 2002b] Giugno, R. and Lukasiewicz, T. (2002b). P-SHOQ(D): A probabilistic extension of SHOQ(D) for probabilistic ontologies in the semantic web. In *JELIA '02: Proceedings of the European Conference on Logics in Artificial Intelligence*, pages 86–97. Springer-Verlag.

[Giugno and Lukasiewicz, 2002a] Giugno, R. and Lukasiewicz, T. (Cosenza, Italy, September 2002a). P-shoq(d): A probabilistic extension of shoq(d) for probabilistic ontologies in the semantic web. In Flesca, S., Greco, S., Leone, N., and Ianni, G., editors, *Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA 2002)*, pages 86–97. Volume 2424 of Lecture Notes in Computer Science, Springer.

[Halpern, 1990] Halpern, J. (1990). An analysis fo first-order logics of probability. *Artificial Intelligence*, 46(3):311–350.

[Heinsohn, 1994] Heinsohn, J. (1994). Probabilistic description logics. In de Mantaras, R. L. and Poole, D., editors, *Proc. of the 10th Conf. on Uncertainty in Artificial Intelligence*, pages 311–318, Seattle, Washington. Morgan Kaufmann.

[Hobbs et al., 1993] Hobbs, J. R., Stickel, M., Appelt, D., and Martin, P. (1993). Interpretation as abduction. *Artificial Intelligence*, 63:69–142.

[Jaeger, 1994] Jaeger, M. (1994). Probabilistic reasoning in terminological logics. In *Proc. of the 4th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'94)*, pages 305–316.

[Kakas and Denecker, 2002] Kakas, A. and Denecker, M. (2002). Abduction in logic programming. In Kakas, A. and Sadri, F., editors, *Computational Logic: Logic Programming and Beyond. Part I*, number 2407 in LNAI, pages 402–436. Springer.

[Kakas et al., 1992] Kakas, A. C., Kowalski, R. A., and Toni, F. (1992). Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770.

[Klinov, 2008] Klinov, P. (2008). Pronto: A non-monotonic probabilistic description logic reasoner. In *ESWC*, pages 822–826.

[Klinov and Parsia, 2008a] Klinov, P. and Parsia, B. (2008a). Optimization and evaluation of reasoning in probabilistic description logic: Towards a systematic approach. In *International Semantic Web Conference*, pages 213–228.

[Klinov and Parsia, 2008b] Klinov, P. and Parsia, B. (2008b). Probabilistic modeling and OWL: A user oriented introduction to $\mathcal{SHIQ}(\mathbf{D})$. In *OWLED*.

[Kok and Domingos, 2005] Kok, S. and Domingos, P. (Bonn, Germany, 2005). Learning the structure of markov logic networks. In *Proceedings of the Twenty-Second International Conference on Machine Learning*, pages 441–448. ACM Press.

[Kok et al., 2005] Kok, S., Singla, P., Richardson, M., and Domingos, P. (2005). The alchemy system for statistical relational ai. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA.

[Koller et al., 1997] Koller, D., Levy, A., and Pfeffer, A. (1997). P-CLASSIC: A tractable probabilistic description logic. In *Proc. of the 14th Nat. Conf. on Artificial Intelligence (AAAI'97)*, pages 390–397. AAAI Press/The MIT Press.

[Koller et al., 2007] Koller, D., Levy, A., and Pfeffer, A. (2007). *Graphical Models in a Nutshell*, pages 13–55. Cambridge, MA: MIT Press.

[Koller and Pfeffer, 1998] Koller, D. and Pfeffer, A. (1998). Probabilistic frame-based systems. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI), Madison, Wisconsin.*

[Lukasiewicz, 2005a] Lukasiewicz, T. (2005a). Probabilistic description logic programs. In *Proc. of ECSQARU*, pages 737–749.

[Lukasiewicz, 2005b] Lukasiewicz, T. (2005b). Stratified probabilistic description logic programs. In *Proc. of ISWC-URSW*, pages 87–97.

[Lukasiewicz, 2008] Lukasiewicz, T. (2008). Expressive probabilistic description logics. *Artif. Intell.*, 172(6-7):852–883.

[Näth, 2007] Näth, T. H. (Februar 2007). Analysis of the average-case behavior of an inference algorithm for probabilistic description logics. Technical report, TU Hamburg-Harburg. http://www.sts.tu-harburg.de/pw-and-m-theses/2007/naet07.pdf.

[Näth and Möller, 2008] Näth, T. H. and Möller, R. (2008). Contrabovemrufum: A system for probabilistic lexicographic entailment. In *Description Logics.*

[Nilsson, 1986] Nilsson, N. (1986). Probabilistic logic. *Artificial Intelligence*, 28:71–87.

[Nottelmann and Fuhr, 2001] Nottelmann, H. and Fuhr, N. (2001). Learning probabilistic datalog rules for information classification and transformation. In *In Proceedings CIKM*, pages 387–394.

[Nottelmann and Fuhr, 2004] Nottelmann, H. and Fuhr, N. (2004). pDAML+OIL: A probabilistic extension to DAML+OIL based on probabilistic datalog. In *Proceedings Information Processing and Management of Uncertainty in Knowledge-Based Systems.*

[Nottelmann and Fuhr, 2006] Nottelmann, H. and Fuhr, N. (2006). Adding probabilities and rules to OWL Lite subsets based on probabilistic datalog. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 14(1):17–41.

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

[Pfeffer et al., 1999] Pfeffer, A., Koller, D., Milch, B., and Takusagawa, K. (1999). SPOOK: A system for probabilistic object-oriented knowledge representation. In *Proc. of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 541–550.

[Poole, 1992] Poole, D. (1992). Logic programming, abduction and probability. In *Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS'92),*, pages 530–538.

[Poole, 1993a] Poole, D. (1993a). Logic programming, abduction and probability: a top-down anytime algorithm for estimating prior and posterior probabilities. *New Generation Computing*, 11(3-4):377–400.

[Poole, 1993b] Poole, D. (1993b). Probabilistic horn abduction and bayesian networks. *AIJ*, 64(1):81–129.

[Poole, 2003] Poole, D. (2003). First-order probabilistic inference. In *Proc. International Joint Conference on Artificial Intelligence IJCAI-03*, pages 985–991.

[Predoiu, 2008] Predoiu, L. (2008). Probabilistic models for the semantic web - a survey. http://ki.informatik.uni-mannheim.de/fileadmin/publication/Predoiu08Survey.pdf.

[Roth, 1996] Roth, D. (1996). On the hardness of approximate reasoning. *Artificial Intelligence, 82*, 1:273–302.

[Russell and Norvig, 2003] Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach (Second Edition)*. Prentice Hall.

[Schmidt-Schauß and Smolka, 1991] Schmidt-Schauß, M. and Smolka, G. (1991). Attributive concept descriptions with complements. *Artif. Intell.*, 48(1):1–26.

[Sebastiani, 1994] Sebastiani, F. (1994). A Probabilistic Terminological Logic for Modelling Information Retrieval. In Croft, W. and Rijsbergen, C. v., editors, *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 122–130, Dublin, Ireland. Springer-Verlag.

[Shanahan, 2005] Shanahan, M. (2005). Perception as abduction: Turning sensor data into meaningful representation. *Cognitive Science*, 29:103–134.

[Smyth and Poole, 2004] Smyth, C. and Poole, D. (2004). Qualitative probabilistic matching with hierarchical descriptions. In *Proc. Knowledge Representation and Reasoning (KR&R 2004)*.

[Taskar et al., 2007] Taskar, B., Abbeel, P., Wong, M.-F., and Koller, D. (2007). *Relational Markov Networks*, pages 175–199. Cambridge, MA: MIT Press.

[Yelland, 2000] Yelland, P. Y. (2000). An alternative combination of Baysian networks and description logics. In *Proc. of the 7th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2000)*, pages 225–234.