Log-Linear Description Logics

Mathias Niepert and **Jan Noessner** and **Heiner Stuckenschmidt**
KR & KM Research Group
Universität Mannheim
Mannheim, Germany

Abstract

Log-linear description logics are a family of probabilistic logics integrating various concepts and methods from the areas of knowledge representation and reasoning and statistical relational AI. We define the syntax and semantics of log-linear description logics, describe a convenient representation as sets of first-order formulas, and discuss computational and algorithmic aspects of probabilistic queries in the language. The paper concludes with an experimental evaluation of an implementation of a log-linear DL reasoner.

1 Introduction

Numerous real-world problems require the ability to handle both deterministic and uncertain knowledge. Due to differences in epistemological commitments made by their particular AI communities, researchers have mostly been concerned with either one of these two types of knowledge. While the representation of purely logical knowledge has been the focus of knowledge representation and reasoning, reasoning about knowledge in the presence of uncertainty has been the major research theme of the machine learning and uncertainty in AI communities. Nevertheless, there have been some attempts to combine the diverse concepts and methods by several researchers. The resulting approaches include probabilistic description logics [Jaeger, 1994; Koller *et al.*, 1997; Lukasiewicz and Straccia, 2008] and statistical relational languages [Getoor and Taskar, 2007] to name but a few. While the former have mainly been studied on a theoretical level, statistical relational languages have been proven useful for a number of practical applications such as data integration [Niepert *et al.*, 2010] and NLP [Riedel and Meza-Ruiz, 2008]. A line of work that has regained importance with the use of semantic web languages is the combination of probabilistic models with description logics [Lukasiewicz and Straccia, 2008]. Probabilistic description logic programs [Lukasiewicz, 2007], for instance, combine description logic programs under the answer set and well-founded semantics with independent choice logic [Poole, 2008].

In this paper, we represent and reason about uncertain knowledge by combining log-linear models [Koller and Friedman, 2009] and description logics [Baader *et al.*, 2003]. Log-linear models allow us to incorporate both probabilistic and deterministic dependencies between description logic axioms. The ability to integrate heterogeneous features make log-linear models a commonly used parameterization in areas such as NLP and bioinformatics and a number of sophisticated algorithms for inference and parameter learning have been developed. In addition, log-linear models form the basis of some statistical relational languages such as Markov logic [Richardson and Domingos, 2006].

The logical component of the presented theory is based on description logics for which consequence-driven reasoning is possible [Krötzsch, 2010; Baader *et al.*, 2005]. We focus particularly on the description logic $\mathcal{EL}^{++}$ which captures the expressivity of numerous ontologies in the medical and biological sciences and other domains. $\mathcal{EL}^{++}$ is also the description logic on which the web ontology language profile OWL 2 EL is based [Baader *et al.*, 2005]. Reasoning services such as consistency and instance checking can be performed in polynomial time. It is possible to express disjointness of complex concept descriptions as well as range and domain restrictions [Baader *et al.*, 2008]. In addition, role inclusion axioms (RIs) allow the expression of role hierarchies $r \sqsubseteq s$ and transitive roles $r \circ r \sqsubseteq r$.

In real-world applications, uncertainty occurs often in form of *degrees of confidence* or *trust*. The semantic web community, for instance, has developed numerous data mining algorithms to generate confidence values for description logic axioms with ontology learning and matching being two prime applications. Most of these confidence values have no clearly defined semantics. Confidence values based on lexical similarity measures, for instance, are in widespread use while more sophisticated algorithms that generate actual probabilities make often naïve assumptions about the dependencies of the underlying probability distribution. Hence, formalisms and inference procedures are needed that incorporate degrees of confidence in order to represent uncertain axioms and to compute answers to probabilistic queries while utilizing the logical concepts of *coherency* and *consistency*.

To respond to this need, we introduce log-linear description logics as a novel formalism for combining deterministic and uncertain knowledge. We describe a convenient representation of log-linear description logics that allows us to adapt existing concepts and algorithms from statistical relational AI to answer standard probabilistic queries. In particular, we for-

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
RI	$r_1 \circ ... \circ r_k \sqsubseteq r$	$r_1^{\mathcal{I}} \circ ... \circ r_k^{\mathcal{I}} \subseteq r^{\mathcal{I}}$

Table 1: The DL $\mathcal{EL}^{++}$ without concrete domains.

mulate maximum a-posteriori queries and present an efficient algorithm that computes most probable *coherent* models, a reasoning service not supported by previous probabilistic description logics. We conclude the paper with an experimental evaluation of ELOG, our implementation of a log-linear description logic reasoner.

2 Description Logics

Description logics (DLs) are a family of formal knowledge representation languages. They provide the logical formalism for ontologies and the Semantic Web. We focus on the particular DL $\mathcal{EL}^{++}$ *without concrete domains*, henceforth denoted as $\mathcal{EL}^{++}$ [Baader *et al.*, 2005]. We conjecture that the presented ideas are also applicable to other description logics for which materialization calculi exist [Krötzsch, 2010].

Concept and role descriptions in $\mathcal{EL}^{++}$ are defined recursively beginning with a set $\mathsf{N_C}$ of concept names, a set $\mathsf{N_R}$ of role names, and a set $\mathsf{N_I}$ of individual names, and are built with the constructors depicted in the column "Syntax" of Table 1. We write a and b to denote individual names; r and s to denote role names; and C and D to denote concept descriptions. The semantics are defined in terms of an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is the non-empty *domain* of the interpretation and $\cdot^{\mathcal{I}}$ is the interpretation function which assigns to every $A \in \mathsf{N_C}$ a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, to every $r \in \mathsf{N_R}$ a relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to every $a \in \mathsf{N_I}$ an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

A constraint box (CBox) is a finite set of general concept inclusion (GCIs) and role inclusion (RIs) axioms. Given a CBox $\mathcal{C}$, we use $\mathsf{BC}_{\mathcal{C}}$ to denote the set of *basic concept descriptions*, that is, the smallest set of concept descriptions consisting of the top concept $\top$, all concept names used in $\mathcal{C}$, and all nominals $\{a\}$ appearing in $\mathcal{C}$. A CBox $\mathcal{C}$ is in normal form if all GCIs have one of the following forms, where $C_1, C_2 \in \mathsf{BC}_{\mathcal{C}}$ and $D \in \mathsf{BC}_{\mathcal{C}} \cup \{\bot\}$:

$$C_1 \sqsubseteq D; \qquad C_1 \sqsubseteq \exists r.C_2;$$
$$C_1 \sqcap C_2 \sqsubseteq D; \quad \exists r.C_1 \sqsubseteq D;$$

and if all role inclusions are of the form $r \sqsubseteq s$ or $r_1 \circ r_2 \sqsubseteq s$. By applying a finite set of rules and introducing new concept and role names, any CBox $\mathcal{C}$ can be turned into a normalized CBox of size polynomial in $\mathcal{C}$. For any $\mathcal{EL}^{++}$ CBox $\mathcal{C}$ we write $\mathsf{norm}(\mathcal{C})$ to denote the set of normalized axioms that result from the application of the normalization rules to $\mathcal{C}$.

An interpretation $\mathcal{I}$ *satisfies* an axiom c if the condition in the column "Semantics" in Table 1 holds for that axiom. An interpretation $\mathcal{I}$ is a *model* of a CBox $\mathcal{C}$ if it satisfies every axiom in $\mathcal{C}$. A concept C *is subsumed by* a concept D with respect to a CBox $\mathcal{C}$, written $C \sqsubseteq_{\mathcal{C}} D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in every model of $\mathcal{C}$. A normalized CBox is *classified* when subsumption relationships between *all* concept names are made explicit. A CBox $\mathcal{C}$ is *coherent* if for all concept names C in $\mathcal{C}$ we have that $C \not\sqsubseteq_{\mathcal{C}} \bot$. For every axiom c and every set of axioms $\mathcal{C}'$, we write $\mathcal{C} \models c$ if every model of $\mathcal{C}$ is also a model of $\{c\}$ and we write $\mathcal{C} \models \mathcal{C}'$ if $\mathcal{C} \models c'$ for every $c' \in \mathcal{C}'$. For a finite set $\mathsf{N_U} \subseteq \mathsf{N_C} \cup \mathsf{N_R}$ of concept and role names the *set of all normalized axioms constructible from* $\mathsf{N_U}$ is the union of (a) all normalized GCIs constructible from concept and role names in $\mathsf{N_U}$ and the top and bottom concepts; and (b) all normalized RIs constructible from role names in $\mathsf{N_U}$.

3 Log-Linear Models

Log-linear models are parameterizations of undirected graphical models (Markov networks) which are central to the areas of reasoning under uncertainty and statistical relational learning [Koller and Friedman, 2009; Getoor and Taskar, 2007]. A Markov network $\mathcal{M}$ is an undirected graph whose nodes represent a set of random variables $\{X_1, ..., X_n\}$ and whose edges model direct probabilistic interactions between adjacent nodes. A distribution P is a log-linear model over a Markov network $\mathcal{M}$ if it is associated with:

- a set of features $\{f_1(D_1), ..., f_k(D_k)\}$, where each D_i is a clique in $\mathcal{M}$ and each f_i is a function from D_i to $\mathbb{R}$,
- a set of real-valued weights $w_1, ..., w_k$, such that

$$P(X_1, ..., X_n) = \frac{1}{Z} \exp\left(\sum_{i=1}^{k} w_i f_i(D_i)\right),$$

where Z is a normalization constant.

Arguably one of the more successful statistical relational languages is Markov logic [Richardson and Domingos, 2006] which can be seen as a first-order template language for log-linear models with binary variables. A Markov logic network is a finite set of pairs $(F_i, w_i), 1 \leq i \leq k$, where each F_i is a first-order formula and each w_i a real-valued weight associated with F_i. With a finite set of constants C it defines a log-linear model over possible worlds. A *possible world* $\mathbf{x}$ is a truth assignments to ground atoms (predicates without variables) with respect to C. Each variable $X_j, 1 \leq j \leq n$, corresponds to a ground atom and feature f_i corresponds to the number of true groundings of F_i in possible world $\mathbf{x}$.

4 Log-Linear Description Logics

Log-linear description logics integrate description logics with probabilistic log-linear models. While the syntax of log-linear description logics is that of the underlying description logic, it is possible (but not necessary) to assign real-valued weights to axioms. The semantics is defined by a log-linear probability distribution over *coherent* ontologies. In the remainder of the paper, we focus on the log-linear description logic $\mathcal{EL}^{++}$ without nominals and concrete domains[1] which we denote as $\mathcal{EL}^{++}$-LL.

[1] We conjecture that the theory is extendable to capture nominals and concrete domains. However, we leave this to future work.

4.1 Syntax

The syntax of log-linear description logics is equivalent to the syntax of the underlying description logic except that it is possible to assign weights to GCIs and RIs. More formally, a $\mathcal{EL}^{++}$-LL CBox $\mathcal{C} = (\mathcal{C}^{\mathsf{D}}, \mathcal{C}^{\mathsf{U}})$ is a pair consisting of a *deterministic* $\mathcal{EL}^{++}$ CBox $\mathcal{C}^{\mathsf{D}}$ and an *uncertain* CBox $\mathcal{C}^{\mathsf{U}} = \{(c, w_c)\}$ which is a set of pairs (c, w_c) with each c being a $\mathcal{EL}^{++}$ axiom and w a real-valued weight assigned to c. Given a $\mathcal{EL}^{++}$-LL CBox $\mathcal{C}$ we use $\mathsf{BC}_{\mathcal{C}}$ to denote the set of basic concept descriptions occurring in $\mathcal{C}^{\mathsf{D}}$ or $\mathcal{C}^{\mathsf{U}}$.

While the *deterministic* CBox contains axioms that are known to be true, the *uncertain* CBox contains axioms for which we only have a *degree of confidence*. Intuitively, the greater the weight of an axiom the more likely it is true. Every axiom can either be part of the deterministic or the uncertain CBox but not both. The deterministic CBox is assumed to be coherent.

4.2 Semantics

The semantics of log-linear DLs is based on joint probability distributions over *coherent* $\mathcal{EL}^{++}$ CBoxes. The weights of the axioms determine the log-linear probability distribution. For a $\mathcal{EL}^{++}$-LL CBox $(\mathcal{C}^{\mathsf{D}}, \mathcal{C}^{\mathsf{U}})$ and a CBox $\mathcal{C}'$ over the same set of basic concept descriptions and role names, we have that

$$P(\mathcal{C}') = \begin{cases} \frac{1}{Z} \exp\left(\sum_{\{(c,w_c) \in \mathcal{C}^{\mathsf{U}} : \mathcal{C}' \models c\}} w_c\right) & \text{if } \mathcal{C}' \text{ is } \textit{coherent} \\ & \text{and } \mathcal{C}' \models \mathcal{C}^{\mathsf{D}}; \\ 0 & \text{otherwise} \end{cases}$$

where Z is the normalization constant of the log-linear probability distribution P. The semantics of the log-linear description logic $\mathcal{EL}^{++}$-LL leads to exactly the probability distributions one would expect under the open world semantics of description logics as demonstrated by the following example.

Example 1. *Let* $\mathsf{BC}_{\mathcal{C}} = \{C, D\}$*, let* $\mathcal{C}^{\mathsf{D}} = \emptyset$*, and let* $\mathcal{C}^{\mathsf{U}} = \{\langle C \sqsubseteq D, 0.5\rangle, \langle C \sqcap D \sqsubseteq \bot, 0.5\rangle\}$*. Then*[2]*,* $P(\{C \sqsubseteq D, C \sqcap D \sqsubseteq \bot\}) = 0$*,* $P(\{C \sqsubseteq D\}) = Z^{-1}\exp(0.5)$*,* $P(\{C \sqsubseteq D, D \sqsubseteq C\}) = Z^{-1}\exp(0.5)$*,* $P(\{C \sqcap D \sqsubseteq \bot\}) = Z^{-1}\exp(0.5)$*,* $P(\{D \sqsubseteq C\}) = Z^{-1}\exp(0)$*, and* $P(\emptyset) = Z^{-1}\exp(0)$ *with* $Z = 3\exp(0.5) + 2\exp(0)$*.*

4.3 Herbrand Representation

In order to formalize probabilistic queries for log-linear description logics we represent $\mathcal{EL}^{++}$-LL CBoxes as sets of first-order sentences modeling the uncertain and deterministic axioms. We begin by defining some basic concepts.

Definition 2. *Let* P *be a set of predicate symbols and* C *be a set of constant symbols. An atom is an expression of the form* $p(t_1, ..., t_n)$ *where* $p \in P$ *and* $t_1, ..., t_n$ *are variables or constant symbols. A literal is an atom or its negation. Let* $\mathcal{S}$ *be a set of first-order formulas built inductively from the literals, with universally quantified variables only. A grounding of a formula is obtained by replacing each variable with a constant symbol in* C*. The Herbrand base of* $\mathcal{S}$ *with respect to* C *is the set of ground atoms whose predicate symbols occur in* $\mathcal{S}$*. Each subset of the Herbrand base is a* Herbrand interpretation *specifying which ground atoms are true. A Herbrand interpretation* H *is a* Herbrand model *of* $\mathcal{S}$*, written as* $\models_H \mathcal{S}$*, if and only if it satisfies all groundings of formulas in* $\mathcal{S}$*.*

[2]We omit trivial axioms that are present in every classified CBox such as $C \sqsubseteq \top$ and $C \sqsubseteq C$.

F_1	$\forall c : sub(c, c)$
F_2	$\forall c : sub(c, \top)$
F_3	$\forall c, c', d : sub(c, c') \wedge sub(c', d) \Rightarrow sub(c, d)$
F_4	$\forall c, c_1, c_2, d : sub(c, c_1) \wedge sub(c, c_2) \wedge int(c_1, c_2, d) \Rightarrow sub(c, d)$
F_5	$\forall c, c', r, d : sub(c, c') \wedge rsup(c', r, d) \Rightarrow rsup(c, r, d)$
F_6	$\forall c, r, d, d', e : rsup(c, r, d) \wedge sub(d, d') \wedge rsub(d', r, e) \Rightarrow sub(c, e)$
F_7	$\forall c, r, d, s : rsup(c, r, d) \wedge psub(r, s) \Rightarrow rsup(c, s, d)$
F_8	$\forall c, r_1, r_2, r_3, d, e : rsup(c, r_1, d) \wedge rsup(d, r_2, e) \wedge pcom(r_1, r_2, r_3) \Rightarrow rsup(c, r_3, e)$
F_9	$\forall c : \neg sub(c, \bot)$

Table 2: The set of first-order formulas $\mathcal{F}$. Groundings of the formulas have to be compatible with the types of the predicates specified in Definition 3. $\bot$ and $\top$ are constant symbols representing the bottom and top concept.

The set of formulas $\mathcal{F}$ listed in Table 2 is partially derived from the $\mathcal{EL}^{++}$ completion rules [Baader *et al.*, 2005]. We are now in the position to define a bijective function φ that, given a finite set of concept and role names $\mathsf{N_U}$, maps each *normalized* $\mathcal{EL}^{++}$-LL CBox over $\mathsf{N_U}$ to a subset of the Herbrand base of $\mathcal{F}$ with respect to $\mathsf{N_U}$.

Definition 3 (CBox Mapping). *Let* $\mathsf{N_C}$ *and* $\mathsf{N_R}$ *be sets of concept and role names and let* $\mathsf{N_U} \subseteq \mathsf{N_C} \cup \mathsf{N_R}$ *be a finite set. Let* $\mathcal{T}$ *be the set of normalized* $\mathcal{EL}^{++}$ *axioms constructible from* $\mathsf{N_U}$*. Moreover, let* $\mathcal{H}$ *be the Herbrand base of* $\mathcal{F}$ *with respect to* $\mathsf{N_U}$*. The function* $\varphi : \wp(\mathcal{T}) \rightarrow \wp(\mathcal{H})$ *maps normalized CBoxes to subsets of* $\mathcal{H}$ *as follows:* $(\varphi(\mathcal{C}) = \bigcup_{c \in \mathcal{C}} \varphi(c))$

$$\begin{array}{lcl} C_1 \sqsubseteq D & \mapsto & sub(C_1, D) \\ C_1 \sqcap C_2 \sqsubseteq D & \mapsto & int(C_1, C_2, D) \\ C_1 \sqsubseteq \exists r.C_2 & \mapsto & rsup(C_1, r, C_2) \\ \exists r.C_1 \sqsubseteq D & \mapsto & rsub(C_1, r, D) \\ r \sqsubseteq s & \mapsto & psub(r, s) \\ r_1 \circ r_2 \sqsubseteq r_3 & \mapsto & pcom(r_1, r_2, r_3). \end{array}$$

All predicates are typed meaning that $r, s, r_i, (1 \leq i \leq 3)$, are role names, C_1, C_2 basic concept descriptions, and D basic concept descriptions or the bottom concept.

We now prove that, relative to a finite set of concept and role names, the function φ induces a one-to-one correspondence between Herbrand models of the first-order theory $\mathcal{F}$ and coherent and classified $\mathcal{EL}^{++}$ CBoxes.

Lemma 4. *Let* $\mathsf{N_C}$ *and* $\mathsf{N_R}$ *be sets of concept and role names and let* $\mathsf{N_U} \subseteq \mathsf{N_C} \cup \mathsf{N_R}$ *be a finite set. Let* $\mathcal{T}$ *be the set of normalized* $\mathcal{EL}^{++}$ *axioms constructible from* $\mathsf{N_U}$ *and let* $\mathcal{H}$ *be the Herbrand base of* $\mathcal{F}$ *with respect to* $\mathsf{N_U}$*. Then,*

(a) for any $\mathcal{C} \subseteq \mathcal{T}$ *we have that if* $\mathcal{C}$ *is classified and coherent then* $\varphi(\mathcal{C})$ *is a Herbrand model of* $\mathcal{F}$*; and*

(b) for any $H \subseteq \mathcal{H}$ *we have that if* H *is a Herbrand model of* $\mathcal{F}$ *then* $\varphi^{-1}(H)$ *is a classified and coherent CBox.*

From Lemma 4 we know that, relative to a finite set $\mathsf{N_U}$ of concept and role names, each normalized CBox over $\mathsf{N_U}$ that is classified and coherent, corresponds to exactly one Herbrand model of $\mathcal{F}$. We extend the normalization of $\mathcal{EL}^{++}$ CBoxes to $\mathcal{EL}^{++}$-LL CBoxes as follows.

Definition 5. *Let $\mathcal{C} = (\mathcal{C}^{\mathsf{D}}, \mathcal{C}^{\mathsf{U}})$ be a $\mathcal{EL}^{++}$-LL CBox. Then,* $\mathsf{norm_{LL}}(\mathcal{C}) = \mathsf{norm}(\mathcal{C}^{\mathsf{D}}) \cup \bigcup_{(c,w_c)\in\mathcal{C}^{\mathsf{U}}} \mathsf{norm}(\{c\})$.

Lemma 4 provides the justification for constructing the logical representation of a $\mathcal{EL}^{++}$-LL CBox as follows: $\mathcal{G}$ is a set of *weighted* ground formulas carrying the uncertain information and it is derived from the axioms in the uncertain CBox $\mathcal{C}^{\mathsf{U}}$ as follows. For every pair $(c, w_c) \in \mathcal{C}^{\mathsf{U}}$ we add the conjunction of ground atoms

$$\bigwedge_{\mathsf{g}\in\varphi(\mathsf{norm}(\{c\}))} \mathsf{g}$$

to $\mathcal{G}$ with weight w_c. The set $\mathcal{K}$ is constructed analogously from the deterministic CBox $\mathcal{C}^{\mathsf{D}}$ except that we do not associate weights with the ground formulas.[3]

Example 6. *Let $\mathcal{C}^{\mathsf{D}} = \{C \sqsubseteq D\}$ and $\mathcal{C}^{\mathsf{U}} = \{\langle B \sqcap C \sqcap D \sqsubseteq E, 0.5\rangle\}$. Then,* $\mathsf{norm_{LL}}(\mathcal{C}) = \mathsf{norm}(\{C \sqsubseteq D\}) \cup \mathsf{norm}(\{B \sqcap C \sqcap D \sqsubseteq E\}) = \{C \sqsubseteq D, C \sqcap D \sqsubseteq A, B \sqcap A \sqsubseteq E\}$ *with A a new concept name. We have that* $\varphi(\mathsf{norm}(\{C \sqsubseteq D\})) = \{sub(C, D)\}$ *and* $\varphi(\mathsf{norm}(\{B \sqcap C \sqcap D \sqsubseteq E\})) = \{int(C, D, A), int(B, A, E)\}$. *Hence, we add $sub(C, D)$ to $\mathcal{K}$ and $int(C, D, A) \wedge int(B, A, E)$ with weight 0.5 to $\mathcal{G}$.*

4.4 Maximum A-Posteriori Inference

Under the given syntax and semantics the first central inference task is the maximum a-posteriori (MAP) query: *"Given a $\mathcal{EL}^{++}$-LL CBox, what is a most probable coherent $\mathcal{EL}^{++}$ CBox over the same concept and role names?"* In the context of probabilistic description logics, the MAP query is crucial as it infers a *most probable* classical ontology from a probabilistic one. The MAP query also captures two important problems that frequently occur in the context of the Semantic Web: Ontology learning and ontology matching.

The following theorem combines the previous results and formulates the $\mathcal{EL}^{++}$-LL MAP query as a maximization problem subject to a set of logical constraints.

Theorem 7. *Let $\mathcal{C} = (\mathcal{C}^{\mathsf{D}}, \mathcal{C}^{\mathsf{U}})$ be a $\mathcal{EL}^{++}$-LL CBox, let $\mathsf{N_U}$ be the set of concept and role names used in* $\mathsf{norm_{LL}}(\mathcal{C})$*, and let $\mathcal{H}$ be the Herbrand base of $\mathcal{F}$ with respect to $\mathsf{N_U}$. Moreover, let $\mathcal{K}$ be the set of ground formulas constructed from $\mathcal{C}^{\mathsf{D}}$ and let $\mathcal{G}$ be the set of weighted ground formulas constructed from $\mathcal{C}^{\mathsf{U}}$. Then, with*

$$\hat{H} := \operatorname*{arg\,max}_{\{H\subseteq\mathcal{H}:\, \models_H(\mathcal{K}\cup\mathcal{F})\}} \sum_{\{(G,w_G)\in\mathcal{G}:\, \models_H G\}} w_G \qquad (1)$$

we have that $\varphi^{-1}(\hat{H})$ is a most probable coherent *CBox over $\mathsf{N_U}$ that entails $\mathcal{C}^{\mathsf{D}}$.*

By showing that every partial weighted MAX-SAT problem can be reduced to an instance of the $\mathcal{EL}^{++}$ MAP query problem of Theorem 7 in polynomial time and by invoking the complexity results for the former problem [Creignou *et al.*, 2001] we can prove the following theorem.

Theorem 8. *The maximum a-posteriori problem for $\mathcal{EL}^{++}$-LL CBoxes is NP-hard and APX-complete.*

Conversely, we can solve a MAP query by reducing it to a (partial) weighted MAX-SAT instance with a set of clauses polynomial in the number of axioms *and* basic concept descriptions. There exist several sophisticated algorithms for weighted MAX-SAT. We developed ELOG[4], an efficient $\mathcal{EL}^{++}$-LL reasoner based on integer linear programming (ILP). Let $\mathcal{F}^{\mathsf{N_U}}$ be the set of all groundings of formulas in $\mathcal{F}$ with respect to the set of constant symbols $\mathsf{N_U}$. For each ground atom g_i occurring in a formula in $\mathcal{G}, \mathcal{K}$, or $\mathcal{F}^{\mathsf{N_U}}$ we associate a unique variable $x_i \in \{0, 1\}$. Let C_j^G be the set of indices of ground atoms in sentence $G_j \in \mathcal{G}$, let C_j^K be the set of indices of ground atoms in sentence $K_j \in \mathcal{K}$, and let $C_j^F(\bar{C}_j^F)$ be the set of indices of unnegated (negated) ground atoms of sentence $F_j \in \mathcal{F}^{\mathsf{N_U}}$ in clausal form. With each formula $G_j \in \mathcal{G}$ with weight w_j (without loss of generality we assume non-negative weights) we associate a unique variable $z_j \in \{0, 1\}$. Then, the ILP is stated as follows

$$\max \sum_{j=1}^{|\mathcal{G}|} w_j z_j \qquad \text{subject to}$$

$$\sum_{i\in C_j^G} x_i \geq |C_j^G| z_j,\ \forall j \quad (2) \text{ and } \sum_{i\in C_j^K} x_i \geq |C_j^K|,\ \forall j \quad (3)$$

$$\text{and } \sum_{i\in C_j^F} x_i + \sum_{i\in \bar{C}_j^F} (1 - x_i) \geq 1,\ \forall j \quad (4).$$

Every state of the variables x_i of a solution for the ILP corresponds (via the function φ) to a most probable classified and coherent CBox over $\mathsf{N_U}$ that entails $\mathcal{C}^{\mathsf{D}}$. Adding all constraints of type (4) at once, however, would lead to a large and potentially intractable optimization problem. Therefore, we employ a variant of the cutting plane inference algorithm first proposed in the context of Markov logic [Riedel, 2008]. We first solve the optimization problem *without* constraints of type (4). Given a solution to the partial problem we determine in polynomial time those constraints of type (4) that are violated by this solution, add those to the formulation, and solve the updated problem. This is repeated until no violated constraints remain. Please note that the MAP query derives a most probable CBox *and* classifies it at the same time.

4.5 Conditional Probability Inference

The second type of inference task is the conditional probability query: *"Given a $\mathcal{EL}^{++}$-LL CBox, what is the probability of a conjunction of axioms?"* More formally, given a $\mathcal{EL}^{++}$-LL CBox $\mathcal{C}$ and a set $\mathcal{C}^{\mathsf{Q}}$ of normalized $\mathcal{EL}^{++}$-LL axioms over the same concept and role names, the *conditional probability query* is given by $P(\mathcal{C}^{\mathsf{Q}} \mid \mathcal{C}) = \sum_{\{\mathcal{C}':\mathcal{C}^{\mathsf{Q}}\subseteq\mathcal{C}'\}} P(\mathcal{C}')$ where each $\mathcal{C}'$ is a classified and normalized CBox over the same set of basic concept descriptions and role names.

[3] Note that the number of formulas in $\mathcal{G}$ and $\mathcal{K}$ is equal to the number of axioms in $\mathcal{C}^{\mathsf{U}}$ and $\mathcal{C}^{\mathsf{D}}$, respectively.

[4] http://code.google.com/p/elog-reasoner/

Algorithm 1 **MC-ILP** for $\mathcal{EL}^{++}$-LL

Input: weights w_j and variables z_j of the MAP ILP

1: $c_j \leftarrow w_j$ for all j
2: $x^{(0)} \leftarrow$ solution of ILP with all w_j set to 0
3: **for** $k \leftarrow 1$ to n **do**
4: set all z_j and all w_j to 0
5: **for all** z_j satisfied by $x^{(k-1)}$ **do**
6: with probability $1 - e^{-c_j}$ fix z_j to 1 in ILP$^{(k)}$
7: **end for**
8: **for all** z_j not fixed to 1 **do**
9: with probability 0.5 set w_j to 1 in ILP$^{(k)}$
10: **end for**
11: $x^{(k)} \leftarrow$ solution of ILP$^{(k)}$
12: **end for**

Determining the exact conditional probability is infeasible in the majority of use-cases. Hence, for the reasoner ELOG[4], we developed a Markov chain Monte Carlo (MCMC) variant similar to MC-SAT, a slice sampling MCMC algorithm [Poon and Domingos, 2006]. Simpler sampling strategies such as Gibbs sampling are inadequate due to the presence of deterministic dependencies. Poon and Domingos showed that the Markov chain generated by MC-SAT satisfies *ergodicity* and *detailed balance*. In practice, however, it is often too time-consuming to obtain uniform samples as required by MC-SAT and, thus, we loosen the uniformity requirement in favor of higher efficiency.

Algorithm 1 lists the pseudo-code of MC-ILP which is based on the previous MAP ILP except that we also add, for each query axiom not in $\mathcal{C}^{\mathsf{U}}$, a constraint of type (2) associated with a new variable z_j and weight 0. In each iteration, after fixing certain variables to 1 (line 6), we build an ILP where the coefficients of the objective function are set to 1 with probability 0.5 and remain 0 otherwise (line 9). The solution is then *as close as possible* to the uniform sample. MC-ILP samples coherent and classified CBoxes from the joint distribution and determines the probability of a conjunction of axioms as the fraction of samples in which they occur together.

5 Experiments

To complement the presented theory we also assessed the practicality of the reasoning algorithms. After all, the development of the theory was motivated primarily by the need for algorithms that, given a set of axioms with confidence values, compute a most probable *coherent* ontology. Two examples of this problem are ontology learning and ontology matching. Due to space considerations, we only discuss and evaluate the results on an instance of the former problem.

The ontology learning community has developed and applied numerous machine learning and data mining algorithms to generate confidence values for DL axioms. Most of these confidence values have no clearly defined semantics. Confidence values based on lexical similarity measures, for instance, are in widespread use while more sophisticated algorithms that generate actual probabilities make often naïve assumptions about the dependencies of the underlying probability distribution. Hence, formalisms are needed that incorporate these various types of confidence values in order to compute most probable ontologies while utilizing the logical concepts of coherency and consistency.

Axiom type	Algorithm	Precision	Recall	F_1 score
Subsumption	Greedy	0.620	0.541	0.578
	$\mathcal{EL}^{++}$-LL MAP	0.784	0.514	**0.620**
Disjointness	Greedy	0.942	0.980	0.961
	$\mathcal{EL}^{++}$-LL MAP	0.935	0.990	0.961

Figure 1: Results for weighted axioms derived from the AMT questionnaires *without* additional known axioms.

Axiom type	Algorithm	Precision	Recall	F_1 score
Subsumption	Greedy	0.481	0.669	0.559
	$\mathcal{EL}^{++}$-LL MAP	0.840	0.568	**0.677**
Disjointness	Greedy	0.948	0.960	0.954
	$\mathcal{EL}^{++}$-LL MAP	0.937	0.992	0.964

Figure 2: Results for weighted axioms derived from the AMT questionnaires *with* additional known axioms.

We decided to generate confidence values using a "crowdsourcing" service. Probably the best known crowdsourcing platform is the *Amazon Mechanical Turk* (AMT)[5]. With AMT, Amazon offers numerous options for designing customized questionnaires. Due to its relatively high publicity (about 100,000 tasks were available at the time of this writing), it attracts a lot of users and consequently seems most suitable for our scenario.

For the evaluation of the different methods, we used the EKAW ontology as the gold standard. It consists of 75 classes, 33 object properties, 148 subsumption, and 2,299 disjointness axioms when materialized, and models the domain of scientific conferences. We generated HITs (human intelligence tasks) by creating questionnaires each with 10 yes/no questions. Half of these were used to generate confidence values for subsumption (disjointness) axioms. For the pair of class labels *Conference_Paper* and *Poster_Session*, for instance, the two types of yes/no questions were:

(a) Is every *Conference_Paper* also a *Poster_Session*?

(b) Can there be anything that is both a *Conference_Paper* and a *Poster_Session*?

For each pair of classes *and* for each type of question we obtained 9 responses from *different* AMT workers. The confidence value for a subsumption (disjointness) axiom was computed by dividing the number of "yes" ("no") answers by 9. We applied a threshold of 0.5, that is, only when the majority of the 9 workers answered with "yes" ("no") did we assign a confidence value to the axiom. Moreover, we halved the weights of the disjointness axioms for reasons of symmetry. This resulted in 2,507 axioms (84 subsumption and 2,423 disjointness) with confidence values. Based on these axioms we constructed two different sets. One consisting of the weighted axioms only and the other with 79 additional axioms implied by the gold standard ontology which are of

[5] http://www.mturk.com

the form $r \sqsubseteq s$ (7 axioms), $C \sqsubseteq \exists r.D$ (40 axioms), and $\exists r.C \sqsubseteq D$ (32 axioms).

We compared the $\mathcal{EL}^{++}$-LL reasoner ELOG with a greedy approach that is often employed in ontology learning scenarios. The greedy algorithm sorts the axioms in descending order according to their confidence values and adds one axiom at a time to an initially empty ontology. However, it adds an axiom only if it does not render the resulting ontology incoherent. To compute precision and recall scores we materialized all subsumption and disjointness axioms of the resulting ontology. We used the reasoner Pellet for the materialization of axioms and the coherence checks. All experiments were conducted on a desktop PC with Intel Core2 Processor P8600 with 2.4GHz and 2GB RAM. The reasoner ELOG, all ontologies, supplemental information, and the experimental results are available at `http://code.google.com/p/elog-reasoner/`.

ELOG's MAP inference algorithm needs on average 3.7 seconds and the greedy algorithm 33.6 seconds for generating the two coherent ontologies. Note that the 3.7 seconds include the time to classify the ontology. ELOG's cutting plane inference method needed 6 and 7 iterations, respectively. These results indicate that the MAP inference algorithm under the $\mathcal{EL}^{++}$-LL semantics is more efficient than the greedy approach for small to medium sized ontologies. Figures 1 and 2 depict the recall, precision, and F_1 scores for both algorithms. The F_1 score for the ontologies computed using $\mathcal{EL}^{++}$-LL MAP inference is, respectively, 5% and 12% higher than for the one created using the greedy algorithm. Furthermore, while the addition of known axioms leads to an F_1 score increase of about 5% under the $\mathcal{EL}^{++}$-LL semantics it decreases for the greedy algorithm.

The MC-ILP algorithm needs between 0.4 and 0.6 seconds to compute one MCMC sample for either $\mathcal{EL}^{++}$-LL CBox. We find this to be an encouraging result providing further evidence that MCMC inference can be applied efficiently to compute conditional probabilities for conjunctions of axioms over real-world $\mathcal{EL}^{++}$-LL ontologies.

6 Conclusion and Future Work

With this paper we introduced log-linear description logics as a family of probabilistic logics integrating several concepts from the areas of knowledge representation and reasoning and statistical relational AI. The combination of description logics and log-linear models allows one to compute probabilistic queries efficiently with algorithms originally developed for typical linear optimization problems. We focused on the description logic $\mathcal{EL}^{++}$ without nominals and concrete domains but we believe that the theory is extendable to other DLs for which materialization calculi exist.

Future work will be concerned with the inclusion of nominals and concrete domains, negated axioms in the deterministic CBox, and the design of algorithms for creating more compact representations of classified CBoxes.

Acknowledgments

Many thanks to Johanna Völker, Anne Schlicht, and the anonymous IJCAI reviewers for their valuable feedback.

References

[Baader *et al.*, 2003] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proceedings of IJCAI*, 2005.

[Baader *et al.*, 2008] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope further. In *Proceedings of OWLED*, 2008.

[Creignou *et al.*, 2001] Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity classifications of boolean constraint satisfaction problems*. SIAM, 2001.

[Getoor and Taskar, 2007] Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[Jaeger, 1994] M. Jaeger. Probabilistic reasoning in terminological logics. In *Proceedings of KR*, 1994.

[Koller and Friedman, 2009] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[Koller *et al.*, 1997] Daphne Koller, Alon Levy, and Avi Pfeffer. P-classic: A tractable probabilistic description logic. In *Proceedings of AAAI*, 1997.

[Krötzsch, 2010] Markus Krötzsch. Efficient inferencing for OWL EL. In *Proceedings of JELIA*, 2010.

[Lukasiewicz and Straccia, 2008] Thomas Lukasiewicz and Umberto Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *J. of Web Sem.*, 6, 2008.

[Lukasiewicz, 2007] Thomas Lukasiewicz. Probabilistic description logic programs. *Int. J. Appr. Reas.*, 45, 2007.

[Niepert *et al.*, 2010] M. Niepert, C. Meilicke, and H. Stuckenschmidt. A Probabilistic-Logical Framework for Ontology Matching. In *Proceedings of AAAI*, 2010.

[Poole, 2008] David Poole. The independent choice logic and beyond. In *Probabilistic inductive logic programming*. Springer-Verlag, 2008.

[Poon and Domingos, 2006] Hoifung Poon and Pedro Domingos. Sound and efficient inference with probabilistic and deterministic dependencies. In *Proceedings of AAAI*, 2006.

[Richardson and Domingos, 2006] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2), 2006.

[Riedel and Meza-Ruiz, 2008] Sebastian Riedel and Ivan Meza-Ruiz. Collective semantic role labelling with markov logic. In *Proceedings of CoNLL*, 2008.

[Riedel, 2008] Sebastian Riedel. Improving the accuracy and efficiency of map inference for markov logic. In *Proceedings of UAI*, 2008.