Loopy Propagation in a Probabilistic Description Logic

Fabio Gagliardi Cozman and Rodrigo Bellizia Polastro

Escola Politécnica, Universidade de São Paulo - São Paulo, SP - Brazil
fgcozman@usp.br, rodrigopolastro@gmail.com

Abstract. This paper introduces a probabilistic description logic that adds probabilistic inclusions to the popular logic $\mathcal{ALC}$, and derives inference algorithms for inference in the logic. The probabilistic logic, referred to as CR$\mathcal{ALC}$ ("credal" $\mathcal{ALC}$), combines the usual acyclicity condition with a Markov condition; in this context, inference is equated with calculation of (bounds on) posterior probability in relational credal/Bayesian networks. As exact inference does not seem scalable due to the presence of quantifiers, we present first-order loopy propagation methods that seem to behave appropriately for non-trivial domain sizes.

1 Introduction

A description logic offers a formal language where one can describe concepts such as "A mother is a woman who has a child" [2]. To do so, a description logic typically uses a decidable fragment of first-order logic, and tries to reach a practical balance between expressivity and complexity. The last decade has seen a significant increase in interest in description logics as a vehicle for large-scale knowledge representation, for instance in the *semantic web* [4]. Indeed, the language OWL, proposed by the W3 consortium as the "data layer" of their archictecture for the semantic web, is an XML encoding for quite expressive description logics [21].

Description logics are not geared towards the representation of uncertainty about individuals and concepts: one cannot express that "with high probability, a bird is a flying animal". The literature contains a number of proposals that add probabilistic uncertainty to description logics, as this is central to the management of semantic data in large repositories. The goal of this paper is to contribute in such a direction.

In this paper we consider a probabilistic extension of the popular logic $\mathcal{ALC}$, where we allow *probabilistic inclusions* such as $P(\mathsf{FlyingBird}|\mathsf{Bird}) = 0.99$. Section 2 offers a brief appraisal of related work in the literature. The syntax and semantics of our proposed probabilistic logic are introduced in Section 3. A notable feature of our proposal is that we adopt an interpretation-based semantics that avoids the challenges of direct inference and lets us deal smoothly with probabilities over assertions. We then adopt a Markov condition, attached to the usual acyclicity condition of description logics, that connects the logic with the theory of relational credal/Bayesian networks. In Section 4 we discuss the inference problem for the logic, and note that exact inference does not seem to be scalable when quantified concepts are employed. Thus we derive a first-order version of loopy propagation, and show evidence of the scalability of the method even when probabilities are not uniquely specified. We briefly discuss infinite domains in Section 5.

S. Greco and T. Lukasiewicz (Eds.): SUM 2008, LNAI 5291, pp. 120–133, 2008.

2 Probabilistic Description Logics

This section reviews the literature on probabilistic description logics; some basic concepts are defined in this paragraph. Assume a vocabulary containing *individuals*, *concepts*, and *roles* [2]. Concepts and roles are combined to form new concepts using a set of *constructors*; constructors in the $\mathcal{ALC}$ logic [45] are *conjunction* ($C \sqcap D$), *disjunction* ($C \sqcup D$), *negation* ($\neg C$), *existential* restriction ($\exists r.C$), and *value* restriction ($\forall r.C$). *Concept inclusions/definitions* are denoted respectively by $C \sqsubseteq D$ and $C \equiv D$, where C and D are concepts. A set of concept inclusions and definitions is a *terminology*. Concept $(C \sqcup \neg C)$ is denoted by $\top$, and concept $(C \sqcap \neg C)$ is denoted by $\bot$ (when we use $\top$ or $\bot$, we assume them to be defined through some C that does not appear anywhere else in the terminology). If an inclusion/definition contains a concept C in its right hand side and a concept D in its left hand side, say that C *directly uses* D. Indicate the transitive closure of "directly uses" by *uses*. A terminology is *acyclic* if it is a set of concept inclusions/definitions such that no concept in the terminology uses itself [2]. Typically terminologies only allow the left hand side of a concept inclusion/definition to contain a concept name (and no constructors). Usually one is interest in *concept subsumption*: whether $C \sqsubseteq D$ for concepts C and D. A terminology may be associated to a set of *assertions* about individuals, such as $\mathsf{Fruit}(\mathsf{appleFromJohn})$ and $\mathsf{buyFrom}(\mathsf{houseBob}, \mathsf{John})$. A set of assertions $\mathcal{A}$ is called an *Abox*. An assertion $C(a)$ directly uses assertions of concepts (resp. roles) directly used by C instantiated by a (resp. by (a, b) for $b \in \mathcal{D}$), and likewise for the "uses" relation in a recursive fashion. The semantics of a description logic is almost always given by a *domain* $\mathcal{D}$ and an *interpretation* $\mathcal{I}$. The domain $\mathcal{D}$ is a nonempty set; we often assume its cardinality to be given as input. The interpretation function $\mathcal{I}$ maps each individual to an element of the domain, each concept name to a subset of the domain, each role name to a binary relation on $\mathcal{D} \times \mathcal{D}$. The interpretation function is extended to other concepts as follows: $\mathcal{I}(C \sqcap D) = \mathcal{I}(C) \cap \mathcal{I}(D)$, $\mathcal{I}(C \sqcup D) = \mathcal{I}(C) \cup \mathcal{I}(D)$, $\mathcal{I}(\neg C) = \mathcal{D} \backslash \mathcal{I}(C)$, $\mathcal{I}(\exists r.C) = \{x \in \mathcal{D} | \exists y : (x, y) \in \mathcal{I}(r) \wedge y \in \mathcal{I}(C)\}$, $\mathcal{I}(\forall r.C) = \{x \in \mathcal{D} | \forall y : (x, y) \in \mathcal{I}(r) \rightarrow y \in \mathcal{I}(C)\}$. An inclusion $C \sqsubseteq D$ is entailed iff $\mathcal{I}(C) \subseteq \mathcal{I}(D)$, and $C \equiv D$ is entailed iff $\mathcal{I}(C) = \mathcal{I}(D)$. An assertion $C(a)$ is *consistent* iff $\mathcal{I}(a) \in \mathcal{I}(C)$ for some interpretation, and likewise for $r(a, b)$; an Abox is consistent iff all its assertions are consistent at once. Logics in the literature offer significantly larger sets of features, such as numerical restrictions, role hierarchies, inverse and transitive roles (the OWL language contains several such features [21]). Most description logics have direct translations into multi-modal logics [44] and fragments of first-order logic [5] (the translation to first-order logic is particularly important here: each concept C is interpreted as a unary predicate $C(x)$; each role r is interpreted as a binary predicate $r(x, y)$; other constructs have direct translations into first-order logic, such as $\exists r.C$ to $\exists y : r(x, y) \wedge C(y)$ and $\forall r.C$ to $\forall y : r(x, y) \rightarrow C(y)$).

Several probabilistic descriptions logics have appeared in the literature. Heinsohn [20], Jaeger [24] and Sebastiani [46] consider probabilistic inclusion axioms such as $P_{\mathcal{D}}(\mathsf{Plant}) = \alpha$, meaning that a randomly selected individual is a Plant with probability α. That is, probabilities are assigned to subsets of the domain $\mathcal{D}$; this characterizes a *domain-based* semantics. Sebastiani allows assessments such as $P(\mathsf{Plant}(\mathsf{Tweety})) = \alpha$ as well, specifying probabilities over the interpretations themselves. For example one

interprets $P(\mathsf{FlyingBird}(\mathsf{Tweety})) = 0.001$ as assigning 0.001 to the probability of all interpretations where Tweety is a flying bird. This characterizes an *interpretation-based* semantics. Overall, most proposals for probabilistic description logics have adopted a domain-based semantics [13,14,18,20,24,30,33,46,52], while relatively few have adopted an interpretation-based semantics [6,46]. The difficulty with domain-based semantics is the problem of *direct inference* [31]: statistical information about the domain does not translate into information about individuals. For example, suppose we learn that $P(\mathsf{FlyingBird}) = 0.3$; a domain-based semantics takes a fixed domain and fixed interpretation and assigns 0.3 to the probability that an element of the domain is a flying bird. However, we learn nothing about $P(\mathsf{FlyingBird}(\mathsf{Tweety}))$, as the interpretation is fixed and Tweety either is a flying bird, or not. For this reason, most proposals for probabilistic description logics with a domain-based semantics simply do not handle assertions. We note that Dürig and Studer do avoid direct inference by only allowing probabilisties over assertions [14]. Lukasiewicz has proposed another strategy, where probabilities over terminologies and assertions blend through an entailment relation with nonmonotonic properties, *lexicographic entailment* [18,33]. Lukasiewicz considers probabilistic versions of very expressive description logics; his logic P-$\mathcal{SHOIN}$(D) is currently the most expressive probabilistic description logic in the literature. In this paper we prefer not to employ nonmonotonic reasoning.

The probabilistic description logics discussed so far share the property that a set of formulas may be satisfied by one *or more* probability measures (in fact, semantics based on *sets of probability measures* are often adopted by probabilistic logics [19]). Another characteristic shared by the probabilistic description logics mentioned so far is that they do not express judgements of independence. However, there has been significant effort in combining logical constructs with Bayesian and Markov networks in the last fifteen years, so as to benefit from independence judgements rather than to suffer from their complexities [17,36]. Indeed, several recent probabilistic description logics have adopted semantics based on Bayesian networks. The first logic to do so, P-CLASSIC, enlarges the logic CLASSIC with a set of Bayesian networks ("p-classes") so as to specify a single probability measure over the domain [30]. A limitation is that P-CLASSIC does not handle assertions. Some characteristics of P-CLASSIC are present in the logics proposed in this paper (acyclicity and Markov conditions); however our interest in obtaining meaningful probabilities over assertions, by resorting to interpretation-based semantics, is a major difference.

Other logics that combine terminologies with Bayesian networks are Yelland's Tiny Description Logic [52], Ding and Peng's BayesOWL language [13], and Staker's logic [49] (none can handle assertions). Costa and Laskey's PR-OWL language [6] adopts an interpretation-based semantics inherited from Multi-entity Bayesian networks (MEBNs) [7], and quite similar to the semantics used in this paper. The PR-OWL language is more expressive than ours, with less guarantees concerning inference and infinite domains; their inference algorithms are based on incremental propositionalization. Finally, most constructs in this paper can be also emulated in Nottelmann and Fuhr's probabilistic version of the OWL language, however our inference methods are completely different from theirs [40]. Besides the literature just reviewed, there is a large body of relevant work on knowledge databases [22] and on fuzzy description logics [34].

3 A Probabilistic Description Logic: CR$\mathcal{ALC}$

Probabilistic inclusions and inferences. Start with a fragment of $\mathcal{ALC}$ by discarding roles for a moment. That is, if C and D are concepts, then $\neg C$, $C \sqcap D$ and $C \sqcup D$ are concepts as well. Concept inclusions and definitions are allowed, denoted by $C \sqsubseteq D$ and $C \equiv D$ where D is a concept and C is a concept name (that is, we do not allow *general concept axioms* [2]). Now introduce *probabilistic inclusions* $P(C|D) = \alpha$, where D is a concept and C is a concept name. If D is $\top$, then we simply write $P(C) = \alpha$. We are interested in computing a *query* $P(A_0(a_0)|\mathcal{A})$ for an Abox $\mathcal{A} = \{A_j(a_j)\}_{j=1}^{M}$ (this is an *inference*).

Acyclicity. Given a probabilistic inclusion $P(C|D) = \alpha$, say that C "directly uses" B if B appears in the expression of D; again, "uses" is the transitive closure of "directly uses", and a terminology is acyclic if no concept uses itself (Section 2). We assume that every terminology is acyclic; this is in fact a common assumption for description logics [2]. The acyclicity assumption allows one to draw any terminology $\mathcal{T}$ as a directed acyclic graph $\mathcal{G}(\mathcal{T})$: each concept name is a node, and if a concept C directly uses concept D, then D is a *parent* of C in $\mathcal{G}(\mathcal{T})$.

Domain/interpretation semantics. As noted in Section 2, in a domain-based semantics we consider measures over the domain $\mathcal{D}$, and the natural interpretation for a probabilistic inclusion is $P(\text{set of } C\text{s} \mid \text{set of } D\text{s}) = \alpha$. An interpretation-based semantics instead postulates probability measures over interpretations (that is, over complete assignments of individuals to concepts: for m concepts and $|\mathcal{D}| = n$, there are $m2^n$ interpretations). The most natural interpretation-based semantics for $P(C|D) = \alpha$ seems to be (as discussed for instance by Lukasiewicz [32]):

$$\forall x : P(C(x)|D(x)) = \alpha. \tag{1}$$

We favor this interpretation-based semantics because it can smoothly interpret a query $P(A(a)|B(b))$ for concepts A and B and individuals a and b. A domain-based semantics would assign 0 or 1 to the probability $P(A(a)|B(b))$, depending on the particular fixed interpretation. That is, the semantics (1) lets us bypass direct inference (the problem of moving from probabilities over domains to probabilities for individuals).[1] Note that asserted facts must be conditioned upon: there is no contradiction between assessment $\forall x : P(C(x)) = \alpha$ and assertion $C(a)$, as we can have $P(C(a)|C(a)) = 1$ while $P(C(a)) = \alpha$.

Following Bacchus [3], we harmonize the semantics (1) and assertions such as $C(a)$ by assuming that all individuals are rigid designators (that is, an individual corresponds to the same element of the domain in all interpretations).

Adding roles: relational networks. We now introduce restrictions $\exists r.C$ and $\forall r.C$ into the logic. To simplify the presentation, without loss of generality we assume that C in

[1] As a digression, we note that the most elaborate attempt to address, rather than bypass, direct inference in probabilistic description logics are Lukasiewicz's. He uses lexicographic entailment to produce direct inference [18,33], and also tries to avoid a drawback of semantics (1): the fact that it forbids exceptions such as $P(C(a)|D(a)) < \alpha$ for some individual a.

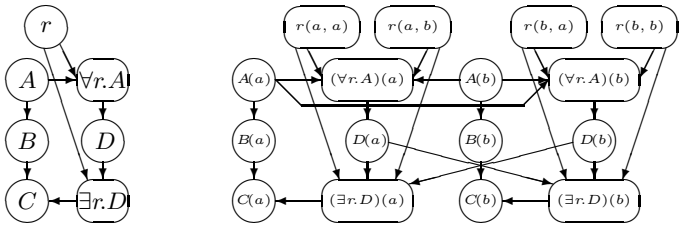

Fig. 1. Graph $\mathcal{G}(\mathcal{T}_1)$ for terminology $\mathcal{T}_1$ in Example 1, and its grounding for domain $\mathcal{D} = \{a, b\}$

these restrictions is a concept name (an auxiliary definition may specify a concept C of arbitrary complexity). As probabilistic inclusions must only have concept names in their conditioned concept, assessments such as $P(\forall r.C|D) = \alpha$ are not allowed.

Now, each restriction $\exists r.C$ and $\forall r.C$ is added as a node to the graph $\mathcal{G}(\mathcal{T})$. As each one of these restrictions directly uses r and C, the graph $\mathcal{G}(\mathcal{T})$ must contain a node for each role r, and an edge from r to each restriction directly using it. Each node $\exists r.C$ or $\forall r.C$ is a *deterministic* node in that its value is completely determined by its parents; again, we emphasize that a direct assessment such as $P(\exists r.C|D) = \alpha$ is not allowed.

Example 1. Consider a terminology $\mathcal{T}_1$ with concepts A, B, C, D. Suppose $P(A) = \alpha_1$, $B \sqsubseteq A$, $C \sqsubseteq B \sqcup \exists r.D$, $P(B|A) = \alpha_2$, $P(C|B \sqcup \exists r.D) = \alpha_3$, and $P(D|\forall r.A) = \alpha_4$. The last three assessments specify beliefs about partial overlap among concepts. Suppose also $P(D|\neg\forall r.A) = \epsilon \approx 0$ (conveying the existence of exceptions to the inclusion of D in $\forall r.A$). Figure 1 depicts $\mathcal{G}(\mathcal{T})$. □

Independence and Markov condition. Probabilistic description logics such as P-CLASSIC, BayesOWL and PR-OWL employ judgements of independence, encoded by a Markov condition, to constrain probability values (down to uniqueness) and to decompose models into small pieces. Other probabilistic logics adopt similar Markov conditions for graph-based assessments [16,36], and a few logics adopt graphs with Markov conditions and allow assessments "outside" of the graphs [1,9].

We take the position that the structure of the "directly uses" relation encodes stochastic independence through a Markov condition. First, for every concept $C \in \mathcal{T}$ and for every $x \in \mathcal{D}$, $C(x)$ is independent of every assertion that does not use $C(x)$, given assertions that directly use C. Second, for every $(x, y) \in \mathcal{D} \times \mathcal{D}$, $r(x, y)$ is independent of all other assertions, except ones that use $r(x, y)$.

The interaction between logical constructs and this Markov condition has its subtleties. For instance, if $\mathcal{G}(\mathcal{T})$ is $A \rightarrow C \leftarrow B$ because $A \sqsubseteq \neg C$ and $B \sqsubseteq C$, then the Markov condition imposes independence of A and B, but this is possible only if $P(A) = 0$ or $P(B) = 0$, as $A \sqcap B$ must be empty [10]. Thus CR$\mathcal{ALC}$ does not exactly match the behavior of standard $\mathcal{ALC}$ when probabilities are unspecified, as logical constraints may interact with the Markov condition; we leave for the future a detailed study of the relationship between CR$\mathcal{ALC}$and standard $\mathcal{ALC}$.

Homogeneity. Note that a terminology in CR$\mathcal{ALC}$ may not specify a single measure over interpretations (in Example 1, the assessment $P(C|B \sqcup \exists r.D) = \alpha_3$ does not

guarantee that $P(C|B \sqcap \exists r.D)$ is constrained down to a single value). We assume the following *homogeneity condition* always holds. We assume that, given a concept C with parents $D_1, \ldots, D_m$, then for any conjunction of the m concepts $\pm D_i$, where $\pm D_i$ is either D_i or $\neg D_i$, $P(C(x)| \pm D_1(x) \sqcap \pm D_2(x) \sqcap \cdots \sqcap \pm D_m(x))$ is a constant across individuals x. Without this condition, a probability left unspecified in a terminology may take a distict value for each individual in the domain, a situation we wish to preclude. With the homogeneity condition, any terminology can be viewed as a non-recursive relational Bayesian network [25], except for the fact that some probabilities may not be precisely specified. Indeed, for a fixed finite domain $\mathcal{D}$, the propositionalization of a terminology $\mathcal{T}$ produces a *credal network*; that is, a Bayesian network where probabilities are not precisely specified [8]. Figure 1 shows a propositionalized version of $\mathcal{T}_1$ (Example 1).

Thus we have a logic with the constructs of $\mathcal{ALC}$, including assertions (with rigidity for individuals), plus probabilistic inclusions with semantics (1), acyclicity and the extended Markov condition, and homogeneity. We refer to the resulting logic as CR$\mathcal{ALC}$ (for "credal" $\mathcal{ALC}$).

Uniqueness. There is a way to guarantee uniqueness of probabilities that may be useful in practice. First, adopt the *unique names assumption*; that is, distinct names for individuals correspond to distinct elements of the domain. Second, assume the following *uniqueness condition*: (i) for each concept C: if C has no parents, then $P(C) = \alpha$ is given; and if C has parents, then either C is specified by a definition, or C has a single parent D and probabilistic inclusions with respect to D and $\neg D$ (that is, either C is determined by its parents or C is a "fully probabilistic" node); and (ii) for each role r, an assessment $P(r) = \alpha$ is made, whose semantics is $\forall x, y : P(r(x, y)) = \alpha$.

The unique names assumption and the uniqueness condition guarantee that, when we ground a terminology, we obtain a Bayesian network (different assumptions guarantee uniqueness in P-CLASSIC and PR-OWL).

Example 2. Consider a terminology $\mathcal{T}_2$ with concepts A, B, C, D, where: $P(A) = \alpha_1$, $B \sqsubseteq A$, $P(B|A) = \alpha_2$, $D \equiv \forall r.A$, $C \equiv B \sqcup \exists r.D$, and $P(r) = \alpha_3$. Figure 1 also applies, but now all probabilities are precisely specified.

4 Inference: First-Order Elimination and Loopy Propagation

Consider an inference in a CR$\mathcal{ALC}$ terminology, defined as the calculation of query $Q = P(A_0(a_0)|\mathcal{A})$ for $\mathcal{A} = \{A_j(a_j)\}_{j=1}^{M}$ (where $|\mathcal{D}| > M$). We start with both the uniqueness condition and *domain closure* (domain with finite cardinality n that is known and given as part of the input). Later we discuss removal of these assumptions.

We first derive the joint probability distribution over the set $V_{\mathcal{T}}^{n}$ containing all assertions generated from $\mathcal{T}$ and a domain $\mathcal{D}$ with cardinality n. To do so, introduce random variables that are indicator functions of grounded relations. We use the same notation for an assertion and its associated random variable. For instance, $C(a)$ and $r(a, b)$ refer both to assertions and to random variables that yield 1 if the assertion holds and 0 otherwise. Denote by $\mathcal{C}$ the set of concept names in $\mathcal{T}$, plus restrictions such as $\exists r.C$

and $\forall r.C$ (where C is always a concept name); by $\mathcal{R}$ the set of role names in $\mathcal{T}$; and by $\mathrm{pa}(C_i(x))$ the parents of concept C_i. Then our assumptions imply:

$$P(V_{\mathcal{T}}^n) = \prod_{C_i \in \mathcal{C};\ x \in \mathcal{D}} P(C_i(x)|\mathrm{pa}(C_i(x))) \times \prod_{r \in \mathcal{R};\ x,y \in \mathcal{D}} P(r(x,y))\,.$$

One can propositionalize any terminology and do inference in the resulting network; such a strategy is clearly not scalable. Our strategy is instead to employ techniques from *first-order variable elimination* [11,12,42]. The first step then is to write Q and $P(V_{\mathcal{T}}^n)$ in a *shattered* form [11]; that is, so that every pair of atoms can be grounded either into identical or disjoint sets of grounded atoms, taking into account constraints on the possible assertions.

A key insight is that in CR$\mathcal{ALC}$ we can syntactically shatter the query and the distribution at once. Define $\mathcal{D}' \doteq \{a_0, \ldots, a_M\}$ and $\mathcal{D}'' \doteq \mathcal{D} \backslash \mathcal{D}'$. The following theorem can be proved by inspecting Expression (2) and noting that it satisfies all conditions required for shattering [12]:

Theorem 1. *Query Q and $P(V_{\mathcal{T}}^n)$ are shattered when $P(V_{\mathcal{T}}^n)$ is written as:*

$$\begin{aligned} &\prod_{C_i \in \mathcal{C}; a' \in \mathcal{D}'} P(C_i(a')|\mathrm{pa}(C_i(a'))) \quad \times \prod_{r \in \mathcal{R}; a', a'' \in \mathcal{D}'} P(r(a', a'')) \\ &\qquad \times \prod_{r \in \mathcal{R}; a' \in \mathcal{D}'; x_0 \in \mathcal{D}''} P(r(a', x_0))\, P(r(x_0, a')) \qquad (2) \\ &\times \prod_{C_i \in \mathcal{C}; x_0 \in \mathcal{D}''} P(C_i(x_0)|\mathrm{pa}(C_i(x_0))) \quad \times \prod_{r \in \mathcal{R}; x_0, x_1 \in \mathcal{D}''} P(r(x_0, x_1))\,, \end{aligned}$$

where $\mathrm{pa}(C_i(x))$ denotes the parents of C_i if C_i is a concept name; and if C_i is a restriction $\exists r.C$ or $\forall r.C$, then $\mathrm{pa}(C_i(a'))$ denotes $\{r(a', a''), C(a'') : a'' \in \mathcal{D}'\} \cup \{r(a', x_0), C(x_0) : x_0 \in \mathcal{D}''\}$, and $\mathrm{pa}(C_i(x_0))$ denotes $\{r(x_0, a'), C(a') : a' \in \mathcal{D}'\} \cup \{r(x_0, x_1), C(x_1) : x_1 \in \mathcal{D}''\}$. □

The first line of Expression (2) encodes a propositional Bayesian network over individuals in Q. These individuals are connected to other individuals through roles $P(r(a', x_0))$ and $P(r(x_0, a'))$ (in $\mathrm{pa}(C_i(a'))$, $\mathrm{pa}(C_i(x_0))$ and in the second line of Expression (2)). The third line builds a relational Bayesian network with "generic" individuals x_0 and x_1, with connections $P(r(x_0, x_0))$ and $P(r(x_0, x_1))$. Figure 2 offers a visual translation of shattering on terminology $\mathcal{T}_2$ and query $P(C(a_0))$. A benefit from explicit shattering of the Q and $P(V_{\mathcal{T}}^n)$ is that we can apply "first-order" d-separation on the shattered network, thus eliminating unnecessary parts of the terminology. For instance, node $(\exists r.D)(x_0)$ can be removed from Figure 2 when we compute $P(C(a_0))$.

First-order variable elimination is, in essence, variable elimination in the shattered network. Some gains are apparent. For instance, node $B(x_0)$ in Figure 2 can be eliminated for all x_0 at once, as $\sum_{B(x_0)} P(V_{\mathcal{T}_2}^n) \propto \sum_{B(x_0)} \prod_{x_0 \in \mathcal{D}''} P(B(x_0)|A(x_0)) \times P(C(x_0)|B(x_0), (\exists r.A)(x_0))$, and we can invert summation and product in the last expression (this is an *inversion elimination* [11]). The elimination of nodes containing restrictions requires new techniques. Note that, while Braz et al do not have quantification in their language [11,12], Poole discusses the network where $\exists x : A(x)$ has single

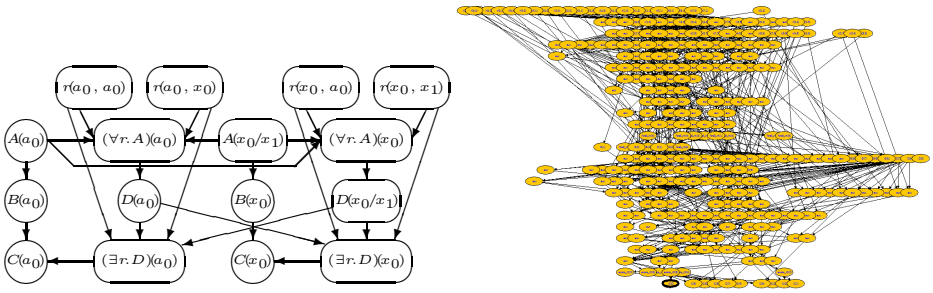

Fig. 2. A shattered version of $\mathcal{T}_2$ and query $P(C(a_0))$; grounded network for $\mathcal{T}_2$ and $|\mathcal{D}| = 10$

parent $A(x)$. Poole shows that $P(\exists x : A(x)) = 1 - (1 - P(A(x)))^n$ when the groundings of $A(x)$ are independent [42]. Such a nice result does not apply when restrictions are themselves parameterized, as is the case here (but approximations proposed later do use this insight).

We first note the algebraic structure of distributions $P(C_i|\text{pa}(C_i))$ when C_i is a restriction, using indicator functions. Remember that such distributions only yield 0 or 1 probabilities, as a restriction is completely determined by its parents. In the next two expressions, $\forall$ stands for the indicator function of $\forall r.C(x)$, $\exists$ stands for the indicator function of $\exists r.C(x)$, and likewise C stands for $C(x)$ and r stands for $r(x, y)$. Then:

$$P(\forall|\text{pa}(\forall)) = (\forall)\left(\prod_{y\in\mathcal{D}} 1 - r + rC\right) + (1 - \forall)\left(1 - \prod_{y\in\mathcal{D}} 1 - r + rC\right); \quad (3)$$

$$P(\exists|\text{pa}(\exists)) = (\exists)\left(1 - \prod_{y\in\mathcal{D}} 1 - rC\right)(1 - \exists)\left(\prod_{y\in\mathcal{D}} 1 - rC\right). \quad (4)$$

In theory, one could apply these expressions to the shattered network produced by Theorem 1, and then run first-order variable elimination. Our experience in trying the algorithm on small examples (such as the examples discussed previously) suggests that exact inference in the shattered network is rarely viable. The difficulty is that, *in exactly those cases where propositionalization does not work due to the presence of quantifiers, first-order inference seems to fail as well, at least in practice*. Note that first-order variable elimination does not guarantee elimination of "first-order" nodes beforehand; in the worst case the network (or vast parts of it) must be propositionalized. The difficulty is that restrictions typically lead to many connections between grounded nodes due to their quantifiers. To illustrate this fact, Figure 1 (right) shows a propositionalization of terminology $\mathcal{T}_2$ with a $n = 10$, produced using the Primula system for relational Bayesian networks (www.cs.aau.dk/~jaeger/Primula/). The network is dense due to auxiliary nodes that must be inserted to encode deterministic relations (mostly quantifiers); without these auxiliary nodes the probability tables cannot even be stored. Exact inference in this network does not seem possible even with the best available algorithms (calculation of $P(C(a_0))$ was possible up to $n = 9$ and failed for $n > 9$ due to memory exhaustion).

Instead of insisting on exact methods, a wiser and more scalable strategy is to look for approximations. Here the challenge is to find approximations that can exploit the structure of quantifiers. We investigate variational methods that "break" connections in networks; among those, the *loopy propagation* algorithm seems particularly suitable [39]. Another idea would be to use variational approximations of Noisy-OR [29] in Expressions (3) and (4); we leave this possibility for the future.

A natural strategy then is to propositionalize a terminology and run loopy propagation in the resulting network. Loopy propagation deals with quantifiers in a straightforward manner: a restriction node receives messages from its neighbors, and then locally produces messages by replacing the "local" distribution $P(C|\mathrm{pa}(C))$ by an appropriate Expression (3) or (4). Note that Poole's analysis of quantifiers, mentioned previously, applies locally during loopy propagation.

However, we can do *much better*, and this is one of the main insights of this paper. We can use the shattered network produced by Theorem 1 directly, and concoct a *first-order* version of loopy propagation: simply run loopy propagation in the shattered network, and combine the local messages using Expressions (3) or (4) as appropriate; if a message flows from a parameterized node to a restriction node, then it must be raised to a power (equal to the number of individuals that can be substituted for the parameter) before combination in Expressions (4) or (3). Note that it is not just the case that we are running loopy propagation in a conveniently modified version of $\mathcal{G}(\mathcal{T})$; we are indeed running a first-order version of loopy propagation, because all the messages that are sent in the parameterized portion of the shattered network would be replicated were the terminology propositionalized. Thus, the same excellent empirical performance that has been observed for loopy propagation (in propositional networks) is necessarily transferred to this first-order loopy propagation scheme.

Example 3. Consider node $(\forall r.A)(x_0)$ in Figure 2. A possible schedule of messages has nodes $A(a_0)$, $A(x_0/x_1)$, $r(x_0, a_0)$ and $r(x_0, x_1)$ sending messages to $(\forall r.A)(x_0)$. A message to be sent to node $D(x_0/x_1)$ is easily produced as, locally to the approximation scheme, $P((\forall r.A)(x_0) = 1) = \prod_{y \in \mathcal{D}} 1 - P(r(x_0, y))\,(1 - P(A(y)))$.

The idea that a first-order loopy propagation scheme can be built in probabilistic relational models was advanced by Jaimovich et al [28] for Markov networks without observed variables, and more recently by Sigla and Domingos [48] for Markov logic. What the shattered network allows us to do is to apply first-order loopy propagation on a structure that is fixed beforehand even when observations are made.

To illustrate the performance of this first-order loopy propagation scheme, consider again Example 2. The next table shows $P(C(a_0))$ for a domain containing individuals $a_0, \ldots, a_{n-1}$, for several n. Whenever possible we show the result of exact inference with a state-of-art algorithm (in the SamIam package at reasoning.cs.ucla.edu/samiam, using the recursive decomposition algorithm). Note that inferences converge to a stable value for n large; the analysis of Section 5 sheds light on this issue.

n	1	2	3	5	9	10	20	50
Loopy: $P(C(a_0))$	0.5175	0.5383	0.5291	0.4885	0.4296	0.4223	0.4049	0.4050
Exact: $P(C(a_0))$	0.4350	0.4061	0.4050	0.4050	0.4050	—	—	—

As another example, we have built a larger terminology containing 15 nodes, with 3 restrictions and a considerable amount of arcs amongst nodes (details are omitted due to lack of space). Exact inference became unfeasible even for small domains; loopy propagation produced inferences in seconds for $n = 10, 20$.

We have worked so far under the uniqueness assumption. In practice it may be useful to drop such a demanding assumption, as one may for instance choose not to specify $P(r)$ for all roles. In theory, lack of uniqueness (in the presence of homogeneity) is easy to handle: instead of a single probability measure, we must now deal with a set of probability measures; instead of computing a single value for an inference, we must compute a minimum and a maximum value for the probabilities of interest [8]. In practice, the calculation of exact probability bounds without uniqueness appears quite challenging computationally. However, the picture is different for approximate algorithms. Indeed, the *L2U algorithm* [23] is a version of loopy propagation for credal networks with binary variables that has been observed empirically to have excellent performance. One can extend L2U to first-order just as we did for loopy propagation, now letting messages carry probability intervals. Again, messages are locally combined and quantifiers can be dealt with in a local fashion. And again, the first-order algorithm has the property that its performance is identical to what would be obtained were the terminology grounded and L2U performed in the grounded network.

To illustrate the performance of this first-order L2U scheme, we return again to Example 2. But now we take $P(r)$ to be *entirely unspecified*; that is, it can be any value in the interval $[0, 1]$. This is in accord with the usual description logics where no information is provided for roles except through their use in restrictions. In the next table we show inferences for $P(C(a_0))$ for a domain for several n (an inference now produces an interval containing lower/upper probabilities). Note the convergence of probabilities as n increases. Perhaps more surprinsingly, note the little influence $P(r)$ has in the value of $P(C(a_0))$; this suggests that one may leave various probabilities free in a knowledge base and still get meaningful answers.

n	1	3	5	10	20	50
L2U: $P(C(a_0))$	[0.405000 0.464500]	[0.405000 0.406783]	[0.405000 0.405030]	[0.405000 0.405000]	[0.405000 0.405000]	[0.405000 0.405000]

5 Infinite Domains

Infinite domains are useful, when a domain is finite but very large, and necessary, when the available information does not constrain the cardinality of the domain. However, infinite domains are challenging: there are issues concerning existence and uniqueness of a joint measure, and then there are obvious difficulties with inference based on propositionalization. Indeed, first-order variable elimination may fail for infinite domains when probabilities are expressed in Braz et al's language [12].

We use results by Jaeger [26,27] to prove existence and uniqueness for CR$\mathcal{ALC}$:

Theorem 2. *Every terminology $\mathcal{T}$ in* CR$\mathcal{ALC}$ *defines a unique joint distribution under the uniqueness condition, and $P(V_{\mathcal{T}}^{\infty}) = \lim_{n\to\infty} P(V_{\mathcal{T}}^{n})$.*

Proof. As every terminology in CR$\mathcal{ALC}$ under the given assumptions defines a non-recursive relational Bayesian network, it defines a unique joint distribution over an appropriate algebra [27, Th. 4.7]. Now note that all restrictions can be rewritten as *max-based combination functions* [25], and consequently they are *exponentially convergent* [26, Def. 3.4]. The limit of the joint distribution as n grows is then the joint distribution for an infinite domain [26, Th. 3.9]. □

Similar results have been proved for the BLOG language [35]. Existence has also been proved for other probabilistic description logics where infinite domains may lead to violation of uniqueness [7,41,47]; existence also obtains for logics that assign probabilities over domains (even without uniqueness) [18,33].

We now turn to inference with infinite domains, restricting ourselves to queries conditioned on $\top$. Here again Theorem 1 comes handy. We simply analyze the shattered graph top down, taking limits at each Expression (3) or (4) that we meet. (Gaifman's theorem [15] is necessary here to prove correctness, as it shows that probabilities for restrictions are obtained by taking limits over disjunctions/conjunctions.)

For example, if $P(r) > 0$ and $P(A) > 0$, then for any x, we have both limits $\lim_{n\to\infty} P((\exists r.A)(x)|\mathrm{pa}((\exists r.A)(x))) = 1, \lim_{n\to\infty} P((\forall r.A)(x)|\mathrm{pa}((\forall r.A)(x))) = 0$ whenever the conditioning events have nonzero probability. There are several other possibilities, by taking combinations of $P(r) = 0$, $P(r) = 1$, $P(A) = 0$, $P(A) = 1$; all of these cases lead to probability 0 or 1 for restrictions. This rationale constructs a *zero-one law* for CR$\mathcal{ALC}$. A similar zero-one law for description logics has been proved by Ycart and Rousset [51] for a uniform distribution over possible assertions; Jaeger's analysis of infinite domains [27] also implies several related zero-one laws. An interesting side effect is that we can make a linear number of queries in $\mathcal{G}(\mathcal{T})$ (one for each restriction, from top to bottom) to build a relational Bayesian network *without roles* on which inferences can be made. The next example should be sufficient to illustrate how these ideas can be used when computing queries for infinite domains.

Example 4. Consider terminology $\mathcal{T}_2$, with $\alpha_1 = 2\alpha_2 = 3\alpha_3 = 0.9$, query $P(C(a_0))$ and an infinite domain. From top down: as $P(r) > 0$ and $P(A) > 0$, $P(\forall r.A) = 0$; thus $P(D) = 0$ and $P(C) = P(B) = P(B|A)\,P(A) + P(B|\neg A)\,P(\neg A) = \alpha_2\alpha_1$ and $P(C(a_0)) = 0.405$. Compare this value with the values obtained in the previous section for n large.

6 Conclusion

This paper started from the desire to represent terminologies with probabilities over concepts, in such a way that queries involving assertions can be handled. A probabilistic version of the $\mathcal{ALC}$ description logic has been introduced, with an interpretation-based semantics that allows probabilistic inclusions and queries on the probability of assertions (thus bypassing the problem of direct inference). The paper contributed with techniques for first-order inference algorithms in finite domains (and to a limited extent, in infinite domains) and in particular with a first-order loopy propagation scheme that is based on the "shattered" version of a terminology. Such techniques may be useful to other languages such as Costa and Laskey's PR-OWL [6]. The use of shattering and loopy propagation may be useful in other logics as well, a point that we leave for future investigation.

The paper clearly left a few topics for the near future: investigation of inference in infinite domains *with observations*, and more empirical testing with improved variational approximations. In the future we plan to investigate extensions of the language, such as at-least and at-most restrictions (these can be handled through loopy propagation) or role inclusions, leading gradually to a full-blown version of the OWL language.

A few closing words on the assumptions we have adopted seem appropriate. Some of them could easily be relaxed; for instance, we could allow probabilistic inclusions with a set of conditioned concepts, provided no such concept is a restriction. Other assumptions seem difficult to remove, such as the rigidity and unique names assumptions on individuals, and the homogeneity condition on probabilities. Also, it seems difficult to remove the acyclicity condition (and the associated Markov condition). Undirected graphs would seem more amenable to a mix of logical and probabilistic constructs as cycles are not a concern [43,50]; however, undirected graphs interact awkwardly with probabilistic logic, as the usual Markov condition for undirected graphs fails to imply factorization of measures in the presence of logical constraints [38]. A possibility is to postulate a factorization from the outset [43,47]; we have preferred to stay with the Markov condition of directed acyclic graphs, even though probabilistic description logics based on undirected graphs certainly deserve more study.

As a final comment, we note that it would also be desirable to drop the domain closure assumption, as description logics never assume anything about cardinality of the domain [37]. Interest must then be in computing minima/maxima of probabilities as n varies. Given the rigidity assumption, a query fixes observations with respect to elements of the domain, and leaves other elements unobserved. Even though no general technique for such optimization problems seems to be developed at this point, in some simple cases one can find bounds and optimal n:

Example 5. Consider terminology $\mathcal{T}_1$ with $\alpha_1 = 2\alpha_2 = 3\alpha_3 = 4\alpha_4 = 1000\epsilon = 0.9$, and query $P(C(a_0))$, with no information about domain cardinality. This terminology is simple enough that we can write down the expressions for the query and optimize over them. We obtain $P(C(a_0)) \in [0.1215, 0.3]$; the lower bound is obtained for $P(r) = 0$ (any n) and the upper bound for $P(r) = 1$ and $n = \infty$.

Acknowledgements

This work was partially funded by FAPESP (04/09568-0); the first author is partially supported by CNPq, and the second author is supported by HP Brazil R&D. We thank all these organizations.

References

1. Andersen, K.A., Hooker, J.N.: Bayesian logic. Decision Support Systems 11, 191–210 (1994)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: Description Logic Handbook. Cambridge University Press, Cambridge (2002)
3. Bacchus, F.: Representing and Reasoning with Probabilistic Knowledge: A Logical Approach. MIT Press, Cambridge (1990)

4. Berners-Lee, T., Hendlers, J., Lassila, O.: The semantic web. In: Scientific American, pp. 34–43 (2001)
5. Borgida, A.: On the relative expressiveness of description logics and predicate logics. Artificial Intelligence 82(1-2), 353–367 (1996)
6. Costa, P.C.G., Laskey, K.B.: PR-OWL: A framework for probabilistic ontologies. In: Conf. on Formal Ontology in Information Systems (2006)
7. da Costa, P.C.G., Laskey, K.B.: Of Klingons and starships: Bayesian logic for the 23rd century. In: Conf. on Uncertainty in Artificial Intelligence (2005)
8. Cozman, F.G.: Credal networks. Artificial Intelligence 120, 199–233 (2000)
9. Cozman, F.G., de Campos, C.P., Ferreira da Rocha, J.C.: Probabilistic logic with independence. Int. Journal of Approximate Reasoning (in press, September 7, 2007) doi: 10.1016/j.ijar.2007.08.002
10. Polpo de Campos, C., Cozman, F.G., Luna, J.E.O.: Assessing a consistent set of sentences in relational probabilistic logic with stochastic independence. Journal of Applied Logic (to appear)
11. de Salvo Braz, R., Amir, E., Roth, D.: Lifted first-order probabilistic inference. In: Int. Joint Conf. in Artificial Intelligence (IJCAI) (2005)
12. de Salvo Braz, R., Amir, E., Roth, D.: MPE and partial inversion in lifted probabilistic variable elimination. AAAI (2006)
13. Ding, Z., Peng, Y., Pan, R.: BayesOWL: Uncertainty modeling in semantic web ontologies. In: Soft Computing in Ontologies and Semantic Web. Studies in Fuzziness and Soft Computing, vol. 204. Springer, Berlin (2006)
14. Dürig, M., Studer, T.: Probabilistic ABox reasoning: preliminary results. In: Description Logics, pp. 104–111 (2005)
15. Gaifman, H.: Concerning measures on first-order calculi. Israel Journal of Mathematics 2, 1–18 (1964)
16. Getoor, L., Friedman, N., Koller, D., Taskar, B.: Learning probabilistic models of relational structure. In: Int. Conf. on Machine Learning, pp. 170–177 (2001)
17. Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning. MIT Press, Cambridge (2007)
18. Lukasiewicz, T., Giugno, R.: P-SHOQ(D): A probabilistic extension of SHOQ(D) for probabilistic ontologies in the semantic web. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 86–97. Springer, Heidelberg (2002)
19. Halpern, J.Y.: Reasoning about Uncertainty. MIT Press, Cambridge (2003)
20. Heinsohn, J.: Probabilistic description logics. In: Conf. on Uncertainty in Artificial Intelligence, pp. 311–318 (1994)
21. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: The making of a web ontology language. Journal of Web Semantics 1(1), 7–26 (2003)
22. Hung, E., Getoor, L., Subrahmanian, V.S.: Probabilistic interval XML. ACM Transactions on Computational Logic 8(4) (2007)
23. Ide, J.S., Cozman, F.G.: Approximate algorithms for credal networks with binary variables. Int. Journal of Approximate Reasoning 48(1), 275–296 (2008)
24. Jaeger, M.: Probabilistic reasoning in terminological logics. Principles of Knowledge Representation (KR), pp. 461–472 (1994)
25. Jaeger, M.: Relational Bayesian networks. In: Uncertainty in Artificial Intelligence, pp. 266–273. Morgan Kaufmann, San Francisco (1997)
26. Jaeger, M.: Convergence results for relational Bayesian networks. LICS (1998)
27. Jaeger, M.: Reasoning about infinite random structures with relational Bayesian networks. In: Knowledge Representation. Morgan Kaufmann, San Francisco (1998)
28. Jaimovich, A., Meshi, O., Friedman, N.: Template based inference in symmetric relational Markov random fields. Uncertainty in Artificial Intelligence, Canada. AUAI Press (2007)

29. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S.: An introduction to variational methods for graphical models. Machine Learning 37, 183–233 (1999)
30. Koller, D., Pfeffer, A.: Object-oriented Bayesian networks. In: Conf. on Uncertainty in Artificial Intelligence, pp. 302–313 (1997)
31. Kyburg Jr., H.E., Teng, C.M.: Uncertain Inference. Cambridge University Press, Cambridge (2001)
32. Lukasiewicz, T.: Probabilistic logic programming. In: European Conf. on Artificial Intelligence, pp. 388–392 (1998)
33. Lukasiewicz, T.: Expressive probabilistic description logics. Artificial Intelligence (to appear)
34. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web (submitted, 2008)
35. Milch, B., Marthi, B., Sontag, D., Russell, S., Ong, D.L., Kolobov, A.: BLOG: Probabilistic models with unknown objects. In: IJCAI (2005)
36. Milch, B., Russell, S.: First-order probabilistic languages: into the unknown. In: Int. Conf. on Inductive Logic Programming (2007)
37. Motik, B., Horrocks, I., Rosati, R., Sattler, U.: Can OWL and logic programming live together happily ever after? In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 501–514. Springer, Heidelberg (2006)
38. Moussouris, J.: Gibbs and Markov random systems with constraints. Journal of Statistical Physics 10(1), 11–33 (1974)
39. Murphy, K.P., Weiss, Y., Jordan, M.I.: Loopy belief propagation for approximate inference: An empirical study. In: Uncertainty in Artificial Intelligence, pp. 467–475 (1999)
40. Nottelmann, H., Fuhr, N.: Adding probabilities and rules to OWL lite subsets based on probabilistic datalog. Int. Journal of Uncertainty, Fuzziness and Knowledge-based Systems 14(1), 17–42 (2006)
41. Pfeffer, A., Koller, D.: Semantics and inference for recursive probability models. In: AAAI, pp. 538–544 (2000)
42. Poole, D.: First-order probabilistic inference. In: Int. Joint Conf. on Artificial Intelligence (IJCAI), pp. 985–991 (2003)
43. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 62(1-2), 107–136 (2006)
44. Schild, K.: A correspondence theory for terminological logics: Preliminary report. In: Int. Joint Conf. on Artificial Intelligence, pp. 466–471 (1991)
45. Schmidt-Schauss, M., Smolka, G.: Attributive concept descriptions with complements. Artificial Intelligence 48, 1–26 (1991)
46. Sebastiani, F.: A probabilistic terminological logic for modelling information retrieval. In: Int. ACM Conf. on Research and Development in Information Retrieval (SIGIR), Dublin, Ireland, pp. 122–130. Springer, Heidelberg (1994)
47. Sigla, P., Domingos, P.: Markov logic in infinite domains. In: Uncertainty in Artificial Intelligence, pp. 368–375. AUAI Press (2007)
48. Sigla, P., Domingos, P.: Lifted first-order belief propagation. AAAI (2008)
49. Staker, R.: Reasoning in expressive description logics using belief networks. In: Int. Conf. on Information and Knowledge Engineering, Las Vegas, USA, pp. 489–495 (2002)
50. Taskar, B., Abbeel, P., Koller, D.: Discriminative probabilistic models for relational data. In: Conf. on Uncertainty in Artificial Intelligence, Edmonton, Canada (2002)
51. Ycart, B., Rousset, M.-C.: A zero-one law for random sentences in description logics. Colloquium on Mathematics and Computer Science (2000)
52. Yelland, P.M.: Market analysis using a combination of Bayesian networks and description logics. Technical Report SMLI TR-99-78, Sun Microsystems Laboratories (1999)