Reasoning With Weighted Ontologies

Rafael Peñaloza

Theoretical Computer Science TU Dresden, Germany
penaloza@tcs.inf.tu-dresden.de

Abstract. We study the problem of reasoning over weighted ontologies. We assume that every axiom is labeled with an element of a distributive lattice (called its weight) and try to compute its so-called boundary, with respect to a given property. We show that axiom pinpointing is the most general instance of this problem. Finally, we present three applications of the problem of boundary computation.

1 Introduction

In recent years, the study of pinpointing and other related problems has gained notoriety, due to the need to correct ontologies of very large size. The main motivation for pinpointing arises from the fact that ontology development is an error prone activity. Once an ontology grows beyond the few hundred axioms, it is not hard to encounter unexpected consequences that follow from it. Although not all unexpected consequences are necessarily wrong—indeed, they may be correct instances which were just previously unknown to the knowledge engineer—there is always a need to understand why they follow and, given the case, correct the ontology to get rid of them.

Axiom pinpointing deals with the problem of finding minimal subsets of an ontology that entail a given consequence. These sets usually receive the name of *MinAs* or *justifications*. There exist two basic approaches toward pinpointing. One tries to find a monotone Boolean formula ϕ, called the *pinpointing formula*, that encodes the set of all MinAs through a bijection with the minimal valuations that satisfy ϕ [1, 2]. The other idea is to compute the MinAs directly [16, 19, 10]. The latter approach is in fact a special case of the former, in which the pinpointing formula is restricted to appear in disjunctive normal form.

Even when working on particular problems, the study of general solutions can be helpful for shining light over specific problems or characteristics that may not be evident at first sight. Thus, the study of pinpointing of general tableaux [2] showed that termination is not as trivial as intuition may suggest. On the other hand, the automata-based pinpointing algorithm described in [1] has taught us something more about the pinpointing formula. The construction requires a distributive lattice [8]; the set of all monotone Boolean formulas over a set P of propositional variables forms the *free* distributive lattice of $|P|$ generators, and the pinpointing formula is an element of this lattice. Evenmore, the algorithm that finally allows us to compute the pinpointing formula from a so-called axiomatic automaton is actually shown to work for any distributive lattice. Thus, one can wonder whether one can study the pinpointing formula in a general way, by reasoning about general distributive lattices. This paper tries to give a positive answer to that question.

We introduce the notion of a *boundary* for a property. The boundary is a generalization of the pinpointing formula, where the axioms may be labeled arbitrarily, with elements of any distributive lattice. In other words, we have a weighted ontology. Intuitively, a boundary is an element of the lattice that separates the subontologies from which the property follows of those from which the property does not follow. We then investigate whether it is possible to compute such a boundary. In a way, we show that we already have a deep understanding of the boundary, through our knowledge of the pinpointing formula. In fact, we show that the pinpointing formula can be seen as an abstract representation of all the boundaries for a given property, which can be instantiated via a homomorphism between the free distributive lattice and the particular lattice under consideration at the moment.

Reasoning over weighted ontologies has applications beyond a better understanding of axiom-pinpointing. In general, we can think of axioms holding a weight that can represent how trusted it is, or how much it costs to access or use it, etc.

The paper is divided as follows. In Section 2, we motivate and introduce our notion of boundaries for properties. Then, in Section 3 we show that the boundary can always be computed as the image of the pinpointing formula through an adequate lattice homomorphism. Finally, before our conclusions, we present in Section 4 three scenarios where the computation of a boundary solves a reasoning problem with respect to DLs.

2 Property Boundaries

We start this section by introducing the properties we want to reason about. In general, a property can be defined as a set of inputs, whose elements correspond exactly to those satisfying the property. In this work we want to reason with respect to knowledge bases, which are basically sets of axioms. However, not every set of axioms can be called a knowledge base. Consider, for instance, acyclic TBoxes: these are sets of concept definitions that are irredundant and acyclic. Notice that every subset of an acyclic TBox is itself an acyclic TBox. Following these ideas, we consider knowledge bases as sets of axioms satisfying an *admissibility* restriction (we call these sets *admissible*) such that if $\mathcal{T}$ is admissible, then every subset of $\mathcal{T}$ is also admissible.

We define a *property* as a set of tuples of the form $(\mathcal{I}, \mathcal{T})$, where $\mathcal{I}$ is an *input* and $\mathcal{T}$ is an admissible set of *axioms*. Such tuples are called *axiomatized inputs*. We are interested only in properties that are monotone with respect to the knowledge base in the sense that whatever follows from $\mathcal{T}$ must also follow from any (admissible) superset of $\mathcal{T}$. A property $\mathcal{P}$ is called a *consequence property* (abbreviated as c-property) if $(\mathcal{I}, \mathcal{T}) \in \mathcal{P}$ implies that for every admissible set of axioms $\mathcal{S}$, if $\mathcal{T} \subseteq \mathcal{S}$, then $(\mathcal{I}, \mathcal{S}) \in \mathcal{P}$. Examples of c-properties are *un*satisfiability of concepts or subsumption w.r.t. TBoxes. The reason for imposing this monotonicity in the properties is to keep in track with previous notions of pinpointing. Recall that axiom pinpointing tries to find all *minimal* subontologies from which the a given consequence follows; if the property relating ontologies with consequences is not monotone, finding such minimal sets does not make much sense.

Rather than just reasoning over a knowledge base $\mathcal{T}$, we are interested in computing what will be later called a *boundary* based on weights associated to each of the axioms. This boundary will allow us to understand how weight-dependent portions of the ontology influence the property. This is now explained more formally.

Let $(S, \leq)$ be a distributive lattice and $\Gamma = (\mathcal{I}, \mathcal{T})$ an axiomatized input. A mapping $\mathsf{lab} : \mathcal{T} \to S$ is called a *labeling function*. Intuitively, we want to compute the highest point s_Γ in the lattice where we can cut our ontology without losing the property; i. e. that the property still follows from the ontology having only axioms whose label is greater than or equal to s_Γ. This generalizes the notion of a pinpointing formula [2].

Definition 1 (Pinpointing formula). *Let $(\mathcal{I}, \mathcal{T}) \in \mathcal{P}$, lab a function mapping each $t \in \mathcal{T}$ with a unique propositional variable, and P the set of all propositional variables labeling an element of $\mathcal{T}$. A monotone Boolean formula ϕ over P is called a* pinpointing formula *for $(\mathcal{I}, \mathcal{T})$ if for every $\mathcal{S} \subseteq \mathcal{T}$ it holds that $(\mathcal{I}, \mathcal{S}) \in \mathcal{P}$ if and only if the valuation $\{\mathsf{lab}(t) \mid t \in \mathcal{S}\}$ satisfies ϕ.*

Notice that the set of distinct monotone Boolean formulas over P defines a distributive lattice where the order is defined by $\phi \leq \psi$ iff $\psi \Rightarrow \phi$. In this setting, conjunction corresponds to the greatest lower bound operator, and disjunction to the least upper bound. Additionally, each valuation $\mathcal{V}$ uniquely identifies an element of the lattice $\phi_\mathcal{V} = \bigwedge_{p \in \mathcal{V}} p$ and a subontology $\mathcal{T}_\mathcal{V} = \{t \in \mathcal{T} \mid \mathsf{lab}(t) \in \mathcal{V}\}$. This latter definition is equivalent to $\mathcal{T}_\mathcal{V} = \{t \in \mathcal{T} \mid \phi_\mathcal{V} \leq \mathsf{lab}(t)\}$.

In the general case, we want to follow the same ideas as for pinpointing; thus, we define for every element s of the lattice S the subset $\mathcal{T}_s = \{t \in \mathcal{T} \mid s \leq \mathsf{lab}(t)\}$. We are then looking for an element $s_\Gamma \in S$ that behaves as a pinpointing formula with respect to the lattice S and the labeling function lab; i. e. such that for every $s \in S$ it holds that $s \leq s_\Gamma$ if and only if $(\mathcal{I}, \mathcal{T}_s) \in \mathcal{P}$. Unfortunately, such a value does not necessarily exist, as shown by the following example.

Example 1. Consider the distributive lattice $(S_4, \leq_4)$ having the four elements $S_4 = \{0, a_1, a_2, 1\}$, where 0 and 1 are the least and greatest elements, respectively, and a_1, a_2 are incomparable w.r.t. $\leq_4$. Let $\mathcal{T}$ be a set formed by the axioms ax_1 and ax_2, which are labeled by the elements a_1 and a_2 of S_4, respectively, and let $\mathcal{P}$ be the c-property defined as $\mathcal{P} = \{(\mathcal{I}, \mathcal{S}) \mid |\mathcal{S}| \geq 1\}$, where $|\mathcal{S}|$ denotes the cardinality of $\mathcal{S}$. It is easy to see that, given the axiomatized input $\Gamma = (\mathcal{I}, \mathcal{T})$, there is no element s_Γ that satisfies the condition described above. Indeed, if we choose 0 or a_1, then a_2 violates the condition, as $a_2 \not\leq s_\Gamma$, but $(\mathcal{I}, \mathcal{T}_{a_2}) \in \mathcal{P}$. And if 1 is chosen, then 1 itself violates it, as $1 \leq s_\Gamma$ but $(\mathcal{I}, \mathcal{T}_1) \notin \mathcal{P}$.

The reason why the property in this example fails to have an adequate s_Γ arises from asking the condition that $s \leq s_\Gamma$ iff $(\mathcal{I}, \mathcal{T}_s) \in \mathcal{P}$ to be satisfied *by every* element $s \in S$. Indeed, if we look again at the case of pinpointing, we only ask this condition to follow for every valuation; that is, for formulas having no disjunction. Recall that in this specific lattice, disjunction corresponds to the least upper bound (or *join*) operator. Then, valuations are *join prime* elements of the free distributive lattice; that is, they cannot be constructed as the join of other elements of the lattice. This notion will allow us to overcome the problem shown in Example 1.

Definition 2 (Join prime). *Let $(S, \leq)$ be a lattice. Given a set $R \subseteq S$, let $R_\otimes$ denote the closure of R under the meet operator.*[1] *An element $s \in S$ is called* join prime relative to R *if for every $R' \subseteq R_\otimes$, $s \leq \bigoplus_{r \in R'} r$ implies that there is an $r_0 \in R'$ such that $s \leq r_0$. The element s is called simply* join prime *if it is join prime relative to S.*

Obviously, any element that is join prime relative to a set R is also join prime relative to any subset of R. Moreover, if s is join prime relative to R and belongs to the sublattice generated by R, then $s \in R_\otimes$. Nonetheless, the converse is not true; that is, there may be elements of $R_\otimes$ that are not join prime relative to R. We will consider only the elements that are join prime relative to the image of the labeling function lab to define the boundary.

Definition 3 (Boundary). *Let $(S, \leq)$ be a lattice, $\mathcal{P}$ a c-property, $\Gamma = (\mathcal{I}, \mathcal{T})$ an axiomatized input and $\mathsf{lab} : \mathcal{T} \to S$ a labeling function. Let S_{lab} be the image of lab and S^*_{lab} the sublattice generated by S_{lab}. An element $s \in S^*_{\mathsf{lab}}$ is called a* boundary *for Γ in $\mathcal{P}$ if for every element $r \in S^*_{\mathsf{lab}}$ that is join prime relative to S_{lab} it holds that $r \leq s$ iff $(\mathcal{I}, \mathcal{T}_r) \in \mathcal{P}$.*

Returning to Example 1, we have that $S_{\mathsf{lab}} = \{a_1, a_2\}$ and $S^*_{\mathsf{lab}} = S_4$. The only element of S_4 that is not join prime relative to S_{lab} is 1, which is itself a boundary.

Notice that we have restricted the discourse in Definition 3 to the join prime elements relative to S_{lab} that belong to S^*_{lab}. This restriction is done without loss of generality. Indeed, as S is a distributive lattice, so is S^*_{lab}. Then, every element s of S^*_{lab} can be written as the join of all the join prime elements (relative to S_{lab}) in S^*_{lab} that are smaller or equal to s. This in particular shows that for every $r \in S$ that is join prime relative to S_{lab}, if $(\mathcal{I}, \mathcal{T}_r) \in \mathcal{P}$, then there is a $r' \in S^*_{\mathsf{lab}}$ join prime relative to S_{lab} such that $(\mathcal{I}, \mathcal{T}_{r'}) \in \mathcal{P}$. For simplicity, for the rest of this paper we will assume w.l.o.g. that $S = S^*_{\mathsf{lab}}$. As the ontology is finite and finitely generated distributive lattices are also finite, this assumption in particular entails that S is also finite.

3 Boundaries Through Pinpointing

Up to now we have presented the pinpointing formula of a property w.r.t. a given axiomatized input as a special case of boundary, from which we attempted to

[1] To avoid confusion with the Boolean operators, we denote the lattice operators by $\oplus$ (join) and $\otimes$ (meet).

develop the general notion. As it turns out, the pinpointing formula can be seen as an abstract representation of a boundary w.r.t. *distributive* lattices. Given a labeling function, the pinpointing formula can be then instantiated to the desired boundary. We proceed now to explain this in more detail.

Recall that the definition of pinpointing formula requires a labeling function that maps each axiom in $\mathcal{T}$ to a *unique* propositional variable. The closure of the image of this labeling function over the logical operators disjunction and conjunction yields the lattice of all monotone Boolean formulas over the set of labels. This lattice, which we will denote $\mathbb{B}^{\mathcal{T}}$, is in fact the free distributive lattice generated by $|\mathcal{T}|$ incomparable elements. As the labels of any two different axioms are incomparable w.r.t. $\mathbb{B}^{\mathcal{T}}$, this mapping in fact defines the most general distributive lattice one can produce through a labeling function. We will exploit these characteristics building an homomorphism between $\mathbb{B}^{\mathcal{T}}$ and the distributive lattice S that preserves the boundary; i. e. that maps the pinpointing formula to the boundary on S.

In the following, we consider a c-property $\mathcal{P}$ and an axiomatized input $\Gamma = (\mathcal{I}, \mathcal{T})$ such that $\Gamma \in \mathcal{P}$. We denote by $\mathsf{lab}_{\mathbb{B}}$ the labeling function that maps each axiom in $\mathcal{T}$ with a unique propositional variable in $\mathbb{B}^{\mathcal{T}}$, and by ϕ_Γ the pinpointing formula for Γ.

Consider now a distributive lattice S and let lab_S be a labeling function mapping each axiom in $\mathcal{T}$ to an element of S. We build a function $h : \mathbb{B}^{\mathcal{T}} \to S$ inductively, as follows: for every propositional variable $p \in \mathbb{B}^{\mathcal{T}}$, let $t \in \mathcal{T}$ be such that $\mathsf{lab}_{\mathbb{B}}(t) = p$; then $h(p) = \mathsf{lab}_S(t)$. Given two formulas $\phi_1, \phi_2 \in \mathbb{B}^{\mathcal{T}}$, we define $h(\phi_1 \wedge \phi_2) = h(\phi_1) \otimes h(\phi_2)$ and $h(\phi_1 \vee \phi_2) = h(\phi_1) \oplus h(\phi_2)$. Obviously, the function h is an homomorphism. We show now that it maps the pinpointing formula to a boundary.

Theorem 1. *Let ϕ be a pinpointing formula for Γ, then $h(\phi)$ is a boundary for Γ in $\mathcal{P}$.*

Proof. Assume w.l.o.g. that ϕ is in DNF; that is, $\phi = \phi_1 \vee \ldots \vee \phi_n$, where each ϕ_i is a conjunction of propositional variables. Then, for every $i, 1 \leq i \leq n$, $h(\phi_i) \in (S_{\mathsf{lab}})_\otimes$, and $h(\phi) = \bigoplus_{i=1}^{n} h(\phi_i)$. We need to show that, for every $r \in S$ join prime relative to S_{lab}, $r \leq h(\phi)$ iff $(\mathcal{I}, \mathcal{T}_r) \in \mathcal{P}$. For $r \in S$, define the valuation $\mathcal{V}_r := \{\mathsf{lab}_{\mathbb{B}}(t) \mid r \leq \mathsf{lab}_S(t)\}$. Notice that $\mathcal{T}_r = \mathcal{T}_{\mathcal{V}_r}$ and $h(\mathcal{V}_r) = r$. We have then that $(\mathcal{I}, \mathcal{T}_r) \in \mathcal{P}$ iff $(\mathcal{I}, \mathcal{T}_{\mathcal{V}_r}) \in \mathcal{P}$ which, as ϕ is a pinpointing formula, holds iff $\mathcal{V}_r$ satisfies ϕ iff $r = h(\mathcal{V}_r) \leq h(\phi)$. $\square$

This theorem tells us that computing the pinpointing formula for an axiomatized input suffices for obtaining the boundary w.r.t. any distributive lattice and any labeling function. There exist several methods for computing the pinpointing formula either by modifying a decision procedure [2, 1, 16, 13] (glass-box), by using it as a black-box [10], or by a mixture of both glass- and black-box approaches [4, 19]. Unfortunately, it has also been shown that pinpointing is a hard task [3]. Obviously, as computing the pinpointing formula is an instance of boundary computing, the hardness results that exist for pinpointing must also follow in this more general case. It is however possible that for particular lattices and labeling functions, the problem is a simpler one. In such a case, trying to compute first the pinpointing formula and then mapping it to the boundary will unavoidably produce a suboptimal procedure.

For instance let S be a (finite) total order, which is obviously a distributive lattice where all the elements are join prime. Given a labeling function lab, we can compute a boundary for Γ making polynomially many calls to a reasoner as follows. Let $s_n < \ldots < s_2 < s_1$ be all elements of S. Starting from $i = 1$, iteratively test whether $(\mathcal{I}, \mathcal{T}_{s_i}) \in \mathcal{P}$. If the answer is no, then increase i by one, and test again; otherwise, i. e. if the answer is yes, then s_i is the boundary. This black-box process needs at most n calls to the reasoner. Notice that n is bounded by the number of axioms in $\mathcal{T}$, as S is in fact the image of lab. Thus, if $\mathcal{P}$ can be decided in polynomial time in the size of $\mathcal{T}$, then this method computes the boundary in polynomial time in the same measure. In the next section we will show a setting where finding a boundary over a total order is useful.

In [1], we have shown that automata deciding a c-property can be transformed into weighted automata over the distributive lattice $\mathbb{B}^{\mathcal{T}}$ in such a way that the so-called behaviour of these weighted automata yields a pinpointing formula. We then showed how to efficiently compute such a behaviour in time polynomial on the number of states of the automaton. In fact, our algorithm is able to compute the behaviour of any weighted automaton over any distributive lattice. It is then not difficult to see that the same method can be used for computing the boundary directly, without detouring first through the pinpointing formula. Clearly, one can likewise modify the tableau-based glass-box method [2] to take values from an arbitrary distributive lattice, and even black-box approaches making use of Reiter's Hitting Set Tree algorithm [15] such as the ones presented in [10, 19] can benefit from a direct use of the lattice elements while computing a boundary.

On the other hand, computation of a boundary through the pinpointing formula has the clear benefit of needing to be done only once and be applicable even if the lattice or labeling function change, as long as the ontology itself remains unmodified.

4 Applications of the Boundary

Additionally from the computation of a pinpointing formula, there exist other scenarios where the computation of a boundary is a problem of interest. In this section we present three such cases: access control formalisms, fuzzy, and possibility reasoning.

4.1 Access Control

Consider the following setting. We have a distributive lattice defining a rights hierarchy. Each axiom in an ontology is associated to a *security level*: an element in the rights hierarchy representing the minimum rights necessary for accessing or viewing the axiom. Lower elements of the rights hierarchy represent higher security levels; thus, for instance, public axioms may be labeled with the supremum element 1, while confidential information should have a security level close to the infimum of the lattice.

This ontology is then accessed by users. Each user is associated also with an element of the lattice, which corresponds to its *access level*. The access level represents the rights this user has for watching axioms in the ontology: she should be able to see all the axioms having a security level that is at most as high as her access level. In other words, if s is the access level of the user, then she has access to the subontology $\mathcal{T}_s$.

Obviously, the goal of an ontology is not merely to rest alone and be observed by the users; instead, users should be able to reason over it and deduce implicit consequences from the explicitly stated axioms. However, a user with access level s should be able to see only those consequences that follow from $\mathcal{T}_s$, but not others that may be derivable from axioms with a higher security level. Thus, reasoning should depend on the user's access level.

One possible way of performing this reasoning is to first compute the subontology $\mathcal{T}_s$ and only derive consequences from it. Although such a method would yield the correct results, it is in many cases impractical and undesirable. Consider for instance the computation of the concept lattice from the ontology. This task is usually very time consuming, but may be done once and for all, and then stored in an easily accessible way. If we decide to perform reasoning with respect to each of the ontologies $\mathcal{T}_s$, then it would be necessary to compute a concept hierarchy for each access level s, and store them independently. A better strategy is simply to compute the boundary of each subsumption relation building the hierarchy.

For the boundary to behave in the desired manner, the access levels assigned to users need to be join irreducible relative to the set of labels of axioms. This restriction actually makes sense for an access control application. Indeed, elements that are not join prime have an ambiguous behaviour. Informally, if we denote as $\mathcal{P}_s$ all the consequences that can be deduced from $\mathcal{T}_s$, then an access level of the form $s_1 \oplus s_2$ refers to an individual that can see all consequences in $\mathcal{P}_{s_1} \cap \mathcal{P}_{s_2}$, but not those in $\mathcal{P}_{s_1} \setminus \mathcal{P}_{s_2}$. This is a very strange condition, to say the least.

Additionally, it is not uncommon that security and access levels change with time. A simple example of this would be financial information that becomes public after several years being confidential, or an individual that is promoted to a more trusted position. In such a case, computing the pinpointing formula as an abstract representation of the boundary appears as a good idea since changes in the labeling function that defines the security level do not affect the pinpointing formula, and hence no new reasoning, apart from updating the homomorphism from $\mathbb{B}^{\mathcal{T}}$ to the rights hierarchy, is necessary.

4.2 Fuzzy Reasoning

A totally different application of a boundary is aiding in reasoning in DLs extended with fuzzy operators. More precisely, we consider Gödel norm and its residuum. For a better understanding, we briefly recall the main concepts of fuzzy logic and instantiate them to the fuzzy $\mathcal{ALC}$.

Generally speaking, a fuzzy logic is a logic in which the crisp constructors (such as conjunction and disjunction) are replaced by fuzzy constructors, generally instantiated in the shape of t-norms and residua. These functions try to simulate the behaviour of the well-known crisp operators, but allowing for values in the whole range between 0 and 1. More formally, a *triangular norm* (or *t-norm* for short) is a function $\mathsf{t} : [0,1] \times [0,1] \to [0,1]$ that is commutative, monotone, associative and has 1 as its neutral element. A t-norm usually replaces the conjunctive operator of the logic. There exist several t-norms (see, e.g. [11] for examples) but we focus here on the *minimum* t-norm, also known as the *Gödel* t-norm, defined as $\mathsf{t}_G(a,b) = \min\{a,b\}$.

For every t-norm that is left continuous there is a unique binary operation r called the *residuum* or R-implication, such that $\mathsf{r}(x,y) = \sup\{z \mid \mathsf{t}(x,z) \leq y\}$. The name of R-implication is motivated by the fact that this operation is typically used to replace the crisp implication. With these two operators, we can then define others in such a way that desired logical properties are satisfied. For instance, fuzzy disjunction (called *triangular conorm* or *s-norm* for short) is defined as $\mathsf{s}(x,y) = 1 - \mathsf{t}(1-x, 1-y)$ as a generalization of De Morgan laws.

With these operators, we can define a variant of $\mathcal{ALC}$, in which reasoning is done with respect to fuzzy TBoxes. These TBoxes are sets of GCIs labeled with a number in $[0,1]$ representing a degree in which the GCI must be satisfied. When using the Gödel t-norm, this logic is a special case of the logic introduced in [18].

The syntax of fuzzy-$\mathcal{ALC}$ is identical to that of $\mathcal{ALC}$, except for fuzzy-GCIs which, as said before, are tuples of the form $(C \sqsubseteq D, n)$, where C, D is a GCI and $n \in [0,1]$. A fuzzy-TBox is a set of fuzzy-GCIs.

The semantics of this logic is also based on interpretations, but in this case they need to express not only whether e.g. an individual belongs to a concept, but also to which degree this is done. Thus, an interpretation $\mathcal{I}$ is a tuple $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ where $\Delta^\mathcal{I}$ is a non-empty set and $\cdot^\mathcal{I}$ is a function that maps each concept name A to a membership function $A^\mathcal{I} : \Delta^\mathcal{I} \to [0,1]$ and each role name r to a membership function $r^\mathcal{I} : \Delta^\mathcal{I} \times \Delta^\mathcal{I} \to [0,1]$. This function can then be extended to concepts using the standard fuzzy interpretation for the crisp constructors [12, 20] as follows:

$$\begin{aligned}
(C \sqcap D)^\mathcal{I}(d) &= \mathsf{t}(C^\mathcal{I}(d), D^\mathcal{I}(d)),\\
(C \sqcup D)^\mathcal{I}(d) &= \mathsf{s}(C^\mathcal{I}(d), D^\mathcal{I}(d)),\\
(\neg C)^\mathcal{I}(d) &= \mathsf{n}(C^\mathcal{I}(d)),\\
(\forall r.C)^\mathcal{I}(d) &= \inf_{e \in \Delta^\mathcal{I}}\{\mathsf{r}(r^\mathcal{I}(d,e), C^\mathcal{I}(e))\},\\
(\exists r.C)^\mathcal{I}(d) &= \sup_{e \in \Delta^\mathcal{I}}\{\mathsf{t}(r^\mathcal{I}(d,e), C^\mathcal{I}(e))\}.
\end{aligned}$$

An interpretation $\mathcal{I}$ is a *model* of the fuzzy-GCI $(C \sqsubseteq D, n)$ if and only if $\inf_{d \in \Delta^\mathcal{I}}\{\mathsf{r}(C^\mathcal{I}(d), D^\mathcal{I}(d))\} \geq n$. $\mathcal{I}$ is a model of the fuzzy-TBox $\mathcal{T}$ if it is a model of every fuzzy-GCI in $\mathcal{T}$.

We are interested in the problem of *fuzzy subsumption*: we say that a concept C is *fuzzy subsumed* by a concept D with degree $n \in [0,1]$ w.r.t. a fuzzy-

TBox $\mathcal{T}$, denoted as $(C \sqsubseteq_{\mathcal{T}} D, n)$, iff for every model $\mathcal{I}$ of $\mathcal{T}$ it holds that $\inf_{d \in \Delta^{\mathcal{I}}}\{\mathsf{r}(C^{\mathcal{I}}(d), D^{\mathcal{I}}(d))\} \geq n$.

It can be shown that if we consider Gödel t-norm t_G with its standard s-norm s_G, residuum r_G and negation n_G, then for every $n \in [0,1]$ it holds that $(C \sqsubseteq_{\mathcal{T}} D, n)$ if and only if C is (crisp) subsumed by D w.r.t. the ontology $\mathcal{T}_n = \{C \sqsubseteq D \mid (C \sqsubseteq D, m) \in \mathcal{T} \text{ with } n \leq m\}$.

In fuzzy reasoning, it is interesting to find the *best degree bound*; that is, the highest value of n such that $(C \sqsubseteq_{\mathcal{T}} D, n)$ holds. In the particular case of the Gödel norm, this problem reduces to finding the boundary with respect to the total ordering in the segment $[0,1]$, but restricted to those values that label some GCI in $\mathcal{T}$. Notice that this yields a finite total ordering, and hence the black-box approach sketched in the previous section can be used for computing the best degree bound within the same complexity bound as merely deciding subsumption w.r.t. crisp TBoxes. In particular, this means that this problem can be solved for the fuzzy logic f-$\mathcal{EL}^+$ [17] in polynomial time measured in the size of the TBox.

It is important to notice, however, that the choice of Gödel t-norm is fundamental for this approach to work. Indeed, the idempotency of this t-norm and its associated conorm is the main responsible for the generation of the finite total order that makes the definition of boundary applicable in this setting. Thus, although fuzzy logics can be defined for non-idempotent t-norms, the decision procedure obtained by computing boundaries is of no aid in them.

4.3 Possibilistic Reasoning

One other application for the computation of boundaries refers to possibilistic reasoning [6]. Similar to fuzzy reasoning, in possibilistic logic every axiom is labeled with a value in the range $(0,1]$ that expresses the certainty one has of the axiom. This kind of reasoning was introduced for dealing with inconsistent knowledge, by giving priority to formulas with a higher possibilistic degree. This property has made it a good candidate for dealing with uncertainty in DL [9, 7].

Given an ontology $\mathcal{T}$, where every axiom t has an associated certainty degree $\pi(t)$, we define, for every $n \in (0,1]$ the subontology $\mathcal{T}_n$ in the obvious way; that is, $\mathcal{T}_n = \{t \in \mathcal{T} \mid \pi(t) \geq n\}$. We can then reason about the necessity degree of distinct consequences of this knowledge base [14]: we say that C is *subsumed* by D with degree n w.r.t. $\mathcal{T}$ iff the following three conditions are satisfied: (i) $\mathcal{T}_n$ is consistent; (ii) $C \sqsubseteq_{\mathcal{T}_n} D$; and (ii) for all $m > n$, $C \not\sqsubseteq_{\mathcal{T}_m} D$.

For deciding whether a subsumption follows with a given degree n_0, the computation of a boundary is useful in two different ways. First, one needs to find the minimum n such that $\mathcal{T}_n$ is consistent. Notice that, due to Condition (i) above, for any n smaller than such a minimum, reasoning will not yield any consequence; this is known as the *drowning problem* of possibilistic logic [5]. Thus, we can actually cut the ontology to allow only those above the maximum inconsistency level. This maximum inconsistency level is in fact the boundary when considering inconsistency of ontologies as the c-property.

Once we have found the reduced consistent ontology, we need to find the maximum n such that the subsumption relation follows from $\mathcal{T}_n$. This corresponds once again to the computation of a boundary, whose result is exactly the necessity degree.

As in the previous subsection, the lattice considered for possibilistic reasoning is a total ordering, where only finitely many elements are actually relevant. Thus, possibilistic reasoning can be done within the same complexity bounds as standard DL reasoning.

One thing that must be noticed in this case is that the drowning problem is not always a desirable property, even in possibilistic reasoning. Thus, alternative definitions have been devised for such reasoning (see, e.g. [14]). Although it is not absolutely certain that boundary computation is not possible using these alternative definitions, a naïve application of the methods presented in this work yields an incorrect answer. Thus, if boundary computation is to be useful in this setting, a more elaborate approach will be necessary.

5 Conclusions and Future Work

We have presented the problem of boundary computation, which allows us to reason about portions of ontologies, depending on weights associated to the axioms. A previously known instance of boundary computation, which also worked as motivation for our general framework, is that of axiom pinpointing. We showed that the pinpointing formula can be seen as an abstract representation of all boundaries with respect to a given property. In fact, given an instance of a distributive lattice S and a labeling function lab, there is an homomorphism that maps the pinpointing formula to the boundary w.r.t. S and lab. Thus, previously known methods for computing the pinpointing formula can be directly applied to the computation of a boundary. Additionally, if the lattice S and the labeling function lab are known and fixed, the methods can be easily adapted to compute this boundary directly, without detouring through the pinpointing formula, improving this way the total execution time of the algorithm. We have shown the usefulness of our general approach by presenting three basic applications additionally to pinpointing. First, we showed how the boundary can help reasoning over a rights hierarchy, to control access to an ontology and the consequences derivable from it. Then we showed how boundaries relate to non-standard logics by showing their applicability to special cases of fuzzy- and possibilistic reasoning.

This paper gives just the first steps towards a general study of boundaries, and as such there is much work that can still be done. For one, it would be desirable to find more settings where the computation of a boundary yields a value of interest. Another area of future research is the analysis of distributive lattices and labeling functions, with the aim of finding classes where computing the boundary requires less resources than pinpointing, which is known to be hard even for very inexpressive logics. Third, we are interested in finding out whether similar ideas can be applied to more general algebraic structures, rather than distributive lattices. Such a generalization could be helpful for reasoning with respect to other t-norms, or the variants of possibilistic logic. Alternatively, it could be used to compute properties that change over time, or where the repeated use of the same axiom has a different influence on the result than a single application. Last, pinpointing has shown that it is not always necessary to compute the exact pinpointing formula, and that approximate solutions are sometimes easy to compute. We would like to investigate settings where approximate boundaries would be of interest, particularly when an approximation suffices to adequately solve a problem.

References

1. F. Baader and R. Peñaloza. Automata-based axiom pinpointing. In A. Armando, P. Baumgartner, and G. Dowek, editors, *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR 2008)*, volume 4667 of *LNAI*, pages 226–241, Sydney, Australia, 2008. Springer-Verlag.
2. F. Baader and R. Peñaloza. Axiom pinpointing in general tableaux. *Journal of Logic and Computation*, 2009. Special Issue: Tableaux'07. To appear.
3. F. Baader, R. Peñaloza, and B. Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In J. Hertzberg, M. Beetz, and R. Englert, editors, *Proceedings of the 30th German Annual Conference on Artificial Intelligence (KI'07)*, volume 4667 of *LNAI*, pages 52–67, Osnabrück, Germany, 2007. Springer-Verlag.
4. F. Baader and B. Suntisrivaraporn. Debugging SNOMED CT using axiom pinpointing in the description logic $\mathcal{EL}^+$. In *Proceedings of the 3rd Knowledge Representation in Medicine (KR-MED'08)*, volume 410 of *CEUR-WS*, 2008.
5. S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax-based entailment. In *Proc. of the 13th Int. Joint Conf. on Artificial Intelligence (IJCAI'93)*, pages 640–647, 1993.
6. D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In *Handbook of logic in artificial intelligence and logic programming (vol. 3): nonmonotonic reasoning and uncertain reasoning*, pages 439–513, New York, NY, USA, 1994. Oxford University Press, Inc.
7. D. Dubois, J. Mengin, and H. Prade. Possibilistic uncertainty and fuzzy features in description logic. A preliminary discussion. In E. Sanchez, editor, *Fuzzy logic and the semantic web*, pages 101–113. Elsevier, http://www.elsevier.com/, 2006.
8. G. Grätzer. *General Lattice Theory*. Birkhäuser, Basel, second edition edition, 1998.
9. B. Hollunder. An alternative proof method for possibilistic logic and its application to terminological logics. *Int. J. Approx. Reasoning*, 12(2):85–109, 1995.
10. A. Kalyanpur, B. Parsia, M. Horridge, and E. Sirin. Finding all justifications of OWL DL entailments. In K. Aberer, K.-S. Choi, N. F. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, volume 4825 of *LNCS*, pages 267–280, Busan, Korea, 2007. Springer-Verlag.
11. G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall PTR, 1995.
12. R. C. T. Lee. Fuzzy logic and the resolution principle. *Journal of the ACM*, 19(1):109–119, 1972.

13. T. Meyer, K. Lee, R. Booth, and J. Z. Pan. Finding maximally satisfiable terminologies for the description logic $\mathcal{ALC}$. In *Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI 2006).* AAAI Press/The MIT Press, 2006.
14. G. Qi, J. Z. Pan, and Q. Ji. Extending description logics with uncertainty reasoning in possibilistic logic. In *ECSQARU '07: Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 4724 of *LNAI*, pages 828–839, Berlin, Heidelberg, 2007. Springer-Verlag.
15. R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.
16. S. Schlobach, Z. Huang, R. Cornet, and F. Harmelen. Debugging incoherent terminologies. *Journal of Automated Reasoning*, 39(3):317–349, 2007.
17. G. Stoilos, G. B. Stamou, and J. Z. Pan. Classifying fuzzy subsumption in fuzzy-EL+. In F. Baader, C. Lutz, and B. Motik, editors, *Proceedings of the 2008 Description Logic Workshop (DL 2008)*, volume 353 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
18. U. Straccia. Reasoning within fuzzy description logics. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.
19. B. Suntisrivaraporn, G. Qi, Q. Ji, and P. Haase. A modularization-based approach to finding all justifications for OWL DL entailments. In J. Domingue and C. Anutariya, editors, *Proceedings of the 3rd Asian Semantic Web Conference (ASWC'08)*, volume 5367 of *LNCS*, pages 1–15. Springer-Verlag, 2008.
20. C. B. Tresp and R. Molitor. A description logic for vague knowledge. In *Proc. of the 13th Eur. Conf. on Artificial Intelligence (ECAI'98)*, pages 361–365, 1998.