Knowledge Integration for Description Logics

Thomas Meyer[*] and **Kevin Lee**
KRR, National ICT Australia
School of Computer Science and Engineering, UNSW, Sydney, Australia
{tmeyer,kevinl}@cse.unsw.edu.au

Richard Booth
Department of Computing, Macquarie University, Sydney, Australia
rbooth@ics.mq.edu.au

Abstract

Description logic reasoners are able to detect incoherences (such as logical inconsistency and concept unsatisfiability) in knowledge bases, but provide little support for *resolving* them. We propose to recast techniques for *propositional* inconsistency management into the description logic setting. We show that the additional structure afforded by description logic statements can be used to refine these techniques. Our focus in this paper is on the formal semantics for such techniques, although we do provide high-level decision procedures for the knowledge integration strategies discussed.

Introduction

It is becoming clear that high quality ontologies are crucial for reasoning tasks in a variety of areas. A particularly interesting case is the Semantic Web, a second-generation web in which resources are amenable to automated processing (Berners-Lee, Hendler, & Lassila 2001). Description logics (or DLs for short) are proving to be a highly successful class of knowledge representation languages with which to represent ontologies. An important issue in ontology management concerns ways of handling notions of *incoherence*, ranging from logical inconsistency to concept unsatisfiability. For example, (Schlobach & Cornet 2003) show the following incoherent ontology specification for the DICE medical terminology:

$\text{brain} \sqsubseteq \text{CentralNervousSystem}$

$\text{brain} \sqsubseteq \text{BodyPart}$

$\text{CentralNervousSystem} \sqsubseteq \text{NervousSystem}$

$\text{NervousSystem} \sqsubseteq \neg\text{BodyPart}$

According to this, a brain is a body part as well as a central nervous system, while the latter is a type of nervous system, which, in turn, is not a body part. Although not logically inconsistent, *brain* is nevertheless concept unsatisfiable.

DL reasoners like RACER (Haarslev & Möller 2001) and FaCT (Horrocks 1998) will detect such problems, but there is comparatively limited support for *resolving* incoherence. In this paper we propose an approach akin to nonmonotonic reasoning to determine the consequences of a DL knowledge base. But in the style of classical belief revision the original knowledge base is also *weakened* so that its classical consequences correspond exactly to the nonmonotonic consequences of the original knowledge base.

Much of the work in the belief revision community over the past twenty years has focused on dealing with inconsistency, and significant advances have been made (Hansson 1999). However, a serious drawback is that work in this area, by and large, is based on the propositional aspects of the logic. These techniques are directly applicable to DLs but do not exploit the additional expressivity available. We recast the work of (Benferhat *et al.* 2004) on knowledge integration to the DL setup. We demonstrate that our account generalises the propositional approach in appropriate ways. In this paper we provide basic strategies for managing *inconsistency*. We focus on the formal semantics for such strategies, but we also provide associated decision procedures.

The rest of the paper is organised as follows. We commence with a brief introduction to description logics. This is followed by a discussion of propositional knowledge integration, and some suggested modifications to the propositional approach. Then we present the knowledge integration strategies and algorithms for DLs. After a brief look at related work we conclude with a discussion of future work.

Description Logics

Description Logics are a well-known family of knowledge representation formalisms (Baader & Nutt 2003). They are based on the notions of concepts (unary predicates, classes) and roles (binary relations), and are mainly characterised by constructors that allow complex concepts and roles to be built from atomic ones. The expressive power of a DL system is determined by the constructs available for building concept descriptions, and by the way these descriptions can be used in the terminological (Tbox) and assertional (Abox) components of the system. The logics of interest to us are all based on an extension of the well-known DL $\mathcal{ALC}$ (Schmidt-Schauß & Smolka 1991). Concept descriptions are built from concept names using the constructors

[*]National ICT Australia is funded by the Australia Government's Department of Communications, Information and Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence program. It is supported by its members the Australian National University, University of NSW, ACT Government, NSW Government and affiliate partner University of Sydney.

disjunction ($C \sqcup D$), conjunction ($C \sqcap D$), negation ($\neg C$), existential restriction ($\exists R.C$) and value restriction ($\forall R.C$), where C, D stand for concepts and R for a role name. To define the semantics of concept descriptions, concepts are interpreted as subsets of a domain of interest, and roles as binary relations over this domain. An interpretation I consists of a non-empty set Δ^I (the domain of I) and a function $\cdot^I$ (the *interpretation function* of I) which maps every concept name A to a subset A^I of Δ^I, and every role name R to a subset R^I of $\Delta^I \times \Delta^I$. The interpretation function is extended to arbitrary concept descriptions as follows. Let C, D be concept descriptions and R a role name, and assume that C^I and D^I are already defined. Then $(\neg C)^I = \Delta^I \setminus C^I$,

$$(C \sqcup D)^I = C^I \cup D^I, (C \sqcap D)^I = C^I \cap D^I,$$

$$(\exists R.C)^I = \{x \mid \exists y \; s.t. \; (x, y) \in R^I \text{ and } y \in C^I\},$$

$$(\forall R.C)^I = \{x \mid \forall y, (x, y) \in R^I \text{ implies } y \in C^I\}.$$

A DL knowledge base consists of two finite and mutually disjoint sets. A *Tbox* which introduces the *terminology*, and an *Abox* which contains facts about particular objects in the application domain. Tbox statements have the form $C \sqsubseteq D$ (*inclusions*) and $C \doteq D$ (*equalities*) where C and D are (possibly complex) concept descriptions. The semantics of Tbox statements is as follows: an interpretation I *satisfies* $C \sqsubseteq D$ iff $C^I \subseteq D^I$, I *satisfies* $C \doteq D$ iff $C^I = D^I$. Objects in the Abox are referred to by a finite number of *individual names* and these names may be used in two types of assertional statements: *concept assertions* of the type $C(a)$ and *role assertions* of the type $R(a, b)$, where C is a concept description, R is a role name, and a and b are individual names. To provide a semantics for Abox statements it is necessary to add to every interpretation an injective *denotation function* d which satisfies the *unique names assumption*: it maps every individual name a to a different element a^I of the domain Δ^I (we define d separately from the interpretation function to facilitate the definition of a pre-interpretation later in the paper). An interpretation I *satisfies* the assertion $C(a)$ iff $a^I \in C^I$, and it *satisfies* $R(a, b)$ iff $(a^I, b^I) \in R^I$. I is a *model* of a DL (Tbox or Abox) statement ϕ iff it satisfies the statement, and is a model of a DL knowledge base B iff it satisfies every statement in B. The models of a statement ϕ (or knowledge base B) are denoted by $M(\phi)$ (or $M(B)$). Because any equality $C \doteq D$ is equivalent to the set of inclusions $\{C \sqsubseteq D, D \sqsubseteq C\}$ we take a Tbox to contain only inclusions. A DL knowledge base B *entails* a DL statement ϕ, written as $B \vDash \phi$, iff every model of B is a model of ϕ.

Propositional knowledge integration

Propositional knowledge integration, as described in (Benferhat *et al.* 2004), takes as input a *stratified knowledge base* $K = (S_1, \ldots, S_n)$ where, for $i \in \{1, \ldots, n\}$, S_i is a knowledge base, or finite set of propositional sentences (of a finitely generated propositional logic). Sentences in a stratum S_i are all judged to be of equal reliability, while sentences contained in a higher stratum, i.e. in an S_j for $j > i$, are seen as less reliable. In (Benferhat *et al.* 2004) the strategies proposed to minimise the loss of information that occurs when a stratified knowledge base is inconsistent are shown to yield identical results to the *lexicographic system* for knowledge integration (Benferhat *et al.* 1993). It is well-known that lexicographic entailment is a versatile system with desirable theoretical properties. For example, it has been shown in (Benferhat *et al.* 1998; Nebel 1998) that it can be used to model all classical AGM belief revision operators. We provide here an alternative semantic characterisation of lexicographic entailment based on the notion of *exceptions*. The advantage of this characterisation is that it makes a clear distinction between the following two distinct principles at work:

(Independence) Sentences in a stratum are assumed to have been obtained independently

(Precedence) More reliable information should take complete precedence over less reliable information

Independence is used implicitly in the work of (Benferhat *et al.* 2004). It is applied to sentences in each stratum S_i to obtain the preference ordering over the set V of propositional valuations associated with S_i and is formalised in terms of exceptions. The number of S_i-exceptions relative to valuation v is the number of sentences in S_i false in v; the fewer S_i-exceptions, the more preferred v will be.

Definition 1 *The number of ϕ-exceptions $e^\phi(v)$ for a valuation v is 0 if $v \in M(\phi)$ and 1 otherwise. For a finite set of sentences X, the number of X-exceptions for a valuation v is $e^X(v) = \sum_{\phi \in X} e^\phi(v)$. The ordering $\preceq_X$ on V is defined as: $v \preceq_X w$ iff $e^X(v) \leq e^X(w)$.*

It is only after the Independence principle has been applied to *all* strata that the Precedence principle is applied to the orderings associated with the different strata to obtain a lexicographically combined preference ordering.

Definition 2 *Let $K = (S_1, \ldots, S_n)$ be a stratified knowledge base and for $i \in \{1, \ldots, n\}$, let $\preceq_{S_i}$ be a total preorder on V. Then $v \preceq_{lex} w$ iff $\forall j \in \{1, \ldots, n\}$, $[v \preceq_{S_j} w$ or $v \prec_{S_i} w$ for some $i < j]$.*

Definition 2 constructs $\preceq_{lex}$ by starting with $\preceq_{S_1}$ and keeps on refining it with strata orderings of lower preference. We use $\preceq_{lex}$ to define lexicographic entailment:

Definition 3 *A stratified knowledge base K lexicographically entails ϕ, written $K \vDash_{lex} \phi$, iff the $(\preceq_{lex})$-minimal models satisfy ϕ.*

The view of lexicographic entailment as an application of Independence and Precedence is also present in a new strategy for knowledge integration, *conjunctive maxi-adjustment* or CMA, that we propose in Algorithm 1. The idea is to work through K stratum by stratum in order of decreasing precedence, and to construct a consistent knowledge base B, adding as many sentences as possible while maintaining consistency, and weakening those strata responsible for inconsistencies. In this sense it is similar to *whole disjunctive maxi-adjustment* (Benferhat *et al.* 2004), but the way in which strata are weakened is (syntactically) different. With CMA, if S_i is inconsistent with the part of B constructed so far, it is replaced by the disjunction of the cardinality-maximal conjunctions of S_i-formulas consistent with B.
The following example is a simple demonstration of CMA.

Algorithm 1 Conjunctive maxi-adjustment (CMA)

Input: $K = (S_1, \ldots, S_n)$
Output: A consistent knowledge base
$B := \emptyset$
for $i := 1$ to n **do**
  $j := |S_i|$
  repeat
    $\phi := \bigvee$ of all $\bigwedge$s of size j of formulas of S_i
    $j := j - 1$
  until $B \cup \{\phi\}$ is consistent or $j = 0$
  if $B \cup \{\phi\}$ is consistent **then**
    $B := B \cup \{\phi\}$
  end if
end for
return B

Example 1 *Let $K = (S_1, S_2)$, $S_1 = \{\neg(p \wedge q), \neg(q \wedge r), \neg(p \wedge r)\}$ and $S_2 = \{p, q, r\}$. S_1 is consistent, so B is set to S_1. S_2 is inconsistent with B, so ϕ is set to $(p \wedge q) \vee (p \wedge r) \vee (q \wedge r)$. ϕ is inconsistent with B, so the latter is weakened further by setting it to $p \vee q \vee r$. Now ϕ is consistent with B so B is set to $\{\neg(p \wedge q), \neg(q \wedge r), \neg(p \wedge r), p \vee q \vee r\}$.*

It is easily verified that Algorithm 1 always terminates. Furthermore, it produces results that are equivalent to lexicographic entailment, and therefore also to the strategies discussed in (Benferhat *et al.* 2004).

Proposition 1 *Let K be a stratified knowledge base and B the knowledge base obtained from K by Algorithm 1. Then $K \vDash_{lex} \phi$ iff $B \vDash \phi$.*

The principal reason for the introduction of CMA is that the two algorithms for knowledge integration described in the next section are natural extensions of it.

For the rest of the paper we modify our definition of a stratified knowledge base so that each stratum is a *multi-set* of sentences, denoted by square brackets (so a stratum has the form $[\phi_1, \ldots, \phi_n]$). This allows us to prove a result which does not hold if strata are represented as sets.

Proposition 2 *For $K = (S_1, \ldots, S_n)$, $K' = (S'_1, \ldots, S'_n)$ and every $i \in \{1, \ldots, n\}$, let there be a bijection f_i between S_i and S'_i s.t. $f_i(\phi) \equiv \phi$. Then $K \vDash_{lex} \psi$ iff $K' \vDash_{lex} \psi$.*

Knowledge integration for description logics

In this section we recast the techniques of propositional knowledge integration to DLs, with the input being a stratified DL knowledge base $K = (S_1, \ldots, S_n)$ where, for $i \in \{1, \ldots, n\}$, S_i is a finite multi-set of DL sentences. We present versions of lexicographic entailment and the CMA strategy for DL knowledge integration obtained from a direct conversion of the techniques used in the propositional case. We argue that these versions do not exploit the expressivity of DLs adequately, and refine both lexicographic entailment and CMA. We show that the refined versions of lexicographic entailment and CMA produce identical results.

But before doing so we need to deal with a number of technical issues. The first one concerns the CMA strategy and the level of expressivity in DL languages. Recall that propositional CMA makes use of disjunctions of conjunctions of sentences. But DL languages do not allow disjunctions of Tbox sentences with Abox sentences, and an expression such as $(C \sqsubseteq D) \sqcup D(a)$ is thus ill-formed. To deal with this issue we introduce the notion of a *disjunctive DL knowledge base*, or DKB, as a set of DL knowledge bases. The semantics of DKBs is defined as follows.

Definition 4 *A DKB $\mathcal{B}$ is* satisfied *by an interpretation I (I is a model of $\mathcal{B}$) iff I is a model of at least one of the elements of $\mathcal{B}$. $\mathcal{B}$* entails *a DKB Φ ($\mathcal{B} \vDash \Phi$) iff every model of $\mathcal{B}$ is a model of Φ.*

Informally $\mathcal{B}$ can be read as the disjunction of its elements, with a single element of $\mathcal{B}$ viewed as the conjunction of the sentences contained in it. For example, $\{[C \sqsubseteq D, C(a)], [C \sqsubseteq D, D(a)]\}$ states that both $C \sqsubseteq D$ and $C(a)$ hold, or that both $C \sqsubseteq D$ and $D(a)$ hold. There are more fundamental reasons for the use of DKBs as well, which will briefly be touched on in the conclusion.

The next issue to consider is the comparability of DL interpretations. The semantics of propositional lexicographic entailment makes use of the fact that all propositional valuations are "possible worlds" that can all be compared with respect to preference. But the additional structure of DL interpretations makes it impossible to maintain this comparability in the semantics. In particular, whenever two interpretations have different domains or do not map the same individual names to the same elements in the domain, it is counterintuitive to insist that they be comparable in terms of preference. This issue is solved by requiring that only interpretations obtained from the same *pre-interpretation* be comparable. A pre-interpretation is an ordered pair $\pi = (\Delta^\pi, d^\pi)$, where Δ^π is a domain and d^π is a denotation function. Let Π be the class of all pre-interpretations. For every pre-interpretation $\pi = (\Delta^\pi, d^\pi)$, let $\mathcal{I}^\pi$ be the class of interpretations I with $\Delta^I = \Delta^\pi$ and $d^I = d^\pi$. We provide a semantics similar to that of propositional lexicographic entailment. But each ordering $\preceq_{S_i}$ on valuations associated with a stratum S_i will, in the case of DLs, be replaced by a class of orderings $\preceq^\pi_{S_i}$: one for each pre-interpretation π in Π. For a fixed π, the orderings $\preceq^\pi_{S_i}$ for $i \in \{1, \ldots, n\}$ are then lexicographically combined using Definition 2 to obtain the ordering $\preceq^\pi_{lex}$. Lexicographic entailment is then defined in terms of the minimal models of all these orderings. That is, given a preference ordering $\preceq^\pi_{lex}$ for each $\pi \in \Pi$, lexicographic entailment for stratified DL knowledge bases is defined as follows:

($\vDash_{lex}$) $K \vDash_{lex} \Phi$ iff $\bigcup_{\pi \in \Pi} \min_{\preceq^\pi_{lex}} \subseteq M(\Phi)$

where $min_{\preceq^\pi_{lex}}$ refers to the ($\preceq^\pi_{lex}$)-minimal models. The one remaining question is how the preference orderings $\preceq^\pi_{S_i}$ used in the construction of $\preceq^\pi_{lex}$ should be obtained. A first attempt is to use the same technique as that used for propositional lexicographic entailment. That is, for each $I \in \mathcal{I}^\pi$ and each stratum S_i, let the number of S_i-exceptions w.r.t. I be the number of sentences in S_i falsified by I, and use these exceptions to generate the ordering $\preceq^\pi_{S_i}$.

Definition 5 *Let $\pi \in \Pi$, $I \in \mathcal{I}^\pi$, ϕ a DL statement, and X a multi-set of DL statements. The number of ϕ-exceptions*

$e^{\phi}(I)$ for I is 0 if I satisfies ϕ and 1 otherwise. The number of X-exceptions for I is: $e^X(I) = \sum_{\phi \in X} e^{\phi}(I)$. The ordering $\preceq^{\pi}_X$ on $\mathcal{I}^{\pi}$ is defined as: $I \preceq^{\pi}_X J$ iff $e^X(I) \leq e^X(J)$.

The DL version of CMA is presented in Algorithm 2. It is a compilation of DL lexicographic entailment.

Algorithm 2 CMA for DLs (CMA-DL)

Input: $K = (S_1, \ldots, S_n)$
Output: A consistent DKB
$\mathcal{B} := \{\emptyset\}$
for $i := 1$ to n **do**
  $\mathcal{C} := \mathcal{B}$
  for all $B \in \mathcal{C}$ **do**
    $j := |S_i|$
    repeat
      $\mathcal{X} := \{X \mid X \subseteq S_i$ and $|X| = j\}$
      $j := j - 1$
    until $B \cup X$ is consistent for some $X \in \mathcal{X}$
    $\mathcal{B} := (\mathcal{B} \setminus \{B\}) \cup$
        $\{B \cup X \mid (X \in \mathcal{X})\ \&(B \cup X)$ is consistent$\}$
  end for
end for
return $\mathcal{B}$

Proposition 3 *Let K be a stratified DL knowledge base, $\mathcal{B}$ the DKB obtained from K by CMA-DL in Algorithm 2, let lexicographic entailment for DLs be defined in terms of Definition 5, and let Φ be a DKB. Then $K \vDash_{lex} \Phi$ iff $\mathcal{B} \vDash \Phi$.*

To see that Algorithm 2 always terminates, note that elements of $\mathcal{C}$ are always consistent by construction. If j is ever set to 0 in the **repeat** loop, $\mathcal{X}$ will be set to $\{\emptyset\}$, and then $B \cup X$ *has* to be consistent, since $B \in \mathcal{C}$. The following example demonstrates how Algorithm 2 works.

Example 2 *Let $K = (S_1, S_2)$, $S_1 = [C \sqsubseteq \neg D, C \sqsubseteq \neg E, D \sqsubseteq \neg E]$ and $S_2 = [C(a), D(a), E(a)]$. S_1 is consistent, so $\mathcal{B}$ (and $\mathcal{C}$) is set to $\{S_1\}$. Now B is set to the only element of $\mathcal{C}$: $[C \sqsubseteq \neg D, C \sqsubseteq \neg E, D \sqsubseteq \neg E]$. S_2 is inconsistent with B so, during the second iteration of the* **repeat** *loop, $\mathcal{X}$ is set to $\{[C(a), D(a)], [C(a), E(a)], [D(a), E(a)]\}$. Every element of $\mathcal{X}$ is inconsistent with B, and the next iteration of the* **repeat** *loop sets $\mathcal{X}$ to $\{[C(a)], [D(a)], E(a)]\}$. Now all elements of $\mathcal{X}$ are consistent with B, so B is removed from $\mathcal{B}$ and replaced with three multi-sets, yielding:*

$$\begin{aligned}\mathcal{B} = \{&[C \sqsubseteq \neg D, C \sqsubseteq \neg E, D \sqsubseteq \neg E, C(a)],\\ &[C \sqsubseteq \neg D, C \sqsubseteq \neg E, D \sqsubseteq \neg E, D(a)],\\ &[C \sqsubseteq \neg D, C \sqsubseteq \neg E, D \sqsubseteq \neg E, E(a)]\}.\end{aligned}$$

Thus $\mathcal{B}$ states that the three statements $C \sqsubseteq \neg D$, $C \sqsubseteq \neg E$, and $D \sqsubseteq \neg E$ hold, and that, in addition, at least one of $C(a)$, $D(a)$ or $E(a)$ holds. A consequence of all these statements is that exactly *one of $C(a)$, $D(a)$ or $E(a)$ holds.*

Lexicographic entailment for DLs and the CMA-DL strategy are both faithful translations of their propositional counterparts. It is precisely because of this that they do not take the structure of DL statements into account. The following example illustrates this deficiency.

Example 3 *Let $K = (S_1, S_2)$, where $S_1 = [bird(tweety), \neg flies(tweety), bird(chirpy)]$, and $S_2 = [bird \sqsubseteq flies]$. S_1 is consistent, and so $\mathcal{B}$ is set to $\{S_1\}$. S_2 is inconsistent with S_1, the only element of $\mathcal{B}$, and so $\mathcal{B}$ is returned as $\{[bird(tweety), \neg flies(tweety), bird(chirpy)]\}$.*

When Algorithm 2 is applied to K from Example 3, it concludes, correctly, that Tweety is a non-flying bird and that Chirpy is a bird. But it does *not* conclude that Chirpy flies since it has discarded the statement $bird \sqsubseteq flies$. It therefore does not exploit the structure of $bird \sqsubseteq flies$ appropriately. Ideally we should be able to conclude that Tweety is an *exception* and that all birds other than Tweety (including Chirpy) can fly. For this to be possible we need to weaken Tbox statements such as $bird \sqsubseteq flies$, something that is not possible in the propositional case. Semantically we effect such a weakening by modifying the definition of exceptions in Definition 5. For Abox statements the definition stays unchanged, but for Tbox statements the number of exceptions will be *the number of elements in the domain violating the statement*. An element in the domain of an interpretation I violates a statement of the form $C \sqsubseteq D$ if it is in C^I but not in D^I, i.e. if it is in $C^I \cap (\neg D^I)$.

Definition 6 *Let $\pi \in \Pi$, $I \in \mathcal{I}^{\pi}$, ϕ a DL statement, and X a multi-set of DL statements. If ϕ is an Abox statement, the number of ϕ-exceptions $e^{\phi}(I)$ for an interpretation I is 0 if I satisfies ϕ and 1 otherwise. If ϕ is a Tbox statement of the form $C \sqsubseteq D$, the number of ϕ-exceptions for I is:*

$$e^{\phi}(I) = \begin{cases} \left|C^I \cap (\neg D)^I\right| & \text{if } C^I \cap (\neg D)^I \text{ is finite},\\ \infty & \text{otherwise.}\end{cases}$$

The number of X-exceptions for I is $e^X(I) = \sum_{\phi \in X} e^{\phi}(I)$. The ordering $\preceq^{\pi}_X$ on $\mathcal{I}^{\pi}$ is: $I \preceq^{\pi}_X J$ iff $e^X(I) \leq e^X(J)$.

So $\preceq^{\pi}_X$ is a version of cardinality-based circumscription (Liberatore & Schaerf 1995): the more exceptions, the less preferred an interpretation, while interpretations with an infinite number of exceptions are all equally bad.

Using Definition 6 in our construction of lexicographic entailment will ensure that we will be able to conclude, in Example 3, that Chirpy can fly. However, we are still not able to express the conclusion that *all* birds, except for Tweety, can fly. The problem is that the notion of an exception is not expressible in a DL. We cannot state that all birds, with the exception of one, can fly. It is necessary to extend the level of expressivity of the DL languages we are interested in. An appropriate extension, adding *cardinality restrictions* on concepts, was proposed in (Baader, Buchheit, & Hollander 1996). There, its introduction was motivated by the use of DL systems for solving configuration tasks. These restrictions are statements in the Tbox, allowing one to express restrictions on the number of elements a concept may have: $(\geq m\ C)$ and $(\leq n\ C)$ respectively express that the concept C has at least m elements and at most n elements. For our purposes it is sufficient to consider cardinality restrictions of the form $(\leq n\ C)$.

An interpretation I is said to *satisfy* a restriction of the form $(\leq n\ C)$ iff $\left|C^I\right| \leq n$. The statement $C \sqsubseteq D$ is equivalent to stating that the concept $C \sqcap \neg D$ is empty, i.e. that $(\leq 0\ C \sqcap \neg D)$. This demonstrates that the Tbox statements we have considered thus far can all be expressed as cardinality restrictions. Therefore, a Tbox will from now on be a finite multi-set of cardinality restrictions. An interpretation I is a *model* of such a Tbox iff it satisfies each of its restrictions. Other semantic notions such as entailment are extended in the obvious way. With the inclusion of cardinality restrictions we can now rephrase S_2 in Example 3 as $\{(\leq 0\, bird \sqcap \neg flies)\}$. And, using Definition 6, K now lexicographically entails that Tweety is a non-flying bird, that Chirpy is a flying bird, and that there is at most one non-flying bird, $(1 \leq bird \sqcap \neg flies)$, which is a *weakening* of S_2. So it follows that, barring Tweety, all birds can fly.

The next step is to refine the CMA-DL strategy to coincide with the modified version of lexicographic entailment for DLs. This strategy, referred to as *refined CMA-DL*, is described in Algorithm 3. The main difference between the two algorithms is in the construction of $\mathcal{X}$. Abox sentences are treated exactly as in Algorithm 2: the j-weakening $W^j(B^A)$ of the Abox B^A of a DL knowledge base B (where $j \leq |B^A|$), contains all those sub multi-sets of B^A where j elements have been *removed*. That is, $W^j(B^A) = \{B^A \setminus Y \mid Y \subseteq B^A \text{ and } |Y| = j\}$. So, for $B^A = [C(a), D(a), E(a)]$, $W^1(B^A) = \{[C(a), D(a)], [C(a), E(a)], [D(a), E(a)]\}$. For a Tbox sentence τ of the form $(\leq n\ C)$, let $W(\tau) = \{(\leq (n + j)\ C) \mid j \geq 0\}$. That is $W(\tau)$ is the set of all weakened versions of τ. Furthermore, for a Tbox sentence τ of the form $(\leq n\ C)$ and $\tau' \in W(\tau)$ of the form $(\leq m\ C)$, we let $w^\tau(\tau') = m - n$. So $w^\tau(\tau')$ measures the extent to which τ' is a weakening of τ. For $j \geq 0$, the j-weakening $W^j(B^T)$ of the Tbox B^T of a DL knowledge base B contains all those weakened versions of B^T for which the sum of the extent of the weakening is j. That is, for $B^T = [\tau_1, \ldots, \tau_n]$, $W^j(B^T) = \{[\tau'_1, \ldots, \tau'_n] \mid \sum_{i=1}^n w^{\tau_i}(\tau'_i) = j\}$. For example, for $B^T = [(\leq 0\ C), (\leq 0\ D)]$, $W^2(B^T) = \{[(\leq 0\ C), (\leq 2\ D)], [(\leq 1\ C), (\leq 1\ D)], [(\leq 2\ C), (\leq 0\ D)]\}$. And for $j \geq 0$, the j-weakening $\mathcal{W}^j(B)$ of a DL knowledge base B contains all combinations of i-weakenings of B^A and k-weakenings of B^T for which i and k add up to j. That is,

$$\mathcal{W}^j(B) = \left\{ A \cup T \;\middle|\; \begin{array}{l} A \in W^i(B^A), T \in W^k(B^T), \\ i \leq |B^A|, \text{ and } j = i + k \end{array} \right\}.$$

For example, if $B = [(\leq 0\ C), C(a), C(b)]$, then

$$\mathcal{W}^2(B) = \left\{ \begin{array}{l} [(\leq 1\ C), C(a)], [(\leq 1\ C), C(b)], \\ [(\leq 0\ C)], [(\leq 2\ C), C(a), C(b)] \end{array} \right\}.$$

So $\mathcal{W}^2(B)$ contains those weakenings of B in which exactly two exceptions occur. The j-weakenings of DL knowledge bases are used in the **repeat** loop of Algorithm 3 where $\mathcal{X}$ is set to the j-weakening of S_i. As required, RCMA-DL is a compilation of lexicographic entailment using Definition 6.

Proposition 4 *Let K be a stratified DL knowledge base, $\mathcal{B}$ the DKB obtained from K by RCMA-DL in Algorithm 3, let lexicographic entailment for DLs be defined in terms of Definition 6, and let Φ be a DKB. Then $K \vDash_{lex} \Phi$ iff $\mathcal{B} \vDash \Phi$.*

The proof that Algorithm 3 terminates hinges on the fact that weakenings of Tbox sentences allow for *more* exceptions, and the fact that the maximum number of exceptions to cater for is bounded by the number of individual names occurring in K. The example below demonstrates Algorithm 3 (where b, f, t and c abbreviate $bird$, $flies$, $tweety$ and $chirpy$).

Algorithm 3 Refined CMA-DL (RCMA-DL)

Input: $K = (S_1, \ldots, S_n)$
Output: A consistent DKB
$\mathcal{B} := \{\emptyset\}$
for $i := 1$ to n **do**
  $\mathcal{C} := \mathcal{B}$
  for all $B \in \mathcal{C}$ **do**
    $j := 0$
    repeat
      $\mathcal{X} := \mathcal{W}^j(S_i)$
      $j := j + 1$
    until $B \cup X$ is consistent for some $X \in \mathcal{X}$
    $\mathcal{B} := (\mathcal{B} \setminus \{B\}) \cup$
        $\{B \cup X \mid (X \in \mathcal{X})\ \&(B \cup X) \text{ is consistent}\}$
  end for
end for
return $\mathcal{B}$

Example 4 *Let $K = (S_1, S_2)$, $S_1 = [b(t), b(c)]$, and $S_2 = [\neg f(t), \neg f(c), (\leq 0\, b \sqcap \neg f)]$. S_1 is consistent so RCMA-DL sets $\mathcal{B}$ (and $\mathcal{C}$) to $\{S_1\}$. Now B is set to the only element of $\mathcal{C}$: $[b(t), b(c)]$. S_2 is inconsistent with B, so $\mathcal{X}$ is set to $\mathcal{W}^1(S_2) = \{[\neg f(t), \neg f(c), (\leq 1\ b \sqcap \neg f)], [\neg f(t), (\leq 0\ b \sqcap \neg f)], [\neg f(c), (\leq 0\ b \sqcap \neg f)]\}$. Every element of $\mathcal{X}$ is inconsistent with B, and so $\mathcal{X}$ is set to $\mathcal{W}^2(S_2) = \{[\neg f(t), \neg f(c), (\leq 2\ b \sqcap \neg f)], [\neg f(t), (\leq 1\ b \sqcap \neg f)], [\neg f(c), (\leq 1\ b \sqcap \neg f)], [(\leq 0\ b \sqcap \neg f)]\}$. Now all the elements of $\mathcal{X}$ are consistent with B, resulting in a $\mathcal{B}$ containing $B \cup X$ for every X in $\mathcal{X}$. Combined, the four elements of the DKB $\mathcal{B}$ show that exactly one of the following four cases hold: a) Tweety and Chirpy are the only two non-flying birds; b) Tweety is the only non-flying bird; c) Chirpy is the only non-flying bird; d) All birds fly, including Tweety and Chirpy.*

Related work

One of the first attempts to deal with inconsistency in logic-based terminological systems can be found in (Nebel 1990), where it is phrased as a belief revision problem. More recently the solution of (Schlobach & Cornet 2003) is to provide support for inconsistency by correcting it. They propose a non-standard reasoning system for debugging inconsistent terminologies. The idea is to provide an explanation by pinpointing the source of the inconsistency, while correction is left to human experts. In contrast, the approach taken in (Huang *et al.* 2005) assumes that ontology reparation will be too difficult. They propose to tolerate inconsistency and apply a non-classical form of inference to obtain meaningful results. Our approach is a hybrid of these. We employ a version of lexicographic entailment to determine the consequences of an inconsistent DL knowledge base, but the original knowledge base is also *weakened* so that its classical consequences correspond exactly to the nonmonotonic consequences of the original knowledge base.

In (Quantz & Royer 1992) a technique is described for assigning a preference semantics for defaults in terminological logics which uses exceptions, and therefore has some similarities to our work. They draw a distinction between strict inclusions (Tbox statements of the form $A \sqsubseteq B$) and *defaults* which is interpreted as "soft" inclusions. In our framework this distinction can be modelled with two strata in which all strict inclusions occur in S_1 and all soft inclusions in S_2. In this sense our framework is more expressive than theirs. More importantly, their formal semantics is not cardinality-based, and therefore yields quite different results from ours. And finally, unlike us, they do not provide a weakening of the original knowledge base.

An altogether different approach is the explicit introduction of nonmonotonicity into DLs, usually some variant of default logic. See (Baader, Küsters, & Wolter 2003) for an overview. While it is difficult to draw direct comparisons with our work, similar intuitions might be identified and exploited.

Conclusion

We have proposed knowledge integration strategies for DLs based on techniques developed in the propositional case, and provided corresponding algorithms with disjunctive knowledge bases (DKBs) as output. It can be shown that the elements of a DKB produced as the output of Algorithm 2 or 3 are always pairwise inconsistent (modulo logical equivalence), a property which is useful (i) when inconsistency management is an iterative process and (ii) as part of support provision for ontology engineers. This forms the basis of an argument that the structure of DKBs are important and ought to be retained as outputs of our algorithms.

We have shown how the structure of DL languages can be exploited to define basic knowledge integration strategies. Our focus was on the formal semantics of knowledge integration strategies, although we also provided high-level decision procedures. The next step is the development of tableaux-based algorithms for implementing the strategies outlined in the paper. Some initial results suggest that the complexity of the integration strategies may be no worse than consistency checking in the DL under consideration.

An obvious question to consider is whether any additional structure, such as the specification of role hierarchies and transitivity of roles, can be exploited further to modify the knowledge integration strategies in appropriate ways. Such additional structure might also be used to ameliorate other problems. For example, DKBs can be exponential in size, which will severely affect their use in practice. But it might be possible to limit the size of strata in stratified KBs using the structure of sentences contained in it. A simple example is the use of principles such as *specificity*: if the sentences $bird \sqsubseteq flies$ (all birds fly) and $penguin \sqsubseteq \neg flies$ (all penguins don't fly) occur in the same stratum, an application of specificity will ensure that $penguin \sqsubseteq \neg flies$ is given precedence over $bird \sqsubseteq flies$, provided that $penguin \sqsubseteq bird$ takes precedence over both. Finally, the management of other notions of incoherence, such as concept unsatisfiability, is currently the topic of further investigation.

References

Baader, F., and Nutt, W. 2003. Basic description logics. In Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P., eds., *The Description Logic Handbook*.

Baader, F.; Buchheit, M.; and Hollander, B. 1996. Cardinality restrictions on concepts. *Artificial Intelligence* 88:195–213.

Baader, F.; Küsters, R.; and Wolter, F. 2003. Extensions to description logics. In Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P., eds., *The Description Logic Handbook*.

Benferhat, S.; Cayrol, C.; Dubois, D.; Lang, J.; and Prade, H. 1993. Inconsistency management and prioritized syntax-based entailment. In Bajcsy, R., ed., *Proceedings of IJCAI-93*, 640–645.

Benferhat, S.; Dubois, D.; Lang, J.; Prade, H.; Saffioti, A.; and Smets, P. 1998. A general approach for inconsistency handling and merging information in prioritized knowledge bases. In *Proceedings of KR'98*, 466–477.

Benferhat, S.; Kaci, S.; Le Berre, D.; and Williams, M. 2004. Weakening conflicting information for iterated revision and knowledge integration. *Artificial Intelligence* 153:339–371.

Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. The semantic web. *Scientific American* 284(5):3443.

Haarslev, V., and Möller, R. 2001. Racer system description. In Goré, R.; Leitsch, A.; and Nipkow, T., eds., *IJCAR 2001, LNAI 2100*.

Hansson, S. O. 1999. *A Textbook of Belief Dynamics*.

Horrocks, I. 1998. The FaCT system. In de Swart, H., ed., *Tableaux '98, LNAI 1397*, 307–312.

Huang, Z.; van Harmelen, F.; ten Teije, A.; Groot, P.; and Visser, C. 2005. Reasoning with inconsistent ontologies: Framework and prototype. URL: `http://wasp.cs.vu.nl/sekt/download/sekt341.pdf`.

Liberatore, P., and Schaerf, M. 1995. Relating belief revision and circumscription. In *Proceedings of IJCAI-95*, 1557–1563.

Nebel, B. 1990. *Reasoning and Revision in Hybrid Representation Systems*. LNAI 422. Springer-Verlag.

Nebel, B. 1998. How hard is it to revise a belief base. In Dubois, D., and Prade, H., eds., *Belief Change*, Handbook of Defeasible Reasoning and Uncertainty Management Systems. Kluwer. 77–145.

Quantz, J., and Royer, V. 1992. A preference semantics for defaults in terminological logics. In *Proceedings of KR'92*, 294–305.

Schlobach, S., and Cornet, R. 2003. Non-standard reasoning services for the debugging of description logic terminologies. In *Proceedings of IJCAI 2003*, 355–360.

Schmidt-Schauß, M., and Smolka, G. 1991. Attributive concept descriptions with complements. *Artificial Intelligence* 48:1–26.