Preferential Default Reasoning on the Semantic Web

Rafael Kiesel, Erman Acar and Stefan Schlobach

Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
rafael.kiesel@web.de, {erman.acar, k.s.schlobach}@vu.nl

Abstract. The world of the Semantic Web includes many inconsistent knowledge bases, whose axioms are often unreliable and can therefore be naturally interpreted as defaults. In this article we introduce an extension of Sengupta et. al.'s semantics [12] with a preference order on the defaults with respect to satisfaction. In particular, the semantics we propose is an improved version of Heymans et. al.'s semantics [8] for a defeasible ontology language. One crucial difference we adopt is that the preference relation between interpretations is not only defined with respect to the same individuals, but any. We provide an algorithm to translate the entailment problem with respect to ordered default knowledge bases to the one for classical knowledge bases, and show its soundness and completeness. As a practical application scenario, we test our algorithm on various subsets of LOD Cloud, and report on our result.

Keywords: Default Reasoning · Inconsistency · LOD Laundromat.

1 Introduction

Many real world ontologies in the Semantic Web contain inconsistent data. These inconsistencies arise due to different reasons. One reason is their extreme size, a second is that they model very general domain knowledge with diversifying information, e.g., encyclopaedic knowledge [4]. Hence, it is challenging to maintain the consistency while keeping the relevant information accurate and reliable.

In order to integrate the knowledge across different ontologies, one often confronts the problem of alignment which is the process of determining correspondences between matching entities [6]. The resulting ontology can also be highly inconsistent due to the diverse domain knowledge that subontologies capture e.g., medical knowledge[1] and marine knowledge[2]. Since it is often considered expensive or impractical to have a domain expert to debug the resulting ontology, automatic handling of inconsistencies is an important task.

The automatic approaches usually involve a change in the classical semantics to be able to cope with inconsistent knowledge. One particular class of approaches avoids inconsistency by decreasing the number of derivable facts [12, 10, 8, 11]. Our approach falls into the same category. To this end, one first identifies which axioms cause the inconsistencies and weakens their strength e.g., by replacing them with so called *normal defaults*. However, reasoning with normal defaults alone (as in [12]) is not sufficient

[1] http://www.ihtsdo.org/our-standards/

[2] http://www.ics.forth.gr/isl/MarineTLO/

when it comes to the large amount of information that can be entailed. This is because checking entailment in such semantics in real-life scenarios is often infeasible due to high complexity [12].

In this article we introduce an extension of Sengupta et. al.'s semantics [12] with a preference order on the defaults with respect to satisfaction. We argue that, for a given set of defaults, one may have a preference on which defaults should be satisfied. Indeed, this preference may be due to the reliability of the defaults, or due to the importance of the information entailed by the defaults, or any other particular reason. Our semantics is an improved version of Heymans et. al.'s semantics [8] for a defeasible ontology language. In our semantics, interpretations are not only preferred when they satisfy a better default for the same, but also for any other individual. By introducing this preference order on the defaults one can increase the amount of information entailed by a default knowledge base while keeping it consistent. Moreover, it can greatly reduce the number of models of a default knowledge base, resulting in a possibly faster way of checking entailment.

Furthermore, we provide an algorithm to transform the entailment problem from the ordered default knowledge bases to the one for classical knowledge bases. We also show the soundness and completeness of our algorithm. We test it on different real-world semantic datasets i.e., subsets of the Linked Open Data Cloud (LOD Cloud) [7], and report our findings.

In the following we first go over some preliminaries for description logics with defaults. In Section 3, the introduced semantics is extended to implement a preference order on defaults. After presenting some theoretical results, we discuss advantages of this approach and test it on real data, in Section 4. In Section 5 we discuss related work. Conclusion and future work close the paper.

2 Preliminaries

Our semantics is a generalisation of the semantics presented in Sengupta et. al. [12] for defaults in description logics. For convenience, we adopt their notation. We note that the choice of a particular description logic is not relevant for our work, as long as it is decidable. Moreover, we assume that the reader is familiar with the basics of description logics, otherwise we refer the reader to [1] which is the standard text.

The semantics presented in [12] is defined for a knowledge base in combination with a set of *free defaults*. Given two concepts A and B of a knowledge base Σ, a free default $A \sqsubseteq_d B$ represents the rule that every named individual, which is a member of A is also a member of B, unless we derive $\neg B$. Then, a default (description logic) knowledge base is $\langle \Sigma, \mathcal{D} \rangle$, where Σ is a knowledge base and $\mathcal{D}$ is a finite set of free defaults.

Given a default knowledge base, the notion of an interpretation $\mathcal{I}$ is extended to the set of free defaults as $\mathcal{D}^{\mathcal{I}} = (\mathcal{X}_1^{\mathcal{I}}, \ldots, \mathcal{X}_n^{\mathcal{I}})$, where $\mathcal{X}_i^{\mathcal{I}}$ contains the interpreted set of named individuals that satisfy the i^{th} default. That is, for $C_i \sqsubseteq_d D_i$, the definition of $\mathcal{X}_i^{\mathcal{I}}$ is $((\neg C_i)^{\mathcal{I}} \cup D_i^{\mathcal{I}}) \cap \Delta_{ind}^{\mathcal{I}})$.[3] Here, $\Delta_{ind}^{\mathcal{I}}$ is the set of interpreted named individuals of

[3] This definition is syntactically different from the original definition, but semantically equivalent.

the knowledge base Σ. In the following we always assume $|\mathcal{D}| = n$ and $\mathcal{D} = \{(C_1 \sqsubseteq_d D_1), \ldots, (C_n \sqsubseteq_d D_n)\}$. Models of a default knowledge base $\langle \Sigma, \mathcal{D} \rangle$ are defined using a preferred model semantics; that is, the preference relation $>_{\Sigma,\mathcal{D}}$ on the interpretations is defined as follows: Given two interpretations $\mathcal{I}$ and $\mathcal{J}$ of the knowledge base Σ, we say that $\mathcal{I}$ is preferred to $\mathcal{J}$, denoted by $\mathcal{I} >_{\Sigma,\mathcal{D}} \mathcal{J}$, iff

- for all named individuals a appearing in Σ, it holds that $a^{\mathcal{I}} = a^{\mathcal{J}}$,
- $\mathcal{X}_i^{\mathcal{J}} \subseteq \mathcal{X}_i^{\mathcal{I}}$ for all $1 \leq i \leq n$, and
- $\mathcal{X}_i^{\mathcal{J}} \subset \mathcal{X}_i^{\mathcal{I}}$ for some $1 \leq i \leq n$.

Moreover, an interpretation $\mathcal{I}$ is a *d-model* of $\langle \Sigma, \mathcal{D} \rangle$ iff

$i.$ $\mathcal{I}$ (classically) satisfies Σ,
$ii.$ $C_i^{\mathcal{I}} \setminus \Delta_{ind}^{\mathcal{I}} \subseteq D_i^{\mathcal{I}}$ for each $(C_i \sqsubseteq_d D_i) \in \mathcal{D}$,
$iii.$ there is no $\mathcal{J} >_{\Sigma,\mathcal{D}} \mathcal{I}$, such that $\mathcal{J}$ also satisfies conditions i and ii.

If a default knowledge base has a d-model, we say it is *d-satisfiable*. An axiom α is then *d-entailed* by a default knowledge base $\langle \Sigma, \mathcal{D} \rangle$, written $\langle \Sigma, \mathcal{D} \rangle \models \alpha$, if it holds for every d-model $\mathcal{I}$ of the default knowledge base that $\mathcal{I}$ satisfies α.

3 Generalisation of the Semantics of Free Defaults to Ordered Defaults

Before introducing our approach we motivate it by discussing an example. Thereafter, we show that d-satisfiability and d-entailment is decidable. In particular, to prove the decidability of d-entailment, we present an algorithm and show its sound- and completeness.

3.1 Motivation

Consider the following scenario: We are given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ shown in Table 1, and we want to align them with regard to the $\equiv$ statements in the third column of Table 1. Without further debugging, the resulting ontology does not carry any relevant

Table 1. Example of two consistent ontologies resulting in an inconsistent ontology when merged, by adding the $\equiv$ statements to the union of the ontologies.

Knowledge Base $\mathcal{O}_1$	Knowledge Base $\mathcal{O}_2$	Equivalences
$Republican(mike)$	$Republican(robert)$	$Republican_{\mathcal{O}_1} \equiv Republican_{\mathcal{O}_2}$
$Republican(clara)$	$Democrat(sarah)$	$Democrat_{\mathcal{O}_1} \equiv Democrat_{\mathcal{O}_2}$
$likes(mike, clara)$	$likes(robert, sarah)$	$likes_{\mathcal{O}_1} \equiv likes_{\mathcal{O}_2}$
$likes(clara, mike)$	$likes(sarah, robert)$	
$Republican \sqcap Democrat \sqsubseteq \bot$		
$Republican \sqsubseteq \forall likes.Republican$		
$Democrat \sqsubseteq \forall likes.Democrat$		

information any more, since everything follows from inconsistency. One solution is to use the semantics of free defaults [12]; thereby replacing some classical subsumptions (e.g., $C \sqsubseteq D$) with the corresponding default subsumptions (e.g., $C \sqsubseteq_d D$). By applying this strategy to our example, we obtain a default knowledge base $\langle \Sigma, \mathcal{D} \rangle$, where the TBox of Σ is empty and the set $\mathcal{D}$ of defaults consists of

$$\begin{aligned} \mathit{Republican} \sqcap \mathit{Democrat} &\sqsubseteq_d \bot \\ \mathit{Republican} &\sqsubseteq_d \forall \mathit{likes}.\mathit{Republican} \\ \mathit{Democrat} &\sqsubseteq_d \forall \mathit{likes}.\mathit{Democrat}. \end{aligned}$$

While this results in a d-satisfiable default knowledge base, it still has some undesirable features. Following our example, in Table 2, we highlight three sets of axioms which are satisfied by one of the three types of d-models. Since there exist d-models for each of the axiom sets in Table 2, it follows that neither $\mathit{Republican}(sarah)$ nor $\neg\mathit{Republican}(sarah)$ is d-entailed by the default knowledge base. However, under the assumption that the satisfaction of $\mathit{Republican} \sqcap \mathit{Democrat} \sqsubseteq_d \bot$ is preferred over the satisfaction of the other two defaults, it is more desirable to consider only the third type of interpretations as d-models. Therefore, we extend the preferred models semantics in such a way that it takes an order on the defaults into account.

Table 2. Possible satisfied axiom combinations

d-model type 1	d-model type 2	d-model type 3
$\mathit{Republican}(sarah)$	$\mathit{Republican}(sarah)$	$\neg\mathit{Republican}(sarah)$
$\mathit{Democrat}(robert)$	$\neg\mathit{Democrat}(robert)$	$\neg\mathit{Democrat}(robert)$

3.2 Approach

We adopt the following definition which avoids the problem posed in the previous section.

Definition 1 (Preference relation over models). *Given a ordered default knowledge base $\langle \Sigma, \mathcal{D}, \succ \rangle$ where Σ is a knowledge base, $\mathcal{D}$ is a finite set of free defaults and $\succ$ is a strict partial order on $\mathcal{D}$. And given $\mathcal{I}$ and $\mathcal{J}$ are interpretations of Σ, $\mathcal{I}$ is preferred over $\mathcal{J}$ (w.r.t. $\succ$), denoted as $\mathcal{I} >_{\langle \Sigma, \mathcal{D}, \succ \rangle} \mathcal{J}$, iff*

1. *for all named individuals a appearing in Σ, $a^{\mathcal{I}} = a^{\mathcal{J}}$,*
2. *for all i such that $\mathcal{X}_i^{\mathcal{J}} \not\subseteq \mathcal{X}_i^{\mathcal{I}}$ there exists j, such that $\mathcal{X}_j^{\mathcal{I}} \not\subseteq \mathcal{X}_j^{\mathcal{J}}$ and $(C_j \sqsubseteq_d D_j) \succ (C_i \sqsubseteq_d D_i)$*
3. *$\mathcal{X}^{\mathcal{I}} \neq \mathcal{X}^{\mathcal{J}}$*

For simplicity purposes we will from now on abuse the notation and adopt the shorthand $>$ for $>_{\langle \Sigma, \mathcal{D}, \succ \rangle}$ as long as it is clear from the context. Observe that if $\mathcal{I} > \mathcal{J}$, then there always exists i s.t. $\mathcal{X}_i^{\mathcal{I}} \supsetneq \mathcal{X}_i^{\mathcal{J}}$, since $\succ$ is a strict order. The definitions of d-models, d-satisfiability and d-entailment are changed accordingly by replacing the order on interpretations $>_{\Sigma,\mathcal{D}}$ with the new order $>_{\Sigma,\mathcal{D},\succ}$. This is a generalisation since for $>_{\mathcal{D}} = \emptyset$ the semantics are equivalent i.e. $>_{\Sigma,\mathcal{D}} = >_{\Sigma,\mathcal{D},\succ}$.

Going back to the previous example, we assume that $\mathcal{D}$ is ordered by $\succ$, and

$$Republican \sqcap Democrat \sqsubseteq_d \bot \succ Republican \sqsubseteq_d \forall likes.Republican$$
$$Republican \sqcap Democrat \sqsubseteq_d \bot \succ Democrat \sqsubseteq_d \forall likes.Democrat.$$

For the ordered default knowledge base $\langle \Sigma, \mathcal{D}, \succ \rangle$, there exist strictly less d-models than for the unordered default knowledge base we previously considered. Namely, the d-models for the ordered knowledge base are exactly those of type 3 in Table 2. Therefore, in contrast to the unordered default knowledge base, the ordered default knowledge base entails both $\neg Republican(sarah)$ and $\neg Democrat(robert)$.

3.3 Decidability

We show that decidability of d-satisfiability and d-entailment are maintained with this generalisation. Let $\langle \Sigma, \mathcal{D}, \succ \rangle$ be an ordered default knowledge base in a decidable description logics $\mathcal{L}$, which can be extended with nominal concept expressions, while maintaining decidability. Let Ind_Σ be the set of named individuals occurring in Σ and profile $P = (P_1, \ldots, P_n)$, where $n = |\mathcal{D}|$ and for $1 \leq i \leq n$, $P_i \subseteq \text{Ind}_\Sigma$. We define Σ_P as the knowledge base obtained by adding the following axioms to Σ for each $C_i \sqsubseteq_d D_i \in \mathcal{D}$, as similarly defined in [12]:

1. $\bar{P}_i \sqsubseteq (D_i \sqcap \{a_1, \ldots, a_k\}) \sqcup (\neg C_i \sqcap \{a_1, \ldots, a_k\})$, where $\bar{P}_i$ is the nominal concept expression containing all the named individuals in P_i and $\{a_1, \ldots, a_k\}$ is the named concept expression containing all the named individuals in Ind_Σ.
2. $C_i \sqcap \neg\{a_1, \ldots, a_k\} \sqsubseteq D_i$, where $\{a_1, \ldots, a_k\}$ is the named concept expression containing all the named individuals in Ind_Σ.

With this definition in mind, we can show the decidability of d-satisfiability.

Theorem 1 (Decidability of d-satisfiability). *Let $\langle \Sigma, \mathcal{D}, \succ \rangle$ be an ordered default knowledge base, and $P = (\emptyset, \ldots, \emptyset)$. Then, $\langle \Sigma, \mathcal{D}, \succ \rangle$ is d-satisfiable iff Σ_P is classically satisfiable.*

Next, we show decidability of the generalized d-entailment. For that, we first introduce some notation to move on with an auxiliary lemma.

Let $\langle \Sigma, \mathcal{D}, \succ \rangle$ be an ordered default knowledge base, and $\mathcal{I}$ an interpretation. We define profile $P(\mathcal{I})$ by setting its every component $P_i(\mathcal{I}) = \{a \in \text{Ind}_\Sigma \mid a^\mathcal{I} \in \mathcal{X}_i^\mathcal{I}\}$.

Lemma 1. *Let $\langle \Sigma, \mathcal{D}, \succ \rangle$ be an ordered default knowledge base. Then,*

$$\{\mathcal{I} \mid \mathcal{I} \text{ } d\text{-model of } \langle \Sigma, \mathcal{D}, \succ \rangle\} = \{\mathcal{I} \mid \mathcal{I} \models \Sigma_{P(\mathcal{J})}, \mathcal{J} \text{ } d\text{-model of } \langle \Sigma, \mathcal{D}, \succ \rangle\}.$$

Here, we are interested in $\Sigma_{P(\mathcal{J})}$ in which $\mathcal{J}$ is a d-model. Intuitively, the knowledge base $\Sigma_{P(\mathcal{J})}$ represents all d-model $\mathcal{J}'$ such that $P(\mathcal{J}) = P(\mathcal{J}')$. Therefore, we call it d-model representing knowledge base (i.e., d-KB in short). The above lemma implies that one can check d-entailment by checking classical entailment in all d-KBs, $\Sigma_{P(\mathcal{J})}$. This is useful, since in general there can be more than one d-model $\mathcal{J}$ for one profile $P(\mathcal{J})$.

In the following we give a non-deterministic algorithm, which generates d-KBs and show its soundness and completeness.

Algorithm 1 d-model representing knowledge base generation

```
INPUT: An ordered default knowledge base $\langle \Sigma, \mathcal{D}, \succ \rangle$.
OUTPUT: A $d$-KB taking the order of the defaults into account.
$P \leftarrow (\emptyset, \ldots, \emptyset)$
$N \leftarrow (\emptyset, \ldots, \emptyset)$
$Defaults \leftarrow Array(C_1 \sqsubseteq_d D_1, \ldots, C_n \sqsubseteq_d D_n)$
while $\exists i : P_i \cup N_i \neq \mathrm{Ind}_\Sigma$ do
    $CurrentDefault \leftarrow C_i \sqsubseteq_d D_i$, such that $\forall j : C_j \sqsubseteq_d D_j \succ C_i \sqsubseteq_d D_i \Rightarrow P_j \cup N_j = \mathrm{Ind}_\Sigma$ and $P_i \cup N_i \neq \mathrm{Ind}_\Sigma$
    $CurrentIndividual \leftarrow a_k$ , such that $a \in \mathrm{Ind}_\Sigma$ and $a \notin P_i \cup N_i$
    $PNew \leftarrow P$
    $\mathrm{PNew}_i \leftarrow \mathrm{PNew}_i \cup \{a\}$
    if $\Sigma_{\mathrm{PNew}}$ is satisfiable then
        $P \leftarrow PNew$
    else
        $\mathrm{N}_i \leftarrow \mathrm{N}_i \cup \{a\}$
return $\Sigma_P$
```

The algorithm generates a knowledge base by non-deterministically adding one named individual to the individuals which satisfy one of the default subsumption, and checks whether there is a d-model with the new profile. For the default that is non-deterministically chosen, it must hold that all defaults which are preferable to it have been applied exhaustively. This algorithm uses $|\mathrm{Ind}_\Sigma| \cdot |\mathcal{D}|$ satisfiability checks for knowledge bases of the form Σ_P before the termination. With the two following theorems below, we show that our algorithm is sound and complete.

Theorem 2 (Soundness). *Let $\langle \Sigma, \mathcal{D}, \succ \rangle$ be a d-satisfiable ordered defaults knowledge base. Then, for every knowledge base Σ_P generated by Algorithm 1 there exists d-model $\mathcal{I}$ such that $P = P(\mathcal{I})$.*

Theorem 3 (Completeness). *Let $\langle \Sigma, \mathcal{D}, \succ \rangle$ be a d-satisfiable ordered defaults knowledge base. Then, for any d-model $\mathcal{I}$, there exist non-deterministic choices in Algorithm 1, such that the generated knowledge base is equal to $\Sigma_{P(\mathcal{I})}$.*

While the non-deterministic algorithm suffices for a decidability result, a naive implementation which tries all possibilities via breadth-first or depth-first search is highly inefficient. Therefore, we give an idea for a *smarter* algorithm which is (almost) deterministic, restricted to total orders. Its complete description is included in the appendix. In short, the algorithm iteratively tries to satisfy the currently most preferred default, for which there are named individuals that are not yet asserted to satisfy it, but can satisfy it. When adding the assertion that a named individual satisfies a default in the d-KB that is currently constructed, we check whether there exist named individuals, which cannot satisfy the current default any more, due to the addition of the assertion. In this case we know that there is another, different d-KB we have to consider. This approach leads to less non-deterministic choices being considered.

4 Discussion

4.1 Theoretical aspect

We consider the complexity of different tasks for both default knowledge bases with and without an order; namely for d-satisfiability and d-entailment.

***d*-satisfiability** The complexity of checking d-satisfiability does not depend on whether the default knowledge base is ordered or unordered, since there is a d-model for the ordered default knowledge base if and only if there is a d-model of the unordered default knowledge base, and one can be reduced to the other (and vice versa) in constant time. This follows from the proof of Theorem 1. This also means that we can check d-satisfiability by verifying the classic satisfiability of a knowledge base.

***d*-entailment** We consider the algorithm of generating all d-KBs for an ordered default knowledge base and checking d-entailment by checking classical entailment in all of them iteratively. It is easy to see that the time needed, after the knowledge bases have been generated, is linear in the number of generated knowledge bases. As an example, consider $\langle \mathit{\Sigma}_m, \mathcal{D}_n \rangle$, where

$$\begin{aligned} \mathit{\Sigma}_m &= \{C(a_1), \ldots, C(a_m)\}, \\ \mathcal{D}_n &= \{C \sqsubseteq_d D_1, \ldots, C \sqsubseteq_d D_n, \top \sqsubseteq_d \bigsqcup_{i=1}^{n} \neg D_i\}. \end{aligned}$$

One has to decide for each individual a_i, *which default* of the $n+1$ defaults is not satisfied. This is independent for each individual. Therefore, without an order, this default knowledge base has $(n+1)^m$ d-KBs. However, with any total strict order $\succ$ on these defaults the ordered default knowledge base $\langle \mathit{\Sigma}_m, \mathcal{D}_n, \succ \rangle$ will only have one d-KB. We name this type of inconsistency that occurs when one has to decide *which defaults* are satisfied for some individual, *horizontal inconsistency*. Observe that in case of horizontal inconsistencies, the addition of an order can be very useful, as it can reduce the amount of d-KBs by a factor which is exponential in the number of individuals. While this seems promising, we do not have such guarantee in general. Consider $\langle \mathit{\Sigma}'_m, \mathcal{D}'_n \rangle$, where

$$\begin{aligned} \mathit{\Sigma}'_m &= \{C(a_1), \ldots, C(a_{2m}), R(a_1, a_2), R(a_3, a_4), \ldots, R(a_{2m-1}, a_{2m})\}, \\ \mathcal{D}'_n &= \{C \sqsubseteq_d D_1 \sqcap \forall R.\neg D_1, \ldots, C \sqsubseteq_d D_n \sqcap \forall R.\neg D_n\}. \end{aligned}$$

Checking the above for two individuals a_{2i-1}, a_{2i} and one default $C \sqsubseteq_d D_j \sqcap \forall R.\neg D_j$, we have the choice whether the default should be satisfied either for a_{2i-1}, or a_{2i}. Since we have m pairs of individuals and n defaults, and each of the choices are independent, we have 2^{mn} d-KBs. We name this type of inconsistency that occurs when one has to decide for *which individuals* some default is satisfied, *vertical inconsistency*. In this example, the addition of any order on defaults does not change the number of d-KBs, since there are no horizontal inconsistencies.

Summarising, we see that in the worst case, checking d-entailment is exponential in the number of individuals times the number of rules. Further, it is independent of the presence of an order on the defaults. This, however, does not necessarily mean that the order is useless when it comes to *real-world* data.

4.2 Practical aspect

In order to test the performance of the algorithm on real-world data, we acquired inconsistent subsets of the LOD Cloud [7]. The subsets were acquired by choosing a random OWL subClassOf axiom C rdfs:subClassOf D contained in the LOD dataset, and by gathering all triples which relate C or D to other concepts, roles and lists. This was achieved by following relations from the rdf, rdfs and owl namespaces, which have C or D as object or subject. This process was repeated recursively for the related concepts, roles and lists until a maximum amount of axioms was found, or a recursion depth of three was reached. In order to avoid obtaining the same knowledge base multiple times, we employed two safety measures. First, the search for entities in the rdf, rdfs and owl namespace was disabled; that is, it was still possible to add an axiom of the form A rdfs:subClassOf owl:Thing. However, in this case, the LOD cloud was not queried for axioms which related owl:Thing to other entities. Also, an axiom of the form C rdfs:subClassOf D was only considered as a starting point for the recursive search, if neither C nor D had been initially considered. In addition to the structural information (i.e., TBox information), also a limited amount of class and property assertions was obtained during the search.

The ontologies obtained using this procedure were transformed to ordered default knowledge bases by, first, rewriting all disjointness and equivalence axioms for concepts to concept subsumptions. Afterwards, we removed all subsumption axioms, and used them as defaults. The choice of order was an arbitrary total order.

Using this approach we focused on three empirical questions:

1. How likely is it to find an inconsistent knowledge base, which has exactly one d-KB, when transformed into a default knowledge base with a total order?
2. How does the running time of our algorithm change with respect to the number of named individuals in the knowledge base?
3. How does the running time of our algorithm change with respect to the number of defaults in the knowledge base?

On the first question, we considered 100 inconsistent knowledge bases, each with approximately 100 named individuals, at most 100 axioms relating concepts, at most 100 property assertions and up to 50 properties. For each corresponding ordered default knowledge base, we checked d-satisfiability and whether the ordered default knowledge base had one d-KB or more.

As a result, we found that all of the inconsistent knowledge bases were d-satisfiable and 98 out of the 100 had only one d-KB. For the remaining two we tried to calculate the exact amount of d-KBs. This was, however, possible only for one of them which had two d-KBs. For the other one, the calculation was interrupted due to time constraints, and it had at least 20 d-KBs. This 98% is not necessarily representative for any real-world scenarios in a strict sense. However, it is still plausible that inconsistent ontologies, where the corresponding ordered default knowledge base has more than one d-KB, is *the exception rather than the rule*. This could change for larger datasets with more intricate relationships between the concepts; however, considering a reasonable amount of big datasets was infeasible for us.

This brings us to the next two questions about the scalability of the approach to larger datasets. We considered an inconsistent knowledge base, which referred to a total of 253 named individuals and 22 concept subsumptions.

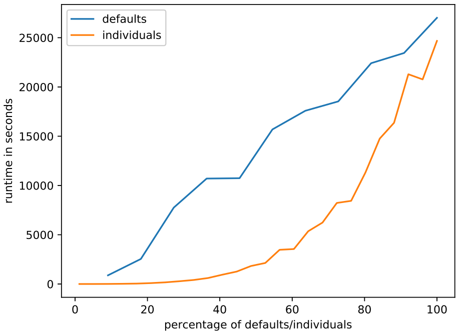

Fig. 1. The running time of the algorithm on the same ontology with different amounts of defaults or named individuals.

We ran the algorithm multiple times on this dataset with a different number of axioms referring to named individuals. This regulated the amount of named individuals appearing in the knowledge base. The orange line in Figure 1 shows the resulting runtime of the algorithm along with the percentage of named individuals occurring in the knowledge base. The number of consistency checks of a classic knowledge base increases linearly in the number of individuals, given that the knowledge base has exactly one d-KB. However, in Figure 1, we observe a convex behaviour in runtime in the number of named individuals. This is due to the fact that by adding more named individuals, the consistency checks also take longer. Yet, the runtime performance of our algorithm behaves almost linear in the number of individuals as it is illustrated in Figure 1. Here, the amount of named individuals remains unchanged, while the percentage of defaults included is increased. The reason that the graphs end in different points is due to different scheduling by the operating system.

5 Related Work

Many frameworks [12, 11, 2, 1] using defaults in the context of description logics have been considered since the introduction of defaults by Reiter [10]. Accordingly, there are various types of semantics that implement default reasoning, among them fixed-point [10, 3] as well as preferred model semantics [12, 8], which are closest to the classical semantics of description logics. Further, the idea of introducing a preference relation on defaults, or more generally, on axioms is not new, and has been at least considered for the fixed-point semantics [3] as well as the preferred model semantics [8]. We focus on the defeasible ontology language [8], since it is closest to our approach, and since it stands as a representative pointing out the differences of other approaches to ours. In their work Heymans et. al. introduced a preferred model semantics for description logics, based on a preference relation on the axioms of an ontology. Starting with the base language $\mathcal{SHOQ}(\mathbf{D})$, an ordered $\mathcal{SHOQ}(\mathbf{D})$-knowledge base is defined as a pair $\langle \mathcal{T}, \succ \rangle$, where $\mathcal{T}$ is a $\mathcal{SHOQ}(\mathbf{D})$ TBox and $\succ$ is a strict partial order on $\mathcal{T}$. We use $O - \mathcal{SHOQ}(\mathbf{D})$ as a shorthand for ordered $\mathcal{SHOQ}(\mathbf{D})$[4].

Intuitively, the order $\succ$ should implement a way of solving conflicts by deciding which axiom should be applied, taking the preference relation into consideration. For the semantics of $O - \mathcal{SHOQ}(\mathbf{D})$, classic $\mathcal{SHOQ}(\mathbf{D})$-interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ satisfying the Unique Naming and Common Domain Assumption are used. Such an interpretation is a *model* of $\langle \mathcal{T}, \succ \rangle$, if for each $A \sqsubseteq B \in \mathcal{T}$ and $x \in \Delta^{\mathcal{I}}$ it holds that, whenever $x \in A^{\mathcal{I}}$ then $x \in B^{\mathcal{I}}$, or there exists another more preferred subsumption $C \sqsubseteq D$, s. t. $x \in C^{\mathcal{I}} \cap D^{\mathcal{I}}$. Based on this definition of a model a preferred model semantics is introduced as follows: For a model $\mathcal{I}$ of a $O - \mathcal{SHOQ}(\mathbf{D})$-knowledge base $\langle \mathcal{T}, \succ \rangle$, its *support* is defined as

$$\mathcal{S}^{\mathcal{I}} = \{(x, A \sqsubseteq B) \mid x \in \Delta^{\mathcal{I}}, A \sqsubseteq B \in \mathcal{T}, \text{and } x \in \neg A \sqcup B\}.$$

A model $\mathcal{I}$ is then preferred over a model $\mathcal{J}$ for $\langle \mathcal{T}, \succ \rangle$ if, for all pairs $(x, A \sqsubseteq B) \in \mathcal{S}^{\mathcal{J}} \setminus \mathcal{S}^{\mathcal{I}}$ there exists a pair $(x, C \sqsubseteq D) \in \mathcal{S}^{\mathcal{I}} \setminus \mathcal{S}^{\mathcal{J}}$, s.t. $C \sqsubseteq D \succ A \sqsubseteq B$. This is denoted by $\mathcal{I} \geq \mathcal{J}$. If $\mathcal{S}^{\mathcal{I}} \neq \mathcal{S}^{\mathcal{J}}$, this preference holds strictly and is denoted by $\mathcal{I} > \mathcal{J}$. By considering only the most preferred models, i.e. the ones that are maximal w.r.t. $>$, one obtains a semantics which makes sure that for each individual, if there is a choice between making two rules classically satisfied for that individual, one always chooses the more preferred one.

While this semantics is very similar to ours, it is not ideal. Because it does not take care of cases where one has to decide between classically satisfying a less preferred rule for one individual or a more preferred one for another individual. This can be fixed simply by relaxing the constraint that defines the order on the models and replacing it as follows; $\mathcal{I} > \mathcal{J}$ iff for all pairs $(x, A \sqsubseteq B) \in \mathcal{S}^{\mathcal{J}} \setminus \mathcal{S}^{\mathcal{I}}$ there exists a pair $(y, C \sqsubseteq D) \in \mathcal{S}^{\mathcal{I}} \setminus \mathcal{S}^{\mathcal{J}}$, s.t. $C \sqsubseteq D \succ A \sqsubseteq B$.

Note that the replacement of x with y in the existential quantification is the only difference. With this semantics, when one has to decide between satisfying different concept subsumptions for different individuals, one shall choose the one where the more preferred concept subsumption is fulfilled.

This preference relation is very similar to ours. The main difference is that in our proposal only named individuals can be exceptions. Indeed, if we restrict the definition of the support to the named individuals, then the preference relations are equivalent.

Heymans et. al. showed that under the Common Domain and Unique Name Assumption [1] entailment in the original semantics is decidable, using a tableau algorithm. We suspect this result carries over to the new semantics. While this would be a

[4] The original shorthand is $\mathcal{OSHOQ}(\mathbf{D})$, however we chose not to use it since in the description logics naming convention every letter is assigned for one feature.

positive theoretical result, we expect that the worst case complexity of checking entailment is exponential in the number of unblockable individuals [8], assuming linearity in the amount of possible tableau's. This is shown in Example 4.1, which can be extended to also work for unnamed individuals. With a possibly exponential amount of unblockable individuals, this leaves us with a worst case complexity, which scales double exponentially. Adding the restriction to have exceptions of defaults only for named individuals, as Sengupta et. al. [12] and we did, reduces the worst case complexity, so that it only scales exponentially in the size of named individuals.

Another framework introduced by Baader and Hollunder also gives the possibility to specify a preference order on defaults. Their approach uses fixed points semantics and gives similar results to those of Heyman et. al., in the sense that a default can only be defeated for a certain individual by asserting a more preferred default for the same individual [3]. Further, defaults are only applied to named individuals in their work.

Two other semantics, which also concern themselves with orders on defaults are those of Bonatti et. al.'s $\mathcal{DL}^N$ semantics [5] and Lehmann's lexicographical closure [9]. In the former, the closed world assumption is somewhat weaker than ours. This results in the inability to derive concept membership for individuals, when it depends on the application of defaults. This stronger closed world assumption however also has its problems: Concepts are reasoned to be unsatisfiable, with the usage of defaults, even though the addition of more axioms can render them satisfiable. The major difference about the latter work [9] is that it obtains a lexicographic preference from a given knowledge base that respects the specificity of the antecedents in case of conflict.

6 Conclusion and Future Work

We have shown that it is possible to extend the semantics of free defaults with a preference order on the axioms, without losing decidability of both satisfiability and entailment. Our semantics improves both the semantics of Sengupta et. al.[12] as well as Heymans et. al.[8] by reducing the amount of d-KBs in the presence of horizontal inconsistencies. For a given knowledge base, this can both lead to a higher efficiency of entailment checks, and an increase in the amount of derivable knowledge. It turns out that, while the worst case runtime of generating all d-KBs (as well as their worst case amount), is exponential in the number of named individuals, working with the semantics we proposed in practice seems to be feasible. This is at least suggested by the empirical evidence, where we tested our approach on inconsistent real-world data i.e., subsets of the LOD-Laundromat [7].

It remains questionable whether these results generalise to larger and possibly more complex real-world examples. Another open problem on the foundational frontier is to find ways of handling ordered default knowledge bases that are not d-satisfiable, or that have more than one d-KB.

As a part of future work, it could be interesting to investigate which existing desirable properties our approach satisfies. Moreover, we need to carry out a fine-grained complexity theoretic analysis, supplemented with an empirical evaluation. This research agenda could make preferential default reasoning practically feasible on sizeable ontologies.

References

1. Baader, F., Calvanese, D., McGuinness, D., Patel-Schneider, P., Nardi, D.: The description logic handbook: Theory, implementation and applications. Cambridge university press (2003)
2. Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. Journal of Automated Reasoning **14**(1), 149–180 (1995)
3. Baader, F., Hollunder, B.: Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. Journal of Automated Reasoning **15**(1), 41–68 (1995)
4. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia-a crystallization point for the web of data. Web Semantics: science, services and agents on the world wide web **7**(3), 154–165 (2009)
5. Bonatti, P.A., Sauro, L.: On the logical properties of the nonmonotonic description logic dln. Artificial Intelligence **248**, 85–111 (2017)
6. Euzenat, J., Shvaiko, P., et al.: Ontology matching, vol. 18. Springer (2007)
7. Fernández, J.D., Beek, W., Martínez-Prieto, M.A., Arias, M.: Lod-a-lot. In: International Semantic Web Conference. pp. 75–83. Springer (2017)
8. Heymans, S., Vermeir, D.: A defeasible ontology language. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". pp. 1033–1046. Springer (2002)
9. Lehmann, D.: Another perspective on default reasoning. Annals of Mathematics and Artificial Intelligence **15**(1), 61–82 (1995)
10. Reiter, R.: A logic for default reasoning. Artificial intelligence **13**(1-2), 81–132 (1980)
11. Scharrenbach, T., Grütter, R., Waldvogel, B., Bernstein, A.: Structure preserving tbox repair using defaults. In: 23rd International Workshop on Description Logics DL2010. p. 384 (2010)
12. Sengupta, K., Hitzler, P., Janowicz, K.: Revisiting default description logics–and their role in aligning ontologies. In: Joint International Semantic Technology Conference. pp. 3–18. Springer (2014)