Approximate Subsumption in $\mathcal{ALCQ}$

Heiner Stuckenschmidt

University of Mannheim, Germany
A5, 6 68159 Mannheim
heiner@informatik.uni-mannheim.de

Abstract. We present a non-standard interpretation for concept expressions in $\mathcal{ALCQ}$ that defines approximate notions of subsumption based on approximating a subset of the concept and role names. We present the non-standard semantics, and the corresponding notion of approximate subsumption, discuss its formal properties and show that is can be computed by syntactic manipulations of concept expressions.

1 Introduction

Description Logics are becoming more and more popular as a formalism for representing and reasoning about conceptual knowledge in different areas such as databases and semantic web technologies. In particular, subsumption reasoning for expressive ontologies has been used to compute matches between conceptual descriptions in the context of different real world tasks including information integration, product and service matching and data retrieval. In practical situations, however, it often turns out that logical reasoning is inadequate in many cases, because it does not leave any room for *partial matches*.

Recently, there are some efforts that try to address this problem by combining description logics with numerical techniques for uncertain reasoning in OWL, in particular with techniques for probabilistic [1] and fuzzy reasoning [2]. These approaches are able to compute partial matches by assigning an assessment of the degree of matching to the subsumption relation. This degree of matching normally is a real number or an interval between zero and one and therefore allows some ordering of the solutions. Although, in principle this is a solution to the problem of computing the best partial match but defining an interpreting numerical assessments of uncertainty is a difficult problem. Further, the reduction to a single numerical assessment of the mismatch does not allow different users to discriminate between different kinds of mismatches.

In this paper, we propose a notion of approximate subsumption that supports the computation of partial matches between complex concept expressions without relying on a single number to represent the degree of mismatch. Instead, *the approach describes the degree of matching in terms of a subset of the aspects of the request that are met by the solution*. This approach allows the user to decide whether to accept a partial match based on whether important aspects are missed or not. In order to implement this approach we borrow from the area of approximate deduction. In particular, we extend the notion of S-Interpretations of propositional logic proposed in [3] to description logics and use the result notion of a non-standard interpretation of concept expressions to define an approximate subsumption operator that computes subsumption with respect to a particular subset of the vocabulary used.

2 Approximation based on Sub-Vocabularies

In propositional logic, the vocabulary of a formula consists of a set of propositional letters. A formula consists of a Boolean expression over these letters. A classical interpretation I assigns to each letter either the value *true* or *false*. The semantics of negation now implies that a letter and its negation cannot have the same truth value, in particular, for all propositional letters p one of the following :

$$\begin{aligned} I(p \wedge \neg p) &= false \\ I(p \vee \neg p) &= true \end{aligned} \tag{1}$$

Checking satisfiability of a formula relies on showing that there is no assignment of truth values that satisfies this condition and makes the whole formula true. A possible way for approximating satisfiability testing for propositional logic is now to restrict the condition above to a subset of the propositional letters. This subset is denoted as S and the corresponding interpretation is called an S-interpretation of the formula [3].

Depending on how the letters not in S are treated, an S-Interpretation is sound or complete with respect to the classical interpretation. One kind of non-standard interpretation called S-3 Interpretation assigns both, a letter and its negation to *true*.

$$I(p \wedge \neg p) = true, p \notin S \tag{2}$$

When applying this interpretation to the satisfiability problem, we observe that formulas that were unsatisfiable before now become satisfiable. This means that the resulting calculus is sound, but incomplete, because some results that could be proven using the principle of proof by refutation can not be proven any more, because the conjunction of the knowledge base with the negation of the result to be proven becomes satisfiable under the new interpretation. The counterpart of S-3 interpretation are S-1 Interpretations that assign *false* to both a letters and their negation if the letters are not in the set S.

$$I(p \vee \neg p) = false, p \notin S \tag{3}$$

Following the same argument as above, S-1 Interpretations define a complete but unsound calculus for propositional logic. In both cases, the advantage of the approach is that we can decide which parts of the problem to approximate by selecting an appropriate set of letters S. Therefore the approach provides a potential solution to the problem of partial matching described above.

The idea of our approach is now to apply the underlying idea of S-Interpretations to the Description Logic $\mathcal{ALCQ}$ which covers most of the expressive power of OWL in order to support approximate subsumption reasoning where parts of the vocabulary are interpreted in the classical way and other parts are approximated. In fact, Cadoli and Schaerf do propose an extension of S-Interpretations to Description logics, but they define S not in terms of a subset of the vocabulary, but in terms of the structure of the concept expression [4]. In [5] it has been shown that this way of applying S-Interpretations to description logics does not produce satisfying results on real data. In this paper, we therefore propose an alternative way of defining S-Interpretations for description logics which is closer to the notion of S-Interpretations in propositional logic. The idea is to interpret description logics as an extension of propositional logic, where class names correspond to propositional letters[1]. As for propositional logic, we select a subset of the class names that is interpreted in the classical way and approximate class names not in this set. In particular, a classical interpretation $(\Delta^{\mathcal{I}}, \mathcal{I})$ of class names requires that a concept name and its negation form a disjoint partition of the domain:

$$\begin{aligned} C^{\mathcal{I}} \cap (\neg C)^{\mathcal{I}} &= \emptyset \\ C^{\mathcal{I}} \cup (\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \end{aligned} \tag{4}$$

We can now define approximations for description logics by relaxing these requirements for a subset of the concept names. The corresponding S-3 and S-1 Interpretations are very similar to the ones for propositional logic. In particular, for S-3 Interpretations we have.

$$C^{\mathcal{I}} \cap (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}}, C \notin S \tag{5}$$

This means that both, C and $\neg C$ are mapped to $\Delta^{\mathcal{I}}$ by the interpretation. As a consequence, the concept name C cannot cause a clash in a tableaux proof and therefore, constraints that force a certain value to be of type C will be ignored in a subsumption proof. The resulting subsumption operator is sound, but incomplete. For S-1 Interpretations, we have

$$C^{\mathcal{I}} \cup (\neg C)^{\mathcal{I}} = \emptyset, C \notin S \tag{6}$$

which means that both C and $\neg C$ are mapped to the empty set. In a tableaux proof, all attempts to construct a model that involves a variable of type C will fail. The corresponding subsumption operator is complete, but unsound with respect to classical subsumption.

While approximation based on concept names is a straightforward application of the notion of S-1 and S-3 interpretations, things become more complicated if we want to extend the approach to relation names. In Description Logics relations are used to formulate constraints that apply to all members of a certain class. The most general formulation of these constraints is in terms of qualified number restrictions. Qualified number restrictions have the following form $(\leq\ n r.C)$ or $(\geq\ n r.C)$ where n is a positive natural number (including zero), r is the name of a binary relation and C is a concept expression. In a tableaux these qualified number restrictions are a second potential source of inconsistency besides the negation operator. In particular, we have

$$(\leq n\, r.C)^{\mathcal{I}} \cap (\geq m\, r.C)^{\mathcal{I}} = \emptyset \text{ for all } n < m$$

on the other hand, we have

$$(\leq nr.C)^{\mathcal{I}} \cup (\geq mr.C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \text{ for all } n \geq m$$

We can use this analogy to extend the notion of S-1 and S-3 interpretations to qualified number restrictions in the following way. For S-3 Interpretations we define that

$$(\leq n\, r.C)^{\mathcal{I}} \cup (\geq m\, r.C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \text{ for all } r \notin S \tag{7}$$

In particular, we weaken the condition for the expression to become the universal concept by making it independent of the values for m and n. Further, we claim that the conjunction of qualified number expressions can never be the empty concept, i.e.

$$(\leq n\, r.C)^{\mathcal{I}} \cap (\geq m\, r.C)^{\mathcal{I}} \neq \emptyset \text{ for all } r \notin S \tag{8}$$

[1] In fact, a description logic that just contains the Boolean operators is equivalent to propositional logic.

This leaves us with a weaker interpretation, because inconsistencies arising from the relations not in the set S cannot be detected. For S-1 interpretations, we make analogous claim by demanding that the union of two qualified number restrictions can never be the universal concept

$$(\leq n\, r.C)^{\mathcal{I}} \cup (\geq m\, r.C)^{\mathcal{I}} \neq \Delta^{\mathcal{I}} \text{ for all } r \notin S \tag{9}$$

Further, we strengthen the interpretation by claiming that the intersection of the two qualified number restrictions on the same relation and concept is always inconsistent

$$(\leq n\, r.C)^{\mathcal{I}} \cap (\geq m\, r.C)^{\mathcal{I}} = \emptyset \text{ for all } r \notin S \tag{10}$$

This gives us a stronger version of the semantics, because any two assertions using this relation in combination with the same concept expression C leads to an inconsistency[2]. The result is a complete but unsound subsumption operator. This unsound approximation operator is exactly what we need for specifying the notion of a partial match, because it forces a match on the constraints involving class names from S and treats constraints involving classes not in S as optional. Using subsumption operators with different sets S, we can focus on different aspects of the matching task and also rank results based on the number of requirements met. In the following, we will therefore concentrate on complete, but unsound approximations of subsumption reasoning for concept expressions based on the idea described above. In particular, we will formally specify non-standard interpretations and define a family of approximate subsumption operators that can be used to compute partial matches.

3 Non-Standard Semantics

In the following, we introduce a non-standard interpretation for concept expressions in the logic $\mathcal{ALCQ}$. A limited vocabulary is a subset $S \subseteq \mathcal{V}$ of the concept and relation names occurring in a concept expression. Our aim is to define approximate reasoning in Description Logics based on such a subset of the vocabulary. For this purpose, we define an upper and a lower approximation of an interpretation $\mathcal{I}$ with respect to a set S referred to as $\mathcal{I}_S^+$ and $\mathcal{I}_S^-$ respectively. We call $\mathcal{I}_S^+$ an upper approximation and $\mathcal{I}_S^-$ a lower approximation of $\mathcal{I}$ with respect to S.

Definition 1 (Lower Approximation). *A lower approximation of an interpretation $\mathcal{I}$ with respect to S is a non standard interpretation $(\Delta^{\mathcal{I}}, \mathcal{I}_S^-)$ such that:*

$$A^{\mathcal{I}_S^-} = \begin{cases} A^{\mathcal{I}} & A \in S \\ \emptyset & otherwise \end{cases} \tag{11}$$

$$(\neg C)^{\mathcal{I}_S^-} = \Delta^{\mathcal{I}} - C^{\mathcal{I}_S^+} \tag{12}$$

$$(C \sqcap D)^{\mathcal{I}_S^-} = C^{\mathcal{I}_S^-} \cap D^{\mathcal{I}_S^-} \tag{13}$$

$$(C \sqcup D)^{\mathcal{I}_S^-} = C^{\mathcal{I}_S^-} \cup D^{\mathcal{I}_S^-} \tag{14}$$

$$(\geq n\, r.C)^{\mathcal{I}_S^-} = \begin{cases} \{x \mid \#\{y.(x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}_S^-}\} \geq n\} & r \in S \\ \{x \mid \#\{y.(x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}_S^-}\} \geq \infty\} & otherwise \end{cases} \tag{15}$$

$$(\leq n\, r.C)^{\mathcal{I}_S^-} = \begin{cases} \{x \mid \#\{y \mid (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}_S^+}\} \leq n\} & r \in S \\ \{x \mid \#\{y \mid (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}_S^+}\} \leq 0\} & otherwise \end{cases} \tag{16}$$

---

[2] As we will see later, it is sufficient if the two restrictions use concept expressions that are logically equivalent

where $(\Delta^{\mathcal{I}}, \mathcal{I}_S^+)$ *is an upper approximation as defined in definition 2*

Definition 2 (Upper Approximation). *An upper approximation of an interpretation* $\mathcal{I}$ *with respect to* S *is a non standard interpretation* $(\Delta^{\mathcal{I}}, \mathcal{I}_S^+)$ *such that:*

$$A^{\mathcal{I}_S^+} = \begin{cases} A^{\mathcal{I}} & A \in S \\ \Delta^{\mathcal{I}} & otherwise \end{cases} \tag{17}$$

$$(\neg C)^{\mathcal{I}_S^+} = \Delta^{\mathcal{I}} - C^{\mathcal{I}_S^-} \tag{18}$$

$$(C \sqcap D)^{\mathcal{I}_S^+} = C^{\mathcal{I}_S^+} \cap D^{\mathcal{I}_S^+} \tag{19}$$

$$(C \sqcup D)^{\mathcal{I}_S^+} = C^{\mathcal{I}_S^+} \cup D^{\mathcal{I}_S^+} \tag{20}$$

$$(\geq n\, r.C)^{\mathcal{I}_S^+} = \begin{cases} \{x | \#\{y.(x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}_S^+}\} \geq n\} & r \in S \\ \{x | \#\{y.(x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}_S^+}\} > 0\} & otherwise \end{cases} \tag{21}$$

$$(\leq n\, r.C)^{\mathcal{I}_S^+} = \begin{cases} \{x | \#\{y | (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}_S^-}\} \leq n\} & r \in S \\ \{x | \#\{y | (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}_S^-}\} < \infty\} & otherwise \end{cases} \tag{22}$$

where $(\Delta^{\mathcal{I}}, \mathcal{I}_S^-)$ *is a lower approximation as defined in definition 1*

A nice property of this definition is that it ensures the existence of a negation normal form that can be computed using the same transformation rules as usually.

Corollary 1 (Negation Normal Form). *For every concept expression* C *there is an expression* $nnf(C)$ *in negation normal form such that* $nnf(C)^{\mathcal{I}_S^-} = C^{\mathcal{I}_S^-}$ *and* $nnf(C)^{\mathcal{I}_S^+} = C^{\mathcal{I}_S^+}$

Another useful property of the non standard interpretation is that it makes concept expressions strictly more general for upper and strictly more specific for lower approximations. This property which we call monotonicity is important in order to be able to guarantee formal properties of approximation methods defined based on this interpretation. Therefore the following theorem describes a central property of approximation in description logics.

Lemma 1 (Monotonicity). *Given a non-standard interpretation as defined above, the following equation holds for all concept expressions* C*:*

$$C^{\mathcal{I}_S^-} \subseteq C^{\mathcal{I}} \subseteq C^{\mathcal{I}_S^+} \tag{23}$$

We can generalize the theorem by observing that the standard interpretation is an extreme case of the non-standard interpretation with $S = \mathcal{V}$. In particular, the general version of monotonicity says that for upper approximations removing names from the set S will make concepts expressions strictly more general. Conversely, for lower approximations concept expressions become less general when we remove concept or relation names from the set S. The corresponding general property is defined in the following theorem:

Lemma 2 (Generalized Monotonicity). *Given a non-standard interpretation as defined above and two sub-vocabularies* S_1 *and* S_2 *with* $S_1 \subseteq S_2$*, the following equations hold for all concept expressions* C*:*

$$C^{\mathcal{I}_{S_1}^-} \supseteq C^{\mathcal{I}_{S_2}^-} \quad C^{\mathcal{I}_{S_1}^+} \subseteq C^{\mathcal{I}_{S_2}^+} \tag{24}$$

The generalized monotonicity property is interesting, because it allows us to successively compute more precise upper and lower approximations of a concept by adding names to the set S. This is convenient in cases where users provide a preference order over the vocabulary indicating the relative importance of different aspects of a concept. In this case, use the preference relation provided by the user to determine a sequence of approximations to be used in the matching process.

4 An Approximate Subsumption Operator

Up to now, we have only considered interpretations as such. As our aim is to develop approximate notions of subsumption as a basis for approximate matching, we now have to define the notion of approximate subsumption based on the non-standard interpretation defined above. It turns out, that this can be done in a straightforward way using the standard definition of the subsumption operator as:

$$\forall \mathcal{I} : \mathcal{I} \models C \sqsubseteq D \Leftrightarrow (C \sqcap \neg D)^{\mathcal{I}} = \emptyset$$

The idea is now to use this definition and replace the standard interpretation $\mathcal{I}$ by a the lower approximation $\mathcal{I}_S^-$ with respect to a certain sub-vocabulary S. Based on the choice of S, this defines different subsumption operators with certain formal properties that will be discussed in the following.

Definition 3 (Approximate Subsumption). *Let $S \subseteq \mathcal{V}$ be a subset of the concept names and $(\Delta^{\mathcal{I}}, \mathcal{I}_S^-)$ a lower approximation, then the corresponding approximate subsumption relation $\underset{S}{\sqsubseteq}$ is defined as follows*

$$\forall \mathcal{I} : \mathcal{I} \models (C \underset{S}{\sqsubseteq} D) \ \Leftrightarrow_{def} \ (C \sqcap \neg D)^{\mathcal{I}_S^-} = \emptyset \tag{25}$$

We say that C is subsumed by D with respect to sub-vocabulary S.

The monotonicity of the non-standard interpretation has an impact on the formal properties of the approximate subsumption operator. In particular, we can establish a relation between the subset of the vocabulary considered and the strength of the subsumption operator. The more concepts we exclude from the set S the weaker the subsumption operator as well as the matches we can compute get. This implies that if we can prove subsumption with respect to a particular set S the subsumption relation also holds for all subsets of S. Conversely, if we fail to prove subsumption with respect to a set S, we can be sure that the subsumption relation does also not hold with respect to any superset of S. These properties are stated formally in the following theorem.

Theorem 1 (Properties of Approximate Subsumption). *Let f be a lower approximation, then the following equation holds:*

$$\left(C \underset{S_2}{\sqsubseteq} D\right) \Rightarrow \left(C \underset{S_1}{\sqsubseteq} D\right) \ \textit{for}\ S_1 \subseteq S_2 \tag{26}$$

$$\left(C \underset{S_1}{\not\sqsubseteq} D\right) \Rightarrow \left(C \underset{S_2}{\not\sqsubseteq} D\right) \ \textit{for}\ S_1 \subseteq S_2 \tag{27}$$

These properties allow us to develop approximation strategies by successively selecting smaller subsets of concepts to be considered for matching and trying to compute the corresponding subsumption relation until we succeed.

5 Computing Approximate Subsumption

A nice feature of our approach is that it can actually be implemented by simply performing syntactic modifications on concept expressions. In particular, in order to check whether a statement $C \underset{S}{\sqsubseteq} D$ holds, we take the expression $(C \sqcap \neg D)$ and transform it into a concept expression that simulates the non-standard interpretation. For the lower approximation, the corresponding transformation $(.)^-$ is defined as follows

$$
\begin{array}{c}
(A)^- \rightarrow \bot \text{ if } A \in S \\
(\neg A)^- \rightarrow \bot \text{ if } A \in S \\
(\neg C)^- \rightarrow \neg(C)^+ \\
(C \sqcap D)^- \rightarrow (C)^- \sqcap (C)^- \\
(C \sqcup D)^- \rightarrow (C)^- \sqcup (C)^- \\
(\leq n\, r.C)^- \rightarrow (\leq 0\, r.(C)^+) \text{ if } r \in S \\
(\leq n\, r.C)^- \rightarrow (\leq n\, r.(C)^+) \text{ if } r \not\in S \\
(\geq n\, r.C)^- \rightarrow (\geq max\, r.(C)^-) \text{ if } r \in S \\
(\geq n\, r.C)^- \rightarrow (\geq n\, r.(C)^-) \text{ if } r \not\in S
\end{array}
$$

Here max is an integer number that is larger than any other number occurring in any qualified number restriction in the concept expression. This is sufficient to model the interpretation that requires less than an infinite number of r-successors. Analogously, we define a transformation function $(.)^+$ that creates a concept expression that simulates the upper approximation of a concept expression. This transformation is defined as follows:

$$
\begin{array}{c}
(A)^+ \rightarrow \top \text{ if } A \in S \\
(\neg A)^+ \rightarrow \top \text{ if } A \in S \\
(\neg C)^+ \rightarrow \neg(C)^- \\
(C \sqcap D)^+ \rightarrow (C)^+ \sqcap (C)^+ \\
(C \sqcup D)^+ \rightarrow (C)^+ \sqcup (C)^+ \\
(\leq n\, r.C)^+ \rightarrow (\leq max - 1\, r.(C)^-) \text{ if } r \in S \\
(\leq n\, r.C)^+ \rightarrow (\leq n\, r.(C)^-) \text{ if } r \not\in S \\
(\geq n\, r.C)^+ \rightarrow (\geq 1\, r.(C)^+) \text{ if } r \in S \\
(\geq n\, r.C)^+ \rightarrow (\geq n\, r.(C)^+) \text{ if } r \not\in S
\end{array}
$$

We again use the number max for modeling an infinite number of r-successors. Further, we have to use the condition ≥ 1 instead of < 0 which is equivalent. It can be shown that these rewriting rules provide a way for computing approximate subsumption as stated by the following theorem.

Theorem 2 (Syntactic approximation I). *Let C and D be concept expressions in $\mathcal{ALCQ}$, then $\mathcal{I} \models C \underset{S}{\sqsubseteq} D$ if and only if $(C \sqcap \neg D)^-$ is unsatisfiable.*

It turns out that the equivalence of a concept expression and its normal form and the symmetry of upper and lower approximation with respect to negation can be used to define an alternative way of computing approximate subsumption based on the syntactic manipulations shown above

Theorem 3 (Syntactic approximation II). *Let C and D be concept expressions in $\mathcal{ALCQ}$, then $\mathcal{I} \models C \underset{S}{\sqsubseteq} D$ if and only if $\mathcal{I} \models (C)^- \sqsubseteq (D)^+$*

This means that we have two rather straightforward ways of computing approximate subsumption using standard DL reasoners.

6 Discussion

We presented an approach for computing approximate subsumption between concept expressions in $\mathcal{ALCQ}$ based on a subset of the vocabulary used in the expressions. The approach solves some of the problems of classical reasoning in description logics, in particular, the inability to accept imperfect matches between concepts without having to leave the realms of formal logic. As a side-effect, the subset of the vocabulary also provides us with a qualitative characterization of the mismatch between the expressions, which is clearly an advantage over numerical approaches for dealing with imperfect matches. An approach for partial matching in description logics that is more similar to ours is reported in [6]. This approach, however, cannot deal with disjunction and qualified number restrictions.

References

1. Giugno, R., Lukasiewicz, T.: P-shoq(d): A probabilistic extension of shoq(d) for probabilistic ontologies in the semantic web. In: Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA'02). (2002)
2. Straccia, U.: Towards a fuzzy description logic for the semantic web (preliminary report). In: Proceedings of the 2nd European Semantic Web Conference ESWC-05. (2005) 167–181
3. Schaerf, M., Cadoli, M.: Tractable reasoning via approximation. Artificial Intelligence **74**(2) (1995) 249–310
4. Cadoli, M., Schaerf, M.: Approximation in concept description languages. In: Proceedings of the International Conference on Knowledge Representation and Reasoning. (1992) 330–341
5. Groot, P., Stuckenschmidt, H., Wache, H.: Approximating description logic classification for semantic web reasoning. In: Proceedings of the 2nd European Semantic Web Conference, Heraklion, Crete (2005)
6. Di Noia, T., Eugenio Di Sciascio, F.D., Mongiello, M.: A system for principled matchmaking in an electronic marketplace. In: Proceedings of the Twelfth International World Wide Web Conference. (2003) 321–330