Complete Classification of Complex $\mathcal{ALCHO}$ Ontologies using a Hybrid Reasoning Approach

Weihong Song, Bruce Spencer, and Weichang Du

Faculty of Computer Science, University of New Brunswick, Fredericton, Canada
{song.weihong, bspencer, wdu}@unb.ca

Abstract. Consequence-based reasoners are typically significantly faster than tableau-based reasoners for ontology classification. However, for more expressive DL languages like $\mathcal{ALCHO}$, consequence-based reasoners are not applicable, but tableau-based reasoners can sometimes require an unacceptably long time for large and complex ontologies. This paper presents a weakening and strengthening approach for classification of $\mathcal{ALCHO}$ ontologies, using a hybrid of consequence- and tableau-based reasoning. We approximate the original ontology O_o by a weakened version O_w and a strengthened version O_s, both are in a less expressive DL $\mathcal{ALCH}$ and classified by a consequence-based main reasoner. The classification from O_w is sound but possibly incomplete with respect to O_o, while that from O_s is complete but possibly unsound. The additional subsumptions derived from O_s may be unsound so are further verified by a tableau-based assistant reasoner. A prototype classifier called WSClassifier is implemented based on this hybrid approach. The experiments results show that for classifying many large and complex $\mathcal{ALCHO}$ ontologies, WSClassifier's performance is significantly faster than tableau-based reasoners.

1 Introduction

The target of classification is to calculate all the subsumption relationships between atomic concepts implied by the input ontology. (Hyper)Tableau-based [9,13] and consequence-based reasoners are two of the mainstream reasoners for ontology classification. Current (hyper)tableau-based reasoners such as HermiT [13], FaCT++ [20], Pellet [18] and RacerPro [8], are able to classify ontologies in very expressive DLs. However, despite various optimizations having been applied, classifying certain existing large and complex ontologies is still a challenge for these reasoners, such as various versions of Galen and FMA ontologies. We regard an ontology complex if it is highly cyclic. In contrast to the (hyper)tableau-based reasoners, consequence-based reasoners [10,11,17] are typically very fast but support less expressive DLs. They are variations of so-called completion-based approaches proposed for the OWL EL family [3,5]. So far the most expressive languages that are supported by consequence-based reasoners are Horn-$\mathcal{SHIQ}$ [10] and $\mathcal{ALCH}$ [17].

In this paper we introduce a hybrid reasoning approach for classification of ontologies in the DL $\mathcal{ALCHO}$, using a consequence-based *main reasoner* MR and a tableau-based *assistant reasoner* AR. MR provides sound and complete classification over the DL $\mathcal{ALCH}$ which is less expressive than $\mathcal{ALCHO}$, while AR provides sound and complete classification over $\mathcal{ALCHO}$. Suppose MR reasoning is much faster than AR. We try to classify an ontology $\mathcal{O}_o$ using MR to do the major work, and AR to do auxiliary work. We produce a weakened version $\mathcal{O}_w$ by removing from $\mathcal{O}_o$ the nominal axioms that are beyond $\mathcal{ALCH}$, and a strengthened version $\mathcal{O}_s$ by adding to $\mathcal{O}_w$ a set of strengthening axioms $\mathcal{O}^+$ in $\mathcal{ALCH}$ that compensate for the removed axioms. $\mathcal{O}_s$ and $\mathcal{O}_w$ are in $\mathcal{ALCH}$ and are classified by MR producing $\mathcal{H}_w$ and $\mathcal{H}_s$, correspondingly. $\mathcal{H}_w$, $\mathcal{H}_o$ and $\mathcal{H}_s$ are classification results of $\mathcal{O}_w$, $\mathcal{O}_o$ and $\mathcal{O}_s$, respectively. Subsumptions in $\mathcal{H}_w$ are sound but may not be complete w.r.t $\mathcal{O}_o$, whereas subsumptions in $\mathcal{H}_s$ are complete but may not be sound. Unsound subsumptions in $\mathcal{H}_s \setminus \mathcal{H}_w$ are detected by AR and filtered out. Those that remain are added to $\mathcal{H}_w$ resulting in the sound and complete classification of $\mathcal{O}_o$. We call this approach weakening and strengthening (WS).

We have implemented a prototype WSClassifier by applying the proposed WS approach and evaluated it. Our previous study on WS approach and application to $\mathcal{ALCHOI}$ ontologies [19] did not guarantee the completeness of classification. The contribution of this paper is to improve the algorithms and theoretically prove our classification result on $\mathcal{ALCHO}$ ontologies is sound(trivial) and complete.

The rest of the paper is organized as follows: Section 2 gives an overview of our hybrid classification procedure for $\mathcal{ALCHO}$ ontologies and prove its completeness. Section 3 introduces computation of strengthening axioms and Section 4 contains the related work. Section 5 is our partial empirical results and conclusion. Finally, we put all the proofs of lemmas and theorems in Appendix B and complete evaluation in Appendix C.

2 Hybrid Classification of Ontologies

The syntax and semantics of $\mathcal{ALCHO}$ follows DL conventions. In this paper, we use A, B, E, F for atomic concepts, C, D for concepts, R, S for roles, a, b for individuals, H, K for conjunctions of atomic concepts, and M, N for disjunctions.

Algorithm 1 describes our hybrid procedure for classifying $\mathcal{O}_o$. It consists of three stages: (1) a *normalization stage* (line 1) during which the ontology is rewritten to simplify the forms of axioms in it; (2) a *main classification stage* (lines 2 to 8) in which $\mathcal{O}_w$ and $\mathcal{O}_s$ are generated and classified using the MR; and (3) a *verification stage* (lines 9 to 17) in which the subsumptions arising from just the $\mathcal{O}_s$ are verified using AR. We use notations $\mathbb{C}$ and $\mathbb{C}_o$ to denote the set of atomic concepts in $\mathcal{O}_o$ before and after normalization, respectively, and $\mathbb{C}^{\top,\bot} = \mathbb{C} \cup \{\top, \bot\}$.

In the normalization stage, the $\mathcal{ALCHO}$ ontology $\mathcal{O}_o$ is rewritten to contain only axioms of forms $\bigsqcap A_i \sqsubseteq \bigsqcup B_j$, $A \sqsubseteq \exists R.B$, $\exists R.A \sqsubseteq B$, $A \sqsubseteq \forall R.B$, $R \sqsubseteq S$, or $N_a \equiv \{a\}$. The procedure extends normalization in František *et al.* [17] by introducing for each $\{a\}$ a *nominal placeholder* N_a and adding a *nominal axiom* $N_a \equiv \{a\}$. We write $\mathbb{NP}$ for the set of all nominal placeholders after normalization. The transformation preserves subsumptions in $\mathcal{O}_o$. We assume all ontologies are normalized in the rest of the paper.

In the verification stage, there are some cases we hand over the classification work to AR: (1) $N_a \sqsubseteq \bot \in \mathcal{H}_s$; (2) $E \sqsubseteq \bot \in \mathcal{H}_s$ but $\mathcal{O}_o \not\models E \sqsubseteq \bot$, then for every $F \in \mathbb{C}^{\top,\bot}$, $E \sqsubseteq F \in \mathcal{H}_s$, while likely only few of them are in $\mathcal{H}_w$, thus $\mathcal{H}_s \setminus \mathcal{H}_w$ may be huge; (3) the fraction $\| \mathcal{H}_s \setminus \mathcal{H}_w \| \,/\, \| \mathbb{C} \|$ is greater than a threshold d. In the latter two cases,

Algorithm 1: HybridClassify($\mathcal{O}_o$)

Input: An $\mathcal{ALCHO}$ ontology $\mathcal{O}_o$
Output: The classification result of $\mathcal{O}_o$

normalize $\mathcal{O}_o$;
$\mathcal{O}_w \leftarrow \mathcal{O}_o$ with nominal axioms $N_a = \{a\}$ removed; /* Weakening */
$\mathcal{H}_w \leftarrow$ MR.classify($\mathcal{O}_w$); /* Classify the weakened ontology. */
$\mathcal{O}^+ \leftarrow$ getNominalStrAx($\mathcal{O}_w, \mathbb{NP}, \mathbb{C}^{\top,\bot}$); /* from Algorithm 4 */
remove all $E \sqsubseteq F$ from $\mathcal{H}_w$ where $\langle E, F\rangle \notin \mathbb{C}^{\top,\bot} \times \mathbb{C}^{\top,\bot}$;
if $\mathcal{O}^+ = \emptyset$ **then return** $\mathcal{H}_w$;
$\mathcal{O}_s \leftarrow \mathcal{O}_w \cup \mathcal{O}^+$;
$\mathcal{H}_s \leftarrow$ MR.classify($\mathcal{O}_s$); /* Classify the strengthened ontology. */
if $N_a \sqsubseteq \bot \in \mathcal{H}_s$ *for some* $N_a \in \mathbb{NP}$ **then return** AR.classify($\mathcal{O}_o$);
remove all $E \sqsubseteq F$ from $\mathcal{H}_s$ where $\langle E, F\rangle \notin \mathbb{C}^{\top,\bot} \times \mathbb{C}^{\top,\bot}$;
if $\parallel \mathcal{H}_s \setminus \mathcal{H}_w \parallel / \parallel \mathbb{C} \parallel > d$ **then return** AR.classify($\mathcal{O}_o$);
$\mathcal{H}_{ws} \leftarrow \mathcal{H}_w$;
foreach $E \sqsubseteq \bot \in \mathcal{H}_s \setminus \mathcal{H}_w$ **do**
  if AR.isSatisfiable($\mathcal{O}_o$, E) **then return** AR.classify($\mathcal{O}_o$);
  else add $E \sqsubseteq \bot$ into $\mathcal{H}_{ws}$;
foreach $E \sqsubseteq F \in \mathcal{H}_s \setminus \mathcal{H}_w$ *where* $F \neq \bot$ **do**
  if *not* AR.isSatisfiable($\mathcal{O}_o$, $E \sqcap \neg F$) **then** add $E \sqsubseteq F$ into $\mathcal{H}_{ws}$;
return $\mathcal{H}_{ws}$

the estimated work for the stage is more than using AR to classify $\mathcal{O}_o$. For (3) we set $d = 1.5$ in our implementation based on the experiments in [7].

In the main classification stage, the major work is to generate the $\mathcal{ALCH}$ ontologies $\mathcal{O}_w$ and $\mathcal{O}_s$. $\mathcal{O}_w$ is produced by simply removing all the nominal axioms of the form $N_a \equiv \{a\}$ from $\mathcal{O}_o$. Since $\mathcal{O}_w \subseteq \mathcal{O}_o$, $\mathcal{O}_o \models \mathcal{O}_w$ and so $\mathcal{H}_w \subseteq \mathcal{H}_o$, i.e. the classification result of $\mathcal{O}_w$ is sound w.r.t. $\mathcal{O}_o$.

Example 1. Consider the following normalized ontology $\mathcal{O}_{o,\mathsf{ex}}$ which we will use as a running example:

$$(1)\ A \sqsubseteq C \quad (2)\ A \sqsubseteq \exists R.E \quad (3)\ E \sqsubseteq N_a \quad (4)\ C \sqsubseteq \forall R.D$$
$$(5)\ D \sqsubseteq G \quad (6)\ A \sqsubseteq \exists S.N_a \quad (7)\ \underset{\sim\sim\sim\sim}{N_a \equiv \{a\}} \quad (8)\ \exists S.D \sqsubseteq F$$

Classification result of $\mathcal{O}_{o,\mathsf{ex}}$ is $\mathcal{H}_{o,\mathsf{ex}} = \{A \sqsubseteq F, A \sqsubseteq C, E \sqsubseteq N_a, D \sqsubseteq G\}$, where $A \sqsubseteq F$ is implied by axioms (1) – (4),(6) – (8). The weakened version $\mathcal{O}_{w,\mathsf{ex}}$ of $\mathcal{O}_{o,\mathsf{ex}}$ is obtained by removing nominal axiom (7). And its classification result $\mathcal{H}_{w,\mathsf{ex}} = \{A \sqsubseteq C, E \sqsubseteq N_a, D \sqsubseteq G\}$. We can see that $A \sqsubseteq F$, which requires (7) to imply, is missing in $\mathcal{H}_{w,\mathsf{ex}}$. We will see later how we add strengthening axioms to get $A \sqsubseteq F$ in $\mathcal{H}_{s,\mathsf{ex}}$.

The most difficult part of the procedure is to find $\mathcal{O}_s$ which entails no fewer subsumptions than $\mathcal{O}_o$. A sufficient condition is that for any $A, B \in \mathbb{C}^{\top,\bot}$ such that $\mathcal{O}_s \not\models A \sqsubseteq B$, there exists a model $\mathcal{I}$ of $\mathcal{O}_s$ satisfying $A \sqcap \neg B$, and it can be transformed to a model $\mathcal{I}'$ of $\mathcal{O}_o$ satisfying $A \sqcap \neg B$. Since $\mathcal{O}_s$ is obtained from $\mathcal{O}_w$ by adding strengthening axioms, every model $\mathcal{I}$ of $\mathcal{O}_s$ satisfies all axioms in $\mathcal{O}_o$ except possibly the nominal axioms, which require the interpretation of each $N_a \in \mathbb{NP}$ to have exactly one instance, whereas for an arbitrary model $\mathcal{I}$ of $\mathcal{O}_s$, $N_a^{\mathcal{I}}$ could have zero or multiple instances. However, if for each N_a, $N_a^{\mathcal{I}} \neq \emptyset$ and all the instances are "identical", they can be replaced by a single instance. Concretely, these instance have the same label set:

Definition 1 *Given an interpretation $\mathcal{I} = (\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$, an atomic concept A is called a* **label** *of an instance x if $x \in A^{\mathcal{I}}$. The set of all the labels of x is named the* **label set** *of x in $\mathcal{I}$, denoted by $LS(x, \mathcal{I})$.*

Such a replacement is called a *condensation* – it condenses all of the different instances into one instance, and thus it transforms $\mathcal{I}$ into a model that satisfies the nominal axiom for N_a. If such a condensation can be done for all nominal placeholders, then we can create a model for $\mathcal{O}_o$.

The strengthening axioms are designed to make these condensations possible. They have the form $N_a \sqsubseteq X$ and $N_a \sqcap X \sqsubseteq \bot$ computed by Algorithm 4, and thus they force X to be a label of the nominal instance N_a, or not to be one, respectively. By "manipulating" labels of nominal instances through these strengthening axioms, we can force them all to be identical in certain models, so that the condensations can occur.

The models we construct for transformation are variants of the canonical model constructed for $\mathcal{ALCH}$ ontologies [17]. Our model construction is introduced in Appendix A. Given $F \in \mathbb{C}^{\top,\bot}$, our approach ensures we can build a model of $\mathcal{O}_s$, which satisfies every $E \sqcap \neg F$ where $\mathcal{O}_s \not\models E \sqsubseteq F$. And every such model can be condensed to a model of $\mathcal{O}_o$ satisfying $E \sqcap \neg F$. Stating the contrapositive: if $\mathcal{O}_o \models E \sqsubseteq F$ then $\mathcal{O}_s \models E \sqsubseteq F$. Thus classification of $\mathcal{O}_s$ is complete. Soundness of our approach is ensured by the verification stage, and completeness is proved in the following section 2.1.

2.1 Condensing Labels and Completeness

Due to space limitation, we only list the important definitions, lemmas, property and theorems in the paper, all their proofs are in Appendix B. Here we briefly introduce some notations used later. We write $\mathcal{O}^+$ for any intermediate (including final) version of strengthening axioms, $\mathcal{O}_w^+$ for the corresponding intermediate (including final) version of strengthened ontology where $\mathcal{O}_w^+ = \mathcal{O}_w \cup \mathcal{O}^+$. We write $\mathcal{O}_s$ for the final version of strengthened ontology. Given $E, F \in \mathbb{C}^{\top,\bot}$ such that $\mathcal{O}_w^+ \not\models E \sqsubseteq F$, a canonical model $\mathcal{I}_{[\mathcal{O}_w^+, \prec_F]}$ of $\mathcal{O}_w^+$ is constructed by first computing a *saturation* $\mathbf{S}_{\mathcal{O}_w^+}$ of $\mathcal{O}_w^+$ and then defining a model based on a total order $\prec_F$. The definition of construction of the canonical model is in Appendix A. The computation of saturation is as follows: Given $\mathcal{O}_w^+$, the saturation $\mathbf{S}_{\mathcal{O}_w^+}$ is initialized as

$$\{\mathsf{init}(A) \mid A \in \mathbb{C}\} \cup \{\mathsf{init}(N_a) \mid N_a \in \mathbb{NP}\}$$

Then $\mathbf{S}_{\mathcal{O}_w^+}$ is expanded by iteratively applying the inference rules in Table 1 and adding the conclusions into $\mathbf{S}_{\mathcal{O}_w^+}$ until reaching a fixpoint. During this process, existing axioms in $\mathbf{S}_{\mathcal{O}_w^+}$ are used as premises and axioms in $\mathcal{O}_w^+$ are used as side conditions. $\mathbf{S}_{\mathcal{O}_w^+}$ contains axioms of the forms $\mathsf{init}(H)$, $H \sqsubseteq M \sqcup A$ and $H \sqsubseteq M \sqcup \exists R.K$. Note Table 1 is modified from Table 3 in [17] by using $\mathbf{R}_{\mathbf{A}}^+$ and $\mathbf{R}_{\mathbf{init}}$ to initialize contexts whenever necessary. The conjunction H that occurs in the premises or conclusions of the inference rules is called the context of the inference.

Table 1. Complete Inference Rules for Normalized $\mathcal{ALCH}$ ontologies

$$\mathbf{R}_{\mathbf{A}}^{+}\ \frac{\mathsf{init}(H)}{H \sqsubseteq A} : A \in H \qquad \mathbf{R}_{\mathbf{A}}^{-}\ \frac{H \sqsubseteq N \sqcup A}{H \sqsubseteq N} : \neg A \in H \qquad \mathbf{R}_{\mathbf{init}}\ \frac{H \sqsubseteq M \sqcup \exists R.K}{\mathsf{init}(K)}$$

$$\mathbf{R}_{\sqcap}^{\mathbf{n}}\ \frac{\{H \sqsubseteq N_i \sqcup A_i\}_{i=1}^{n}}{H \sqsubseteq \bigsqcup_{i=1}^{n} N_i \sqcup M} : \textstyle\bigsqcap_{i=1}^{n} A_i \sqsubseteq M \in \mathcal{O}_w^+ \qquad \mathbf{R}_{\exists}^{+}\ \frac{H \sqsubseteq N \sqcup A}{H \sqsubseteq N \sqcup \exists R.B} : A \sqsubseteq \exists R.B \in \mathcal{O}_w^+$$

$$\mathbf{R}_{\exists}^{-}\ \frac{H \sqsubseteq M \sqcup \exists R.K \quad K \sqsubseteq N \sqcup A}{H \sqsubseteq M \sqcup B \sqcup \exists R.(K \sqcap \neg A)} : \begin{array}{l} \exists S.A \sqsubseteq B \in \mathcal{O}_w^+ \\ R \sqsubseteq_{\mathcal{O}}^{*} S \end{array} \qquad \mathbf{R}_{\exists}^{\perp}\ \frac{H \sqsubseteq M \sqcup \exists R.K \quad K \sqsubseteq \perp}{H \sqsubseteq M}$$

$$\mathbf{R}_{\forall}\ \frac{H \sqsubseteq M \sqcup \exists R.K \quad H \sqsubseteq N \sqcup A}{H \sqsubseteq M \sqcup N \sqcup \exists R.(K \sqcap B)} : \begin{array}{l} A \sqsubseteq \forall S.B \in \mathcal{O}_w^+ \\ R \sqsubseteq_{\mathcal{O}}^{*} S \end{array}$$

Definition 2 *In an interpretation $\mathcal{I} = (\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$, an atomic concept L is called a* **condensing label** *if (1) $L^{\mathcal{I}} \neq \emptyset$ and (2) for any $x, y \in L^{\mathcal{I}}$, $LS(x, \mathcal{I}) = LS(y, \mathcal{I})$.*

If a label applied to some instance is a condensing label then every instance to which it applies has the same label set. This means the label sets of all such instances are identical and can be condensed into one instance.

Definition 3 *Given a model $\mathcal{I}$ of an $\mathcal{ALCHO}$ ontology $\mathcal{O}$, a concept L in $\mathcal{O}$ and an individual name x_L, we define a* **condensation** *function $condense(L, x_L, \mathcal{I})$ that transforms $\mathcal{I}$ into an interpretation $\mathcal{I}' = (\triangle^{\mathcal{I}'}, \cdot^{\mathcal{I}'})$ as follows:*

1). Let n be a fresh instance which is not in $\triangle^{\mathcal{I}}$, and r be a replacement function

$$r(x) = \begin{cases} n & x \in L^{\mathcal{I}} \\ x & \text{otherwise} \end{cases}$$

2). $\triangle^{\mathcal{I}'} = \{r(x) \mid x \in \triangle^{\mathcal{I}}\}$

3). For each concept A, role R and individual o in $\mathcal{O}$,

$A^{\mathcal{I}'} = \{r(x) \mid x \in A^{\mathcal{I}}\}, R^{\mathcal{I}'} = \{(r(x), r(y)) \mid (x, y) \in R^{\mathcal{I}}\}, o^{\mathcal{I}'} = r(o^{\mathcal{I}}), x_L^{\mathcal{I}'} = n$

We say that each $x \in L^{\mathcal{I}}$ is **condensed** *into n. We also say $\mathcal{I}$ is condensed to $\mathcal{I}'$.*

Definition 4 *H is a* **potentially supporting context** *of A in $\mathcal{O}_w^+$ if $H \sqsubseteq M \sqcup A \in \mathbf{S}_{\mathcal{O}_w^+}$.*

Definition 5 *A concept X is called a* **Potentially Supporting concept (PS)** *of some A in $\mathcal{O}_w^+$ if either X or $\neg X$ is a conjunct of a potentially supporting context H of A in $\mathcal{O}_w^+$. The set of all* PS *of A in $\mathcal{O}_w^+$ is denoted by $\mathsf{PS}_{[A, \mathcal{O}_w^+]}$.*

Example 2. (PS) Consider the $\mathcal{ALCH}$ ontology $\mathcal{O}_{w,\mathsf{ex}}$ in Example 1, it can be seen as an ontology $\mathcal{O}_{w,\mathsf{ex}}^+$ where $\mathcal{O}_{\mathsf{ex}}^+ = \emptyset$, and $\mathbb{NP} = \{N_a\}$. The potentially supporting contexts of N_a are N_a, E and $E \sqcap D$. So $\mathsf{PS}_{[N_a, \mathcal{O}_{w,\mathsf{ex}}]} = \{N_a, D, E\}$.

Property 6 *We say $\mathcal{O}_w^+$ is* **decisive** *if for each $N_a \in \mathbb{NP}$, each $X \in \mathsf{PS}_{[N_a, \mathcal{O}_w^+]}$, either $\mathcal{O}_w^+ \models N_a \sqsubseteq X$ or $\mathcal{O}_w^+ \models N_a \sqcap X \sqsubseteq \perp$ holds.*

Lemma 7 *Given O_w^+ and $N_a \in \mathbb{NP}$, if (1) O_w^+ is decisive, and (2) $O_w^+ \not\models N_a \sqsubseteq \bot$. Then for any $F \in \mathbb{C}^{\top,\bot}$, N_a is a condensing label in $\mathcal{I}_{[O_w^+,<_F]}$.*

Lemma 8 *Let $\mathcal{I}$ be a model of an $\mathcal{ALCHO}$ ontology O satisfying $E \sqcap \neg F, E, F \in \mathbb{C}^{\top,\bot}$, where L is a condensing label in $\mathcal{I}$. Then $\mathcal{I}' = condense(L, x_L, \mathcal{I})$ is a model of $O \cup \{L = \{x_L\}\}$ satisfying $E \sqcap \neg F$.*

Lemma 9 *Given some O_w^+ and $F \in \mathbb{C}^{\top,\bot}$, if in $\mathcal{I}_{[O_w^+,<_F]}$ every $N_a \in \mathbb{NP}$ is a condensing label, then for each $E \in \mathbb{C}^{\top,\bot}$ s.t. $O_w^+ \not\models E \sqsubseteq F$, there is a model of O_o satisfying $E \sqcap \neg F$.*

Theorem 10 *If the O_w^+ we compute satisfies: (1) O_w^+ is decisive and (2) all $N_a \in \mathbb{NP}$ are satisfiable in O_w^+. Then the classification result of O_w^+ is complete w.r.t. O_o.*

Proof. Let $E, F \in \mathbb{C}^{\top,\bot}$ be concepts such that $O_w^+ \not\models E \sqsubseteq F$. Since conditions (1) and (2) of Lemma 7 are satisfied, all $N_a \in \mathbb{NP}$ are condensing labels in the canonical model $\mathcal{I}_{[O_w^+,<_F]}$. By Lemma 9, the model can be condensed to a model $\mathcal{I}'$ of O_o for $E \sqcap \neg F$, proving $O_o \not\models E \sqsubseteq F$. So the classification result of O_w^+ is complete w.r.t. O_o. □

In the next section, we demonstrate how we will compute an O_w^+ satisfying condition (1) in Theorem 10 i.e., is decisive. Such an O_w^+ can also satisfy condition (2) in most cases as in our experiments, therefore completeness is achieved. If in a few cases that such an O_w^+ ends up not satisfying condition (2), then we hand over the work to AR. The classification result is still complete.

3 Computing Strengthened Ontology

In this section we show how to create a decisive O_w^+. The property suggests that the strengthening axioms should be of the form $N_a \sqsubseteq X$ or $N_a \sqcap X \sqsubseteq \bot$. We first briefly introduce the strengthening axioms selection function chooseStrAxiom. Based on that, we explain an initial idea to create a decisive O_w^+. Then, we introduce how to compute $\mathsf{OPS}_{[N_a,O_w^+]}$ which is an overestimation of $\mathsf{PS}_{[N_a,O_w^+]}$ from O_w^+ without doing saturation, and give the naive and improved algorithms to compute O^+.

Definition 11 *A strengthening axiom selection* chooseStrAxiom *is a function that takes an $N_a \in \mathbb{NP}$, $X \in \mathbb{C}_o$, and O_w and returns:*

1. *$\emptyset$ only if $O_w \models N_a \sqsubseteq X$ or $O_w \models N_a \sqcap X \sqsubseteq \bot$;*
2. *a choice between $N_a \sqsubseteq X$ or $N_a \sqcap X \sqsubseteq \bot$.*

Initial idea for computing a decisive O_w^+: To create such an ontology, a straightforward idea is to start from O_w, generate saturation $\mathbf{S}_{O_w}$ of O_w, obtain $\mathsf{PS}_{[N_a,O_w]}$ from $\mathbf{S}_{O_w}$ for all $N_a \in \mathbb{NP}$. For each pair $\langle N_a, X\rangle$, $X \in \mathsf{PS}_{[N_a,O_w]}$, we add a strengthening axiom determined by chooseStrAxiom function into O^{1+}. Then, add O^{1+} into O_w and obtain O_w^{1+}. Next, we generate saturation again based on O_w^{1+}. Since the impact of O^{1+}, $\mathsf{PS}_{[N_a,O_w^{1+}]} \supseteq \mathsf{PS}_{[N_a,O_w]}$. Then we repeat the above process until a fixpoint at which $\mathsf{PS}_{[N_a,O_w^{i+1+}]} = \mathsf{PS}_{[N_a,O_w^{i+}]}$ for all $N_a \in \mathbb{NP}$, we call the final O^{i+1+} as O^+, final O_w^{i+1+} as O_s which is decisive. However to implement such a procedure generating the saturation for $\mathsf{PS}_{[N_a,O_w^+]}$ is costly. Thus, instead of $\mathsf{PS}_{[N_a,O_w^+]}$, we compute the overestimated $\mathsf{PS}_{[N_a,O_w^+]}$ called $\mathsf{OPS}_{[N_a,O_w^+]}$.

Definition 12 *Given an ontology* O_w^+, $\mathsf{OPS}_{[N_a,O_w^+]}$ *is a set of* **overestimated potentially supporting concepts** *we compute such that* $\mathsf{OPS}_{[N_a,O_w^+]} \supseteq \mathsf{PS}_{[N_a,O_w^+]}$ *for each* $N_a \in \mathbb{NP}$.

Algorithm 2: getOPS

Input: Normalized $\mathcal{ALCH}$ ontology O_w^+, a concept X, a set of nominal placeholders $\mathbb{NP}$, the set of atomic classes U in the original ontology

Output: A pair $\langle \mathsf{OPS}_{[X,O_w^+]}, \mathsf{Pri}_{[X,O_w^+]} \rangle$

1 $\mathsf{OPS}_{[X,O_w^+]} \leftarrow \emptyset$; $\mathsf{Pri}_{[X,O_w^+]} \leftarrow \emptyset$; $\mathsf{ToProcess} \leftarrow \{X\}$; $\mathsf{Exists} \leftarrow \emptyset$;
2 **while** $\mathsf{ToProcess} \neq \emptyset$ **do**
3   take out a label W from ToProcess;
4   **if** $W \notin \mathsf{Pri}_{[X,O_w^+]}$ **then**
5   add W to $\mathsf{Pri}_{[X,O_w^+]}$;
6   **if** $W = \top$ **then** stop the procedure and use AR to do the classification work;
7   **foreach** $\bigsqcap A_i \sqsubseteq M \bigsqcup W \in O_w^+$ **do** select one A_i and add it into ToProcess;
8   **foreach** $\exists S.Y \sqsubseteq W \in O_w^+$ *and* $R \sqsubseteq_O^* S$ *and* $B \sqsubseteq \exists R.Z \in O_w^+$ **do**
9   add B into ToProcess;
10 **foreach** $W \in \mathsf{Pri}_{[X,O_w^+]}$ **do**
11   **if** $W \in U$ *or* $W \in \mathbb{NP}$ **then** add W to $\mathsf{OPS}_{[X,O_w^+]}$;
12   **foreach** $Y \sqsubseteq \forall S.W \in O_w^+$ *and* $R \sqsubseteq_O^* S$ *and* $B \sqsubseteq \exists R.Z \in O_w^+$ **do** add $\exists R.Z$ to Exists;
13   **foreach** $B \sqsubseteq \exists R.W \in O_w^+$ **do** add $\exists R.W$ to Exists;
14 **foreach** $\exists R.W \in \mathsf{Exists}$ *and* $R \sqsubseteq_O^* S$ **do**
15   add W to $\mathsf{OPS}_{[X,O_w^+]}$;
16   **foreach** $Y \sqsubseteq \forall S.Z \in O_w^+$ **do** add Z to $\mathsf{OPS}_{[X,O_w^+]}$;
17   **foreach** $\exists S.Z \sqsubseteq Y \in O_w^+$ **do** add Z to $\mathsf{OPS}_{[X,O_w^+]}$;
18 **return** $\langle \mathsf{OPS}_{[X,O_w^+]}, \mathsf{Pri}_{[X,O_w^+]} \rangle$

We use Algorithm 2 to compute $\mathsf{OPS}_{[X,O_w^+]}$. Based on Definition 4 and 5, $\mathsf{PS}_{[X,O_w^+]}$ includes all the atoms of H such that $H \sqsubseteq M \sqcup X \in \mathbf{S}_{O_w^+}$. Without a real procedure generating the saturation for $\mathsf{PS}_{[X,O_w^+]}$ as explained above, we actually compute $\mathsf{OPS}_{[X,O_w^+]}$ based on analyzing the relationships among the premises, side conditions and conclusions of the inference rules in Table 1. The procedure of Algorithm 2 can be divided into three parts:

1. (line 2 to 9) We first conduct a search in the converse direction of all possible derivation paths(Definition 16 in Appendix B) of a conclusion $H \sqsubseteq M \sqcup X$ for all possible Hs. In the search we maintain a set $\mathsf{Pri}_{[X,O_w^+]}$. Each concept $W \in \mathsf{Pri}_{[X,O_w^+]}$ may be necessary to the derivation of X, and appears prior to X in the derivation path. More precisely, W corresponds to some potential intermediate conclusion $H \sqsubseteq M' \sqcup W$ which is a necessary conclusion for deriving $H \sqsubseteq M \sqcup X$. Since for any context H, $H \sqsubseteq A$ is the only conclusion that can be derived from $\mathsf{init}(H)$, at least one of such A, which is a conjunct of H, is in $\mathsf{Pri}_{[X,O_w^+]}$. We write C_H^X for such A. $\mathsf{Pri}_{[X,O_w^+]}$ contains at least one conjunct C_H^X for any potentially supporting context H of X.
2. (lines 10 to 13) Check each concept $W \in \mathsf{Pri}_{[X,O_w^+]}$ to see whether it can be a conjunct C_H^X of some potential supporting context H of X. If it can, we find the first conjunct

C^1_H of H from C^X_H. C^1_H (see H^* in Lemma 13 in Appendix B), which is either in U or $\mathbb{NP}$ in line 11 or the filler of concepts in Exists from line 12 to 13, is the initial concept H starts from.

3. (line 14 to 17) Find all the other conjuncts of H based on C^1_H by searching along the derivation paths.

Lemma 13 *Given O^+_w and a concept A, $\mathsf{OPS}_{[A,O^+_w]}$ returned by Algorithm 2 preserves* $\mathsf{OPS}_{[A,O^+_w]} \supseteq \mathsf{PS}_{[A,O^+_w]}$.

Example 3. (OPS) Consider again the ontology $O_{w,\mathsf{ex}}$ in Example 1.

The Execution of Alg. 2: $\mathsf{getOPS}(O_{w,\mathsf{ex}}, N_a, \{N_a\}, \{A, C, D, E, F, G\})$

Line 2 to 9 $\quad \mathsf{Pri}_{[N_a,O_{w,\mathsf{ex}}]} = \{N_a, E\}$

Line 11 $\quad \mathsf{OPS}_{[N_a,O_{w,\mathsf{ex}}]} = \{N_a, E\}$

Line 13 $\quad \mathsf{Exists} = \{\exists R.E\}$

Line 16 $\quad \mathsf{OPS}_{[N_a,O_{w,\mathsf{ex}}]} = \{N_a, E, D\}$

Line 18 $\quad$ Return $\mathsf{OPS}_{[N_a,O_{w,\mathsf{ex}}]} = \{N_a, E, D\}$, $\mathsf{Pri}_{[N_a,O_{w,\mathsf{ex}}]} = \{N_a, E\}$

$\mathsf{OPS}_{[N_a,O_{w,\mathsf{ex}}]} = \mathsf{PS}_{[N_a,O_{w,\mathsf{ex}}]} = \{N_a, E, D\}$ $\quad$ so $\mathsf{OPS}_{[N_a,O_{w,\mathsf{ex}}]} \supseteq \mathsf{PS}_{[N_a,O_{w,\mathsf{ex}}]}$

For each $X \in \mathsf{OPS}_{[X,O^{i+}_w]}$, we will use function $\mathsf{chooseStrAxiom}$ to return a strengthening axiom. Note in Definition 11 for $\mathsf{chooseStrAxiom}$, how can we know $O_w \models N_a \sqsubseteq X$ or $O_w \models N_a \sqcap X \sqsubseteq \bot$? Actually in Algorithm 1, line 4 (we used $\mathsf{getNominalStrAx}$, if using $\mathsf{getNominalStrAxNaive}$, it is the same) executes $\mathsf{getOPS}(O_w, N_a, \mathbb{NP}, U)$ internally before line 3. Then for each $N_a \in \mathbb{NP}$, each $X \in \mathsf{OPS}_{[N_a,O_w]}$, we add a testing axiom $X_a \sqsubseteq N_a \sqcap X$ into O_w in the first round classification and obtain $\mathcal{H}_w$, where X_a is a fresh concept for O_w. In Algorithm 1 and Definition 11, we ignore this detail and only mention using O_w just for simplifying explanation. If $X_a \sqsubseteq \bot$ is found in $\mathcal{H}_w$, then we simply say $O_w \models N_a \sqcap X \sqsubseteq \bot$. If $N_a \sqsubseteq X$ or $N_a \sqcap X \sqsubseteq \bot$ is implied by $\mathcal{H}_w$, we do not add any strengthening axiom for X into O^{i+1+}. We remove the extra testing results from $\mathcal{H}_w$ in line 5 of Algorithm 1. When choosing between $N_a \sqsubseteq X$ and $N_a \sqcap X \sqsubseteq \bot$, we use the heuristics that if X corresponds to a union concept before normalization, and $N_a \sqsubseteq X$ is not implied, then we add $N_a \sqcap X \sqsubseteq \bot$. For other cases, we add $N_a \sqsubseteq X$.

Example 4. For the concepts in $\mathsf{OPS}_{[N_a,O_{w,\mathsf{ex}}]}$, since none of the 4 axioms $N_a \sqsubseteq E$, $E_a \sqsubseteq \bot$, $N_a \sqsubseteq D$ or $D_a \sqsubseteq \bot$ is in $\mathcal{H}_{w,\mathsf{ex}}$, and assume E and D do not associate union concepts before normalization, we add $N_a \sqsubseteq E$ and $N_a \sqsubseteq D$ as strengthening axioms O^{1+}_{ex}.

Using $\mathsf{OPS}_{[X,O^+_w]}$ instead of $\mathsf{PS}_{[X,O^+_w]}$, we design a naive Algorithm 3 based on the above **Initial Idea** to generate O^+ for O^+_w so that O^+_w is decisive. The execution process is as follows:

1). $O^{0+} = \emptyset$. $\quad$ 2). Compute $\mathsf{OPS}_{[N_a,O^{i+}_w]}$ for $\forall N_a, N_a \in \mathbb{NP}$ for the O^{i+}.

3). $O^{i+1+} = \{\mathsf{chooseStrAxiom}(N_a, X) \mid \forall N_a, X, N_a \in \mathbb{NP}, X \in \mathsf{OPS}_{[N_a,O^{i+}_w]}\}$

In each loop, we add the new O^{i+} into O_w and generate O^{i+}_w. When the process converges, $O^{i+1+} = O^{i+}$. Since O^{i+1+} is computed from $\mathsf{OPS}_{[N_a,O^{i+}_w]}$, and $\mathsf{OPS}_{[N_a,O^{i+}_w]} \supseteq \mathsf{PS}_{[N_a,O^{i+}_w]}$ for all N_a and for any i, thus Algorithm 3 guarantees that O^+_w is decisive.

Theorem 14 *Let O^+ be strengthening axiom produced from Algorithm 3, $O^+_w = O_w \cup O^+$ is decisive.*

Algorithm 3: getNominalStrAxNaive (Naive algorithm for computing strengthening axioms)

Input: Normalized $\mathcal{ALCH}$ ontology O_w, a set of nominal placeholders $\mathbb{NP}$, the set of atomic classes U in the original ontology
Output: Strengthening axioms O^+

1 $O^+ \leftarrow \emptyset$;
2 **repeat**
3   newAxioms $= \emptyset$;
4   **foreach** $N_a \in \mathbb{NP}$ **do**
5         $O_w^+ \leftarrow O_w \cup O^+$;
6         $\langle \mathsf{OPS}_{[X,O_w^+]}, \mathsf{Pri}_{[X,O_w^+]} \rangle \leftarrow$ getOPS$(O_w^+, N_a, \mathbb{NP}, U)$; `/* from Algorithm 2 */`
7   **foreach** $X \in \mathsf{OPS}_{[X,O_w^+]}$ **do**
8   newAxioms $\leftarrow$ newAxioms $\cup$ chooseStrAxiom(N_a, X);
9     $O^+ \leftarrow O^+ \cup$ newAxioms;
10 **until** newAxioms $= \emptyset$;
11 **return** O^+

Theorem 15 *Let O^+ be strengthening axioms computed from Algorithm 4, $O_w^+ = O_w \cup O^+$ is decisive.(Proof see Appendix B, the following is just an intuitive explanation)*

Algorithm 4 improves Algorithm 3 by avoiding repetitive search process. In each loop of Algorithm 3, Algorithm 2 getOPS bases on the axioms of the input O_w^{i+1+} to compute $\mathsf{OPS}_{[N_a,O_w^{i+1+}]}$ for N_a, where only search on $O_w^{i+1+} \setminus O_w^{i+}$ is new, the majority work – search on O_w is repeated in each iteration, while Algorithm 4 improves this and only executes getOPS based on O_w for once in line 3. In Algorithm 2 and 3, a strengthening axiom α can take effect only in the following situation: If $\alpha = \{N_b \sqsubseteq X\} \in O_w^{i+}$, $X \in \mathsf{Pri}_{[N_a,O_w^{i+}]}$, then $N_b \in \mathsf{OPS}_{[N_a,O_w^{i+}]}$. If chooseStrAxiom$(N_a, N_b)$ return $N_a \sqsubseteq N_b$ in O_w^{i+1}, then $N_a \in \mathsf{OPS}_{[N_b,O_w^{i+1+}]}$. That means because of α, in the end, $\mathsf{OPS}_{[N_a,O_w^+]} = \mathsf{OPS}_{[N_b,O_w^+]}$.
In Algorithm 4, without really computing each of O_w^{i+} and repeatedly search on them, we achieve similar results based on the merge criteria in line 6 and the merge operation from lines 7 to line 7. Assume $N_b \in g_i.nominals$ and $N_a \in g_j.nominals$, $X \in g_i.ops \cap g_j.pri \neq \emptyset$ in line 6. Then, $X \in g_i.ops$ means $N_b \sqsubseteq X$ will possibly be a strengthening axiom, $X \in g_j.pri$ means X is possibly in N_a's Pri, then g_i and g_j are merged into one group, and finally $\mathsf{OPS}_{[N_a,O_w^+]} = \mathsf{OPS}_{[N_b,O_w^+]}$, too, similar with Algorithm 3.

Example 5. In Algorithm 3, $O_{ex}^{1+} = O_{ex}^{2+}$. Thus, the loop from line 2 to 9 repeats twice. In Algorithm 4, since $\mathbb{NP} = \{N_a\}$, the merge process from line 6 to 7 does not happen. For both algorithms, that means O_{ex}^{1+} does not have impact in later saturation.

Note the naive Algorithm 3 is only used for demonstrating our initial idea. In our implementation, we used the improved Algorithm 4. All three Algorithms 2, 3 and 4 have polynominal complexity. In Algorithm 2, the number of iterations in all levels of loops are bounded by the number of axioms or concepts in O_w, and thus have polynomial complexity. In Algorithm 3, the number of iterations is bounded by the size of O^+, which is also polynomial. In Algorithm 4, the merge loop can only continue for at most $||\mathbb{NP}||$ times. So all the algorithms are polynomial of the size of O_w and terminate.

Algorithm 4: getNominalStrAx (Calculate strengthening axioms for nominals)

Input: Normalized $\mathcal{ALCH}$ ontology O_w, a set of nominal placeholders $\mathbb{NP}$, the set of atomic classes U in the original ontology

Output: Strengthening axioms O^+

groups $\leftarrow \emptyset$;
foreach $N_a \in \mathbb{NP}$ **do**
  $\langle \mathsf{OPS}_{[N_a,O_w]}, \mathsf{Pri}_{[N_a,O_w]} \rangle \leftarrow \mathsf{getOPS}(O_w, N_a, \mathbb{NP}, U)$; `/* from Algorithm 2 */`
  create a group g with $g.nominals = \{N_a\}$, $g.ops = \mathsf{OPS}_{[N_a,O_w]}$, $g.pri = \mathsf{Pri}_{[N_a,O_w]}$;
  add g into groups;
while *there exists* $g_i, g_j \in$ groups *such that* $g_i.ops \cap g_j.pri \neq \emptyset$ **do**
  merge g_i, g_j into one group g, whose properties are unions of corresponding properties of g_i and g_j; remove g_i, g_j from groups and add g;
foreach $g \in$ groups **do**
  foreach $N_a \in g.nominals$ **do**
    foreach $X \in g.ops$ **do**
      $O^+ \leftarrow O^+ \cup \mathsf{chooseStrAxiom}(N_a, X)$;
return O^+

Example 6. : The Overall Execution Results from Alg. 1:

(1) Weakened Ontology O_w: $N_a \equiv \{a\}$ removed   Line 2 of Alg. 1
(2) Class Hierarchy $\mathcal{H}_w$: $A \sqsubseteq C$ $E \sqsubseteq N_a$ $D \sqsubseteq G$ Line 3 of Alg. 1
(3) Strengthening axioms O^+: two axioms added: $N_a \sqsubseteq E$ $N_a \sqsubseteq D$
(4) For (3): Line 4 of Alg. 1 return from Alg. 4 For (5): Line 8 of Alg. 1
(5) Class Hierarchy $\mathcal{H}_s$: $A \sqsubseteq C$ $E \sqsubseteq N_a$ $D \sqsubseteq G$ $A \sqsubseteq F$ $N_a \sqsubseteq E$ $N_a \sqsubseteq D$
(6) Verify the following 6 pairs (indirect subsumptions included): Line 16 to 17
(7) $A \sqsubseteq F$ $E \sqsubseteq D$ $E \sqsubseteq G$ $N_a \sqsubseteq D$ $N_a \sqsubseteq E$ $N_a \sqsubseteq G$, 1 pair validated: $A \sqsubseteq F$
(8) Class Hierarchy H_{ws}=$\mathcal{H}_o$: $A \sqsubseteq C$ $A \sqsubseteq F$ $E \sqsubseteq N_a$ $D \sqsubseteq G$. Line 18 of Alg. 1

4 Related Work

Optimization techniques for ontology classification have been extensively studied in the literature [4,16,7,12]. For tableau-based reasoners, Enhanced Traversal (ET) [4] and KP [16,7] are the most widely used techniques. Optimizations for consequence-based classification of $\mathcal{ELO}$ ontologies were also studied [12], and the most effective technique is overestimation. Firstly, the algorithm saturates the ontology using inference rules for $\mathcal{EL}$ and obtains sound subsumptions. Next, potential subsumptions are obtained by continuing saturation with a new overestimation rule added. Finally, the potential subsumptions are checked using a sound and complete but slower procedure for $\mathcal{ELO}$. Comparing with this procedure, we support a more expressive DL $\mathcal{ALCHO}$.

In the area of hybrid reasoning, Romero et al. [1,2] proposed classification based on modules given to a $\mathcal{SROIQ}$ reasoner R and an efficient $\mathcal{L}$-reasoner $R_{\mathcal{L}}$ supporting a fragment $\mathcal{L}$ of $\mathcal{SROIQ}$. Given O_o, they find a set of classes $\Sigma^{\mathcal{L}}$ whose superclasses in O_o can be computed by classifying a subset $\mathcal{M}^{\mathcal{L}}$ of O_o in DL $\mathcal{L}$. The superclasses of remaining classes are computed using R. However, because of the restriction of locality-based modular approach used for computing $\Sigma^{\mathcal{L}}$, nominal axioms $N_a \equiv \{a\}$ cannot be moved out from $\mathcal{M}^{\mathcal{L}}$ [6]. Therefore, in order to guarantee completeness, either $R_{\mathcal{L}}$ supports nominals or all the work is assigned to R. In current implementation of MORe, $R_{\mathcal{L}}$ does not support non-safe [12] use of nominals in the Galen ontologies, so R has to do all the work. In sum, module extraction technique MORe uses for classification is a more general technique that is primarily intended for ontologies in DLs whose expressivity is beyond ALCHO. In contrast, our approach currently only supports language $\mathcal{ALCHO}$, combines the two reasoners differently, handles nominals, and improves on its full reasoner more often for complex and highly cyclic ontologies.

Our approach generally belongs to theory approximation [15]. Yuan *et. al.* [14] encoded $\mathcal{SROIQ}$ ontologies into $\mathcal{EL}^{++}$ with additional data structures, and classified by a tractable, sound but incomplete algorithm [14]. A strengthened approximation of $\mathcal{SROIQ}$ TBoxes with the OWL 2 RL profile [21] is used for query answering.

5 Empirical Results and Conclusion

We have implemented our prototype hybrid classifier WSClassifier in Java. The classifier uses ConDOR as the main $\mathcal{ALCH}$ reasoner and HermiT as the assistant reasoner for DL $\mathcal{ALCHO}$. WSClassifier adopts a well-known preprocessing step to eliminate transitive roles [10], hence supports DL $\mathcal{SHO}$. We set the Java heap space to 12GB and the time limit to 9 days for all reasoners. We compared the classification time of WSClassifier with main-stream tableau-based reasoners and another hybrid reasoner MORe on all large and highly cyclic ontologies available to us, on the ORE dataset and on some proposed variants. For classifying FMA-constitutionalPartForNS(FMA-C) which is the only real-world large and highly cyclic ontologies with nominals we have access to, WSClassifier used 21.2 seconds, while HermiT used 140,882 seconds with configuration of simple core blocking and individual reuse. Other reasoners did not get a result because they ran out either of time or memory. We put the evaluation result Table 2 of comparison of classification performance in Appendix C. The results of Table 2 show, excluding ORE dataset, that WSClassifier is significantly faster than the tableau-based reasoners on 7 out of 10 ontologies. For the other 3 of 10 ontologies, WSClassifier detected that strengthening axioms made some concepts unsatisfiable in O_s, and so failed over to HermiT. We see a major speedup for WSClassifier on ORE's FMA-lite which is highly cyclic and is our targeted ontology. For the remaining 112 ORE ontologies which are not highly cyclic and thus not our target, WSClassifier failed over to HermiT on one of them. Our average reasoning time on these 112 ontologies is longer than that of the other reasoners mainly due to our overheads: computing normalized axioms and transmitting the ontology to and from ConDOR.

We have presented a hybrid reasoning technique for soundly and completely classifying an $\mathcal{ALCHO}$ ontology based on a weakening and strengthening approach. The input ontology is approximated by two $\mathcal{ALCH}$ ontologies, one weakened O_w and one strengthened O_s, which are classified by a fast consequence-based reasoner. The subsumptions of O_w and O_s are a subset and a superset of the subsumptions of the original ontology, respectively. Subsumptions implied by O_s but not by O_w are further checked by a (slower) $\mathcal{ALCHO}$ reasoner. This approach is possibly applied to different language classes, each requiring different strengthening axioms. The implementation can be improved with heuristics for a tighter $\mathsf{OPS}_{[N_a, O_w^+]}$ and better strengthening axioms.

References

1. Armas Romero, A., Cuenca Grau, B., Horrocks, I.: Modular combination of reasoners for ontology classification. In: Proc. of DL (2012)
2. Armas Romero, A., Cuenca Grau, B., Horrocks, I.: MORe: Modular combination of OWL reasoners for ontology classification. In: Proc. of ISWC. pp. 1–16 (2012)
3. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proc. of IJCAI. pp. 364–369 (2005)
4. Baader, F., Hollunder, B., Nebel, B., Profitlich, H.J., Franconi, E.: An empirical analysis of optimization techniques for terminological representation systems. Applied Intelligence 4(2), 109–132 (1994)
5. Baader, F., Lutz, C., Suntisrivaraporn, B.: CEL – a polynomial-time reasoner for life science ontologies. In: Proc. of IJCAR. pp. 287–291 (2006)
6. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: A logical framework for modularity of ontologies. In: Proc. of IJCAI. pp. 298–303 (2007)
7. Glimm, B., Horrocks, I., Motik, B., Shearer, R., Stoilos, G.: A novel approach to ontology classification. J. Web Semantics 14, 84–101 (2012)
8. Haarslev, V., Möller, R.: RACER system description. In: Proc. of IJCAR. pp. 701–705 (2001)
9. Horrocks, I., Sattler, U.: A Tableau decision procedure for $\mathcal{SHOIQ}$. J. Automated Reasoning 39(3), 249–276 (2007)
10. Kazakov, Y.: Consequence-driven reasoning for Horn $\mathcal{SHIQ}$ ontologies. In: Proc. of IJCAI. pp. 2040–2045 (2009)
11. Kazakov, Y., Krötzsch, M., Simančík, F.: Concurrent classification of $\mathcal{EL}$ ontologies. In: Proc. of ISWC. pp. 305–320 (2011)
12. Kazakov, Y., Krötzsch, M., Simančík, F.: Practical reasoning with nominals in the $\mathcal{EL}$ family of description logics. In: Proc. of KR. pp. 264–274 (2012)
13. Motik, B., Shearer, R., Horrocks, I.: Hypertableau reasoning for description logics. J. Artificial Intelligence Research 36(1), 165–228 (2009)
14. Ren, Y., Pan, J.Z., Zhao, Y.: Soundness preserving approximation for TBox reasoning. In: Proc. of AAAI (2010)
15. Selman, B., Kautz, H.: Knowledge compilation and theory approximation. J. ACM 43(2), 193–224 (1996)
16. Shearer, R., Horrocks, I.: Exploiting partial information in taxonomy construction. In: Proc. of ISWC. pp. 569–584 (2009)
17. Simančík, F., Kazakov, Y., Horrocks, I.: Consequence-based reasoning beyond Horn ontologies. In: Proc. of IJCAI. pp. 1093–1098 (2011)
18. Sirin, E., Parsia, B., Cuenca Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. J. Web Semantics 5(2), 51–53 (2007)
19. Song, W., Spencer, B., Du, W.: WSReasoner: A prototype hybrid reasoner for $\mathcal{ALCHOI}$ ontology classification using a weakening and strengthening approach. In: Proc. of the 1st Int. OWL Reasoner Evaluation Workshop (2012)
20. Tsarkov, D., Horrocks, I.: FaCT++ description logic reasoner: System description. In: Proc. of IJCAR. pp. 292–297 (2006)
21. Zhou, Y., Cuenca Grau, B., Horrocks, I.: Efficient upper bound computation of query answers in expressive description logics. In: Proc. of DL (2012)

A Canonical Model Construction

Given $F \in \mathbb{C}^{\top,\bot}$ and $\mathcal{O}_w^+$, our target is to construct a model $\mathcal{I}_{[\mathcal{O}_w^+,\prec_F]}$ for $\mathcal{O}_w^+$ satisfying $E \sqcap \neg F$ for any $E \in \mathbb{C}^{\top,\bot}$ such that $\mathcal{O}_w^+ \not\models E \sqsubseteq F$. We write $\mathcal{I}$ for $\mathcal{I}_{[\mathcal{O}_w^+,\prec_F]}$ when the parameters are not important. The construction is done by first computing a *saturation* $\mathbf{S}_{O_w^+}$ of O_w^+ and then defining a model based on it. $\mathbf{S}_{O_w^+}$ contains axioms of the forms $\mathsf{init}(H)$, $H \sqsubseteq M \sqcup A$ and $H \sqsubseteq M \sqcup \exists R.K$ derived using the inference rules.

(1) Computation of saturation
Given an $\mathcal{ALCH}$ ontology O_w^+, the saturation $\mathbf{S}_{O_w^+}$ is initialized as

$$\{\mathsf{init}(A) \mid A \in \mathbb{C}^\top\} \cup \{\mathsf{init}(N_a) \mid N_a \in \mathbb{NP}\}$$

Then $\mathbf{S}_{O_w^+}$ is expanded by iteratively applying the inference rules in Table 1 and adding the conclusions into $\mathbf{S}_{O_w^+}$ until reaching a fixpoint. An inference rule is applied by using existing axioms in $\mathbf{S}_{O_w^+}$ as premises and axioms in O_w^+ as side conditions. We write $O_w^+ \vdash \alpha$ for every $\alpha \in \mathbf{S}_{O_w^+}$ derived from O_w^+. The inference process is obviously sound, i.e. if $O_w^+ \vdash \alpha$ then $O_w^+ \models \alpha$.

(2) Model construction
We define a total order $\prec_F$ over all the concepts in O_w^+ such that F has the least order. If F is $\bot$ or $\top$, the order can be any arbitrary order. We define the domain $\triangle^{\mathcal{I}}$ of $\mathcal{I}_{[O_w^+,\prec_F]}$ as

$$\triangle^{\mathcal{I}} := \{x_H \mid \mathsf{init}(H) \in \mathbf{S}_{O_w^+} \text{ and } H \sqsubseteq \bot \notin \mathbf{S}_{O_w^+}\}$$

where x_H is an instance introduced for H. $\triangle^{\mathcal{I}}$ is nonempty because if O_w^+ is consistent, $\top \sqsubseteq \bot \notin \mathbf{S}_{O_w^+}$. Since $\mathsf{init}(\top) \in \mathbf{S}_{O_w^+}$, $x_\top$ exists.
To define the interpretation for atomic concepts, we first construct the label set $LS(x_H, \mathcal{I})$ for each instance x_H. In this section, we write $\mathcal{I}_H$ for $LS(x_H, \mathcal{I})$. Let A_i be the concept with the ith order from the smallest to the largest according to $\prec_F$. For convenience we write $M \prec_F A_i$ if for each disjunct A in M, $A \prec_F A_i$. Let $\mathcal{I}_H^i$ be a sequence where $\mathcal{I}_H^0 := \emptyset$, and $\mathcal{I}_H^i$ is defined as

$$\mathcal{I}_H^i := \begin{cases} \mathcal{I}_H^i \cup \{A_i\} \text{ if there exists } M \prec_F A_i \text{ such that} \\ \qquad O_w^+ \vdash H \sqsubseteq M \sqcup A_i \text{ and } M \cap \mathcal{I}_H^{i-1} = \emptyset \\ \mathcal{I}_H^{i-1} \quad \text{otherwise} \end{cases}$$

The last element of the sequence is defined as $LS(x_H, \mathcal{I})$. With the $LS(x_H, \mathcal{I})$ defined, the interpretation of an atomic concept A is defined as

$$A^{\mathcal{I}} := \{x_H \mid A \in LS(x_H, \mathcal{I})\}$$

The roles are interpreted to satisfy the axioms $H \sqsubseteq M \sqcup \exists R.K$. For each role R and each H such that $x_H \in \triangle^{\mathcal{I}}$, define

$$\mathcal{I}_H^R := \{K \mid \exists M : O_w^+ \vdash H \sqsubseteq M \sqcup \exists R.K, M \cap \mathcal{I}_H = \emptyset\}$$

A conjunction K is said to be maximal in $LS^R(x_H, \mathcal{I})$ if there is no $K' \in \mathcal{I}_H^R$ with a superset of conjuncts of K. Since $H \sqsubseteq \bot \notin \mathbf{S}_{O_w^+}$, by $\mathbf{R}_\exists^\bot$ rule we have $K \sqsubseteq \bot \notin \mathbf{S}_{O_w^+}$. And by $\mathbf{R}_{\mathbf{init}}$ rule we have $\mathsf{init}(K) \in \mathbf{S}_{O_w^+}$. So x_K is well-defined. The interpretation of roles is defined as

$$R^{\mathcal{I}} := \bigcup_{R' \sqsubseteq_{O_w^+} R} \{(x_H, x_K) \mid K \text{ is maximal in } \mathcal{I}_H^{R'}\}$$

The inference rules in Table 1 is modified from Table 3 in [17] by using $\mathbf{R}_{\mathbf{A}}^{+}$ and $\mathbf{R}_{\mathbf{init}}$ to initialize contexts only when necessary. The change affects only the validity of x_K in the construction for $R^{\mathcal{I}}$ which has been explained above, and the proof that $\mathcal{I}$ satisfies each type of axiom can be kept unchanged from Simancik *et. al.* [17]. So $\mathcal{I}$ is a model of the $\mathcal{ALCH}$ ontology O_w^+. Moreover, for any $E \in \mathbb{C}^{\top,\bot}$, if $O_w^+ \not\models E \sqsubseteq F$, $O_w^+ \nvdash E \sqsubseteq F$. Since F has the least order in $\prec_F$, by the definition of $LS(x_E, \mathcal{I})$ we know $x_E \notin F^{\mathcal{I}}$. Thus $x_E \in (E \sqcap \neg F)^{\mathcal{I}}$, and $\mathcal{I}_{[O_w^+, \prec_F]}$ satisfies $E \sqcap \neg F$.

B Proofs of Lemmas and Theorems

Lemma 7 *Given O_w^+ and $N_a \in \mathbb{NP}$, if (1) O_w^+ is decisive, and (2) $O_w^+ \not\models N_a \sqsubseteq \bot$. Then for any $F \in \mathbb{C}^{\top,\bot}$, N_a is a condensing label in $\mathcal{I}_{[O_w^+, \prec_F]}$.*

Proof. For simplicity we write $\mathcal{I}$ for $\mathcal{I}_{[O_w^+, \prec_F]}$ in this proof. Since $\mathsf{init}(N_a) \in \mathbf{S}_{O_w^+}$ and $O_w^+ \not\models N_a \sqsubseteq \bot$, by the construction of $\mathcal{I}$, x_{N_a} exists and $x_{N_a} \in N_a^{\mathcal{I}}$. Hence it is equivalent to show that for each $x_H \in N_a^{\mathcal{I}}$, $LS(x_H, \mathcal{I}) = LS(x_{N_a}, \mathcal{I})$.

We first show that for each H such that $x_H \in N_a^{\mathcal{I}}$, $x_{N_a} \in H^{\mathcal{I}}$. Since $x_H \in N_a^{\mathcal{I}}$, H is a potentially supporting context of N_a. Let $H = \bigsqcap_{i=1}^{n} C_H^i$, where C_H^i is A or $\neg A$. Since O_w^+ is decisive, we have the following two cases:

- $O_w^+ \models N_a \sqsubseteq C_H^i$ holds for all C_H^i, $1 \le i \le n$, then $O_w^+ \models N_a \sqsubseteq H$. Since $x_{N_a} \in N_a^{\mathcal{I}}$, $x_{N_a} \in H^{\mathcal{I}}$.
- There exists some i such that $O_w^+ \models N_a \sqcap C_H^i \sqsubseteq \bot$, then $O_w^+ \models N_a \sqcap H \sqsubseteq \bot$. By Lemma 3 of paper [17], $x_H \in H^{\mathcal{I}}$, which contradicts with our assumption $x_H \in N_a^{\mathcal{I}}$.

Next we prove $LS(x_{N_a}, \mathcal{I}) \subseteq LS(x_H, \mathcal{I})$ by contradiction. Assume $LS(x_{N_a}, \mathcal{I}) \nsubseteq LS(x_H, \mathcal{I})$, let X be the concept in $LS(x_{N_a}, \mathcal{I}) \setminus LS(x_H, \mathcal{I})$ with the smallest order. Since $X \in LS(x_{N_a}, \mathcal{I})$, there exists $N \prec_F X$ such that $O_w^+ \vdash N_a \sqsubseteq N \sqcup X$ and $N \cap LS(x_{N_a}, \mathcal{I}) = \emptyset$.

$$\because O_w^+ \vdash N_a \sqsubseteq N \sqcup X \ \ \therefore O_w^+ \models N_a \sqsubseteq N \sqcup X$$

$$\because x_H \in N_a^{\mathcal{I}} \wedge x_H \notin X^{\mathcal{I}} \ \ \therefore x_H \in N^{\mathcal{I}} \ \ \therefore LS(x_H, \mathcal{I}) \cap N \neq \emptyset$$

In the above proof, if $N = \bot$, a contradiction arises with $x_H \in N^{\mathcal{I}}$. Otherwise, let $Y \in LS(x_H, \mathcal{I}) \cap N$, there must exist $N' \prec_F Y$ s.t. $O_w^+ \vdash H \sqsubseteq N' \sqcup Y$ and $LS(x_H, \mathcal{I}) \cap N' = \emptyset$.

$$\because O_w^+ \vdash H \sqsubseteq N' \sqcup Y \text{ and } x_{N_a} \in H^{\mathcal{I}} \ \ \therefore x_{N_a} \in (N' \sqcup Y)^{\mathcal{I}}$$

$$\because N' \prec_F Y \text{ and } Y \in N \text{ and } N \prec_F X \ \therefore N' \prec_F X$$

Since X is the smallest in $LS(x_{N_a}, \mathcal{I}) \setminus LS(x_H, \mathcal{I})$, $N' \prec_F X$ and $LS(x_H, \mathcal{I}) \cap N' = \emptyset$, we have $LS(x_{N_a}, \mathcal{I}) \cap N' = \emptyset$ (it is trivially true if $N' = \bot$), and $x_{N_a} \notin N'^{\mathcal{I}}$. Given $x_{N_a} \in (N' \sqcup Y)^{\mathcal{I}}$, we have $Y \in LS(x_{N_a}, \mathcal{I})$, this contradicts with $N \cap LS(x_{N_a}, \mathcal{I}) = \emptyset$. So we conclude that $LS(x_{N_a}, \mathcal{I}) \setminus LS(x_H, \mathcal{I}) = \emptyset$ and $LS(x_{N_a}, \mathcal{I}) \subseteq LS(x_H, \mathcal{I})$.

Finally we need to prove $LS(x_H, \mathcal{I}) \subseteq LS(x_{N_a}, \mathcal{I})$. For each $X \in LS(x_H, \mathcal{I})$, there exists $N \prec_F X$ such that $O_w^+ \vdash H \sqsubseteq N \sqcup X$ and $N \cap LS(x_H, \mathcal{I}) = \emptyset$.

$$\because LS(x_{N_a}, \mathcal{I}) \subseteq LS(x_H, \mathcal{I}) \ \ \therefore N \cap LS(x_{N_a}, \mathcal{I}) = \emptyset \ \ \therefore x_{N_a} \notin N^{\mathcal{I}}$$

$$\because x_{N_a} \in H^{\mathcal{I}} \wedge x_{N_a} \notin N^{\mathcal{I}} \ \therefore x_{N_a} \in X^{\mathcal{I}}$$

Thus we conclude $X \in LS(x_{N_a}, \mathcal{I})$. □

Lemma 8 *Let $\mathcal{I}$ be a model of an $\mathcal{ALCHO}$ ontology $\mathcal{O}$ satisfying $E \sqcap \neg F$, and $E, F \in \mathbb{C}^{\top,\bot}$, where L is a condensing label in $\mathcal{I}$. Then $\mathcal{I}' = condense(L, x_L, \mathcal{I})$ is a model of $\mathcal{O} \cup \{L \equiv \{x_L\}\}$ satisfying $E \sqcap \neg F$.*

Proof. By the definition of condensing label, we have: (1) $L^{\mathcal{I}} \neq \emptyset$; (2) for all $x \in L^{\mathcal{I}}$, $LS(x, \mathcal{I})$ are the same. By (1) and the definition of r in *condense*$(L, x_L, \mathcal{I})$, we have $L^{\mathcal{I}'} = \{x_L^{\mathcal{I}'}\}$, so the axiom $L \equiv \{x_L\}$ is satisfied. By (2), we can further prove $LS(x, \mathcal{I}) = LS(r(x), \mathcal{I}')$ holds for all $x \in \triangle^{\mathcal{I}}$. Next we need to prove $\mathcal{I}' \models \alpha$ from $\mathcal{I} \models \alpha$ for any axiom α in $\mathcal{O}$. We do a case-by-case analysis for every possible form of α:

- $\alpha = \bigsqcap A_i \sqsubseteq \bigsqcup B_j$ Assume $x' \in (\bigsqcap A_i)^{\mathcal{I}'}$, there exists $x \in \triangle^{\mathcal{I}}$ s.t. $x' = r(x)$. Since $LS(x, \mathcal{I}) = LS(x', \mathcal{I}')$, we have $x \in \cap_i A_i^{\mathcal{I}}$, so $x \in \cup_j B_j^{\mathcal{I}}$. Hence $x' \in (\bigsqcup B_j)^{\mathcal{I}'}$.
- $\alpha = A \sqsubseteq \exists R.B$ Assume $x' \in A^{\mathcal{I}'}$, there exists x such that $x' = r(x)$ and $x \in A^{\mathcal{I}}$. Since $\mathcal{I} \models \alpha$, there exists $y \in \triangle^{\mathcal{I}}$ s.t. $(x, y) \in R^{\mathcal{I}}$ and $y \in B^{\mathcal{I}}$. So $(x', r(y)) \in R^{\mathcal{I}'}$ and $r(y) \in B^{\mathcal{I}'}$. Hence $x' \in (\exists R.B)^{\mathcal{I}'}$.
- $\alpha = \exists R.A \sqsubseteq B$ Assume $x' \in (\exists R.A)^{\mathcal{I}'}$, there exists y' such that $(x', y') \in R^{\mathcal{I}'}$ and $y' \in A^{\mathcal{I}'}$. So there exists $(x, y) \in R^{\mathcal{I}}$ s.t. $x' = r(x)$ and $y' = r(y)$. Since $r(y) \in A^{\mathcal{I}'}$, $y \in A^{\mathcal{I}}$. Because $\mathcal{I} \models \alpha$, $x \in B^{\mathcal{I}}$ and thus $x' \in B^{\mathcal{I}'}$.
- $\alpha = A \sqsubseteq \forall R.B$ Assume $x', y' \in \triangle^{\mathcal{I}'}$ s.t. $(x', y') \in R^{\mathcal{I}'}$ and $x' \in A^{\mathcal{I}'}$, there exists $x, y \in \triangle^{\mathcal{I}}$ s.t. $x' = r(x)$, $y' = r(y)$ and $(x, y) \in R^{\mathcal{I}}$. Since $LS(x, \mathcal{I}) = LS(x', \mathcal{I}')$, we have $x \in A^{\mathcal{I}}$. Because $\mathcal{I} \models \alpha$, $y \in B^{\mathcal{I}}$, hence $y' \in B^{\mathcal{I}'}$.
- $\alpha = N_a \equiv \{a\}$ By $\mathcal{I} \models \alpha$ we have $N_a^{\mathcal{I}} = \{a^{\mathcal{I}}\}$. According to the definition of the function *condense*() we have $N_a^{\mathcal{I}'} = \{r(a^{\mathcal{I}})\} = \{a^{\mathcal{I}'}\}$.
- $\alpha = R \sqsubseteq S$ If $(x', y') \in R^{\mathcal{I}'}$, there exists $x, y \in \triangle^{\mathcal{I}}$ s.t. $x' = r(x)$, $y' = r(y)$ and $(x, y) \in R^{\mathcal{I}}$. Since $\mathcal{I} \models \alpha$, $(x, y) \in S^{\mathcal{I}}$ and so $(x', y') \in S^{\mathcal{I}'}$.

So $\mathcal{I}' \models \mathcal{O} \cup \{L = \{x_L\}\}$ holds. Assume $x \in (E \sqcap \neg F)^{\mathcal{I}}$, since $LS(x, \mathcal{I}) = LS(r(x), \mathcal{I}')$ we know $r(x) \in (E \sqcap \neg F)^{\mathcal{I}'}$, so $(E \sqcap \neg F)^{\mathcal{I}'} \neq \emptyset$. □

Lemma 9 *Given some $\mathcal{O}_w^+$ and $F \in \mathbb{C}^{\top,\bot}$, if in $\mathcal{I}_{[\mathcal{O}_w^+, \prec_F]}$ every $N_a \in \mathbb{NP}$ is a condensing label, then for each $E \in \mathbb{C}^{\top,\bot}$ s.t. $\mathcal{O}_w^+ \not\models E \sqsubseteq F$, there is a model of $\mathcal{O}_o$ satisfying $E \sqcap \neg F$.*

Proof. Let $\{L_i \equiv \{x_{L_i}\}\}_{i=1}^n$ be all nominal axioms in $\mathcal{O}_o$. We prove $\mathcal{I}_{[\mathcal{O}_w^+, \prec_F]}$, which satisfies $E \sqcap \neg F$ for each non-subclass $E \in \mathbb{C}^{\top,\bot}$ of F, can be transformed to a model $\mathcal{I}_n$ of $\mathcal{O}^n = \mathcal{O}_w^+ \cup \{L_i \equiv \{x_{L_i}\}\}_{i=1}^n$ such that $(E \sqcap \neg F)^{\mathcal{I}_n} \neq \emptyset$ by induction on n.

By assumption for $n = 0$, $\mathcal{I}_0 = \mathcal{I}_{[\mathcal{O}_w^+, \prec_F]}$. We need to show a model $\mathcal{I}_k$ of $\mathcal{O}^k$ satisfying $E \sqcap \neg F$ can be transformed to a model $\mathcal{I}_{k+1}$ of $\mathcal{O}^{k+1}$ satisfying $E \sqcap \neg F$. This step is proved by applying Lemma 8 where $\mathcal{I} = \mathcal{I}_k$, $\mathcal{O} = \mathcal{O}^k$, $L = L_k$ and $x_L = x_{L_k}$.

Then we have transformed the model $\mathcal{I}_{[\mathcal{O}_w^+, \prec_F]}$ to a model $\mathcal{I}_n$ of $\mathcal{O}^n$ satisfying $E \sqcap \neg F$ where $\|\mathbb{NP}\| = n$. Since $\mathcal{O}^n \supseteq \mathcal{O}_o$, we have $\mathcal{I}_n \models \mathcal{O}_o$ and $(E \sqcap \neg F)^{\mathcal{I}_n} \neq \emptyset$. □

Definition 16 *In a saturation $\mathbf{S}_{\mathcal{O}_w^+}$, the **derivation path** of a conclusion α of the form $H \sqsubseteq M$ or $H \sqsubseteq N \sqcup \exists R.K$ is the sequence of all the inference steps $\mathsf{IS}_\mathsf{H}^1, \ldots, \mathsf{IS}_\mathsf{H}^\mathsf{m}$ in the context H, where: (1) $\alpha \in \mathsf{IS}_\mathsf{H}^\mathsf{m}.\mathsf{conc}$, and (2) for any $n < m$, $\mathsf{IS}_\mathsf{H}^\mathsf{n}$ occurs before $\mathsf{IS}_\mathsf{H}^{\mathsf{n+1}}$ in the saturation process.*

Lemma 17 *Given O^+_w, a concept $A \in \mathbb{C}$, and an axiom $\alpha \in \mathbf{S}_{O^+_w}$ of the form $H \sqsubseteq M \sqcup A$ or $H \sqsubseteq M \sqcup A \sqcup \exists R.K$, then there exists a conjunct B of H such that $B \in \mathsf{Pri}_{[A,O^+_w]}$.*

Proof. According to line 1 and 5 of Algorithm 2, $A \in \mathsf{Pri}_{[A,O^+_w]}$. Let $\mathsf{IS}^1_\mathsf{H}, \ldots, \mathsf{IS}^\mathsf{m}_\mathsf{H}$ be the derivation path of α. We prove the lemma by induction over m.

If $m = 1$, then IS^1_H.rule is $\mathbf{R}^+_\mathbf{A}$, and by the side condition of $\mathbf{R}^+_\mathbf{A}$, A is a conjunct of H. So the lemma holds where $B = A \in \mathsf{Pri}_{[A,O^+_w]}$. Next we show the lemma holds when $m = k$, if it holds for all $m < k$. Since $\alpha \in \mathbf{S}_{O^+_w}$, there must exist some step $\mathsf{IS}^\mathsf{p}_\mathsf{H}$ such that A is a disjunct of the axiom in the conclusion but not in the premise. In this case, $\mathsf{IS}^\mathsf{p}_\mathsf{H}$.rule can only be $\mathbf{R}^+_\mathbf{A}$, $\mathbf{R}^\mathbf{n}_\sqcap$ or $\mathbf{R}^-_\exists$, so we can perform a case analysis as follows.

Case 1 Similarly to the case $m = 1$, we can choose $B = A$ to prove the lemma.

Case 2 $\mathsf{IS}^\mathsf{p}_\mathsf{H}$.rule=$\mathbf{R}^\mathbf{n}_\sqcap$ In this case $\mathsf{IS}^\mathsf{p}_\mathsf{H}$.sc has a single axiom α of the form $\bigsqcap A_i \sqsubseteq \bigsqcup B_j$. By line 7 there exists some A_i s.t. $A_i \in \mathsf{Pri}_{[A,O^+_w]}$. Since $H \sqsubseteq N_i \sqcup A_i \in \mathsf{IS}^\mathsf{p}_\mathsf{H}$.prem, its derivation path $\mathsf{IS}^1_\mathsf{H}, \ldots, \mathsf{IS}^{\mathsf{p}'}_\mathsf{H}$ must satisfy $p' < p \leq k$. By applying the inductive hypothesis to $m = p'$ and $H \sqsubseteq N_i \sqcup A_i$, there exists H's conjunct B s.t. $B \in \mathsf{Pri}_{[A_i,O^+_w]}$. Since $A_i \in \mathsf{Pri}_{[A,O^+_w]}$, according to the Algorithm 2, we can see that $\mathsf{Pri}_{[A_i,O^+_w]} \subseteq \mathsf{Pri}_{[A,O^+_w]}$. So $B \in \mathsf{Pri}_{[A,O^+_w]}$, and the lemma is proved.

Case 3 $\mathsf{IS}^\mathsf{p}_\mathsf{H}$.rule=$\mathbf{R}^-_\exists$ In this case $\mathsf{IS}^\mathsf{p}_\mathsf{H}$.sc has axioms of the forms $R \sqsubseteq^*_O S$ and $\exists S.Y \sqsubseteq A$, and one of the premises $\mathsf{IS}^\mathsf{p}_\mathsf{H}$.prem is of the form $H \sqsubseteq M' \sqcup \exists R.(\bigsqcap^n_{i=1} C^i_{K'})$, which is derived by the process:

$$H \sqsubseteq M_1 \sqcup A' \xrightarrow[A' \sqsubseteq \exists R.C^1_{K'}]{\mathbf{R}^+_\exists} H \sqsubseteq M_1 \sqcup \exists R.C^1_{K'} \cdots \xrightarrow[R \sqsubseteq^*_O S \;\; Y \sqsubseteq \forall S.Z / \exists S.Z \sqsubseteq Y]{\mathbf{R}_\forall / \mathbf{R}^-_\exists} H \sqsubseteq M' \sqcup \exists R.(\bigsqcap_{i=1}^{n} C^i_{K'})$$

The first inference step has a side condition of the form $A' \sqsubseteq \exists R.C^1_{K'}$ and a premise of the form $H \sqsubseteq M_1 \sqcup A'$. By line 8 to 9, A' is added to $\mathsf{Pri}_{[A,O^+_w]}$ where $W = A$ and $Z = C^1_{K'}$. Let $\mathsf{IS}^1_\mathsf{H}, \ldots, \mathsf{IS}^{\mathsf{p}'}_\mathsf{H}$ be the derivation path of $H \sqsubseteq M_1 \sqcup A'$. We can see $p' < k$ since $H \sqsubseteq M_1 \sqcup A'$ must be derived before the kth step. By the inductive hypothesis, there exists H's conjunct B s.t. $B \in \mathsf{Pri}_{[A',O^+_w]}$. Since $A' \in \mathsf{Pri}_{[A,O^+_w]}$, $\mathsf{Pri}_{[A',O^+_w]} \subseteq \mathsf{Pri}_{[A,O^+_w]}$. So $B \in \mathsf{Pri}_{[A,O^+_w]}$, and the lemma is proved. □

Lemma 13 *Given O^+_w and a concept A, $\mathsf{OPS}_{[A,O^+_w]}$ returned by Algorithm 2 preserves $\mathsf{OPS}_{[A,O^+_w]} \supseteq \mathsf{PS}_{[A,O^+_w]}$.*

Proof. By Lemma 17, we have shown that there is at least one conjunct B of H in $\mathsf{Pri}_{[A,O^+_w]}$. Since H's conjuncts are all collected during the derivation of $\mathsf{init}(H)$, we discuss the two cases how $\mathsf{init}(H)$ is derived, and how H's conjuncts are added in each case:

- If $\mathsf{init}(H)$ is introduced at initialization stage, then B is the only conjunct in H belonging to $\mathbb{C}^{\top,\bot}$ or $\mathbb{NP}$, and is added to $\mathsf{OPS}_{[A,O^+_w]}$ in line 11 of Algorithm 2 where $W = B$ and $U = \mathbb{C}^{\top,\bot}$.
- If $\mathsf{init}(H)$ is introduced by $\mathbf{R_{init}}$ rule, then there is a premise of the form $H^* \sqsubseteq M \sqcup \exists R.H$, where H^* is a context different from H. Let $H = \bigsqcap^n_{i=1} C^i_H$, the derivation process of $H^* \sqsubseteq M \sqcup \exists R.H$ is:

$$H^* \sqsubseteq M_1 \sqcup A \xrightarrow[A \sqsubseteq \exists R.C^1_H]{\mathbf{R}^+_\exists} H^* \sqsubseteq M_1 \sqcup \exists R.C^1_H \cdots \xrightarrow[R \sqsubseteq^*_O S \;\; Y \sqsubseteq \forall S.Z / \exists S.Z \sqsubseteq Y]{\mathbf{R}_\forall / \mathbf{R}^-_\exists} H^* \sqsubseteq M \sqcup \exists R.(\bigsqcap_{i=1}^{n} C^i_H)$$

The side condition of the first step is $A \sqsubseteq \exists R.C^1_H$. We first prove $\exists R.C^1_H \in \mathsf{Exists}$, where Exists is the set produced in the loop from lines 10 to 13. If B is C^1_H, then $\exists R.C^1_H$ is added to Exists in line 13 where $W = B$. If B is a conjunct of H other than C^1_H, then B becomes a conjunct after an application of $\mathbf{R}_\forall$ rule, in such case the side condition is $R \sqsubseteq^*_O S$ and $Y \sqsubseteq \forall S.B$, so $\exists R.C^1_H$ is added to Exists in line 12 where $W = B$.

Next we show the lemma holds for all three types of conjuncts C of H:

1. If C is added to the conjuncts of H by $\mathbf{R}^+_\exists$ rule, then $C = C^1_H$ and is added to $\mathsf{OPS}_{[A,O^+_w]}$ in line 15.
2. If C is added to the conjuncts of H by $\mathbf{R}_\forall$ rule, then C is added to $\mathsf{OPS}_{[A,O^+_w]}$ in line 16.
3. If C is added to the conjuncts of H by $\mathbf{R}^-_\exists$ rule, then C is of the form $\neg Z$, and Z is added to $\mathsf{OPS}_{[A,O^+_w]}$ in line 17.

Hence the lemma is proved. □

Theorem 14 *Let O^+ be strengthening axioms computed from Algorithm 3, the ontology $O^+_w = O_w \cup O^+$ is decisive.*

Proof. Since in the last round of the loop $O^{\mathsf{n}+} = O^{\mathsf{n}-1+}$, we know that for each $N_a \in \mathbb{NP}$ and $X \in \mathsf{OPS}_{[N_a,O^{\mathsf{n}+}_w]} = \mathsf{OPS}_{[N_a,O^{\mathsf{n}-1+}_w]}$, $\mathsf{chooseStrAxiom}(N_a, X) \subseteq O^{\mathsf{n}+} \subseteq O^{\mathsf{n}+}_w$. And because $\mathsf{PS}_{[N_a,O^{\mathsf{n}+}_w]} \subseteq \mathsf{OPS}_{[N_a,O^{\mathsf{n}+}_w]}$, thus $O^+_w = O^{\mathsf{n}+}_w$ is decisive.

Theorem 15 *Let O^+ be strengthening axioms computed from Algorithm 4, the ontology $O^+_w = O_w \cup O^+$ is decisive.*

Proof. For each $N_a \in \mathbb{NP}$, we write g_{N_a} for the final group that N_a belongs to after executing lines 6 to 7 of Algorithm 4. We will prove that $\mathsf{OPS}_{[N_a,O^+_w]} \subseteq g_{N_a}.ops$. From lines 10 to 17 of Algorithm 2, we can see that each concept X is added to $\mathsf{OPS}_{[N_a,O^+_w]}$ because of some $W \in \mathsf{Pri}_{[N_a,O^+_w]}$ in line 10, for which we write W_X. And according to the loop in lines 2 to 9, W_X is added to $\mathsf{Pri}_{[N_a,O^+_w]}$ through a search path $N_a = W_0 \rightarrow W_1 \rightarrow \ldots W_n = W_X$, where $W_{i-1} \rightarrow W_i$, $1 \le i \le n$ represents W_i is added to $\mathsf{ToProcess}$ while processing W_{i-1} in the loop where $W = W_{i-1}$. Note that in Algorithm 2, the only case that a strengthening axiom $\alpha \in O^+$ is used is when α is of the form $N_b \sqsubseteq X'$ and used in line 7. Let W^s be the set of W_i such that W_i is added into $\mathsf{ToProcess}$ from $W = W_{i-1}$ in line 7 using such a strengthening axiom $\alpha = N_b \sqsubseteq X'$. We prove the lemma by induction over the size $m = \|W^s\|$ of W^s. The inductive hypothesis is:

> For each $X \in \mathsf{OPS}_{[N_a,O^+_w]}$ and $N_a \in \mathbb{NP}$, let $W_X \in \mathsf{Pri}_{[N_a,O^+_w]}$ be the concept that causes X to be added into $\mathsf{OPS}_{[N_a,O^+_w]}$, $N_a = W_0 \rightarrow W_1 \rightarrow \ldots W_n = W_X$ be the search path of W_X, and $W^{s'}$ be the set of W_i such that $W_{i-1} \rightarrow W_i$ uses a strengthening axiom. If $\|W^{s'}\| < m$ then $X \in g_{N_a}.ops$.

– $m = 0$ In this case $W_X \in \mathsf{Pri}_{[N_a,O_w]}$ and $X \in \mathsf{OPS}_{[N_a,O_w]}$. By line 4 we see X is in the initial group of N_a, and is merged into $g_{N_a}.ops$ while executing lines 6 to 7.

– $m = k$ Let $i_m = \min\{i \mid W_i \in W^s\}$. In this case we have $W_{i_m - 1} = X'$ and $W_{i_m} = N_b$. It is easy to see that X is also added to $\mathsf{OPS}_{[N_b, O_w^+]}$ through the same W_X and a search path $N_b = W_{i_m} \rightarrow \ldots W_n = W_X$, whose $\|W^{s'}\|$ is $k - 1$. By applying the inductive hypothesis to X and N_b, we know $X \in g_{N_b}.ops$. And since $X' \in \mathsf{Pri}_{[N_a, O_w]}$, $X' \in g_{N_a}.pri$. So $g_{N_a}.pri \cap g_{N_b}.ops \neq \emptyset$, and g_{N_a} and g_{N_b} must be the same group according to the merge criteria in line 6 of Algorithm 4. So $X \in g_{N_b}.ops = g_{N_a}.ops$.

Hence $\mathsf{PS}_{[N_a, O_w^+]} \subseteq \mathsf{OPS}_{[N_a, O_w^+]} \subseteq g_{N_a}.ops$ for each $N_a \in \mathbb{NP}$ in O_w^+. According to the construction of O^+ in line 11 of Algorithm 4, we conclude $O_w^+ = O_w \cup O^+$ is decisive.

C Evaluation Results

We have implemented our prototype hybrid reasoner WSClassifier in Java using OWL API. The reasoner uses ConDOR r.12 as the main $\mathcal{ALCH}$ reasoner and HermiT 1.3.6 as the assistant reasoner for DL $\mathcal{ALCHO}$. WSClassifier adopts a well-known preprocessing step to eliminate transitive roles [10], hence supports DL $\mathcal{SHO}$ ($\mathcal{ALCHO}$+ transitivity axioms). We compared the classification time of WSClassifier with tableau-based reasoners HermiT 1.3.6, Fact++ 1.5.3 and Pellet 2.3.0, as well as another hybrid reasoner MORe 0.1.3 which combines ELK and HermiT. All the experiments were run on a laptop with an Intel Core i7-2670QM 2.20GHz quad core CPU and 16GB RAM running Java 1.6 under Windows 7. We set the Java heap space to 12GB and the time limit to 9 days for all reasoners. For HermiT, we set its configuration to simple core blocking and individual reuse which is optimized configuration for running the large and complex ontologies.

We evaluated WSClassifier and other reasoners on all large and complex ontologies available to us, on the ORE dataset and on some proposed variants. The only large and complex ontologies included are FMA-constitutionalPartForNS(FMA-C)[1] and modified versions of Galen in which some concepts starting with a lower case letter and subsumed by *SymbolicValueType* are modeled as nominals. The ontologies containing "EL" in the name are constructed based on Galen-EL[2]. Galen-EL-n1Y and Galen-EL-n2Y were provided [12]. Galen-Heart-n1 and Galen-Heart-n2 are subontologies, respectively. Galen-EL-n1YE and Galen-EL-n2YE have some nominals removed and Galen-Union-n is made by adding disjunctions of nominals. We used two common s-maller complex ontologies – Wine and DOLCE. We use the ORE dataset,[3] where 2 ontologies without axioms are removed. In all cases, we reduce the language to $\mathcal{SHO}$. The ontologies are available from our website.[4].

For FMA-C which is the only real-world large and complex ontologies with nominals we have access to, WSClassifier finished classification in 21.2 seconds, while HermiT used 140,882 seconds. Other reasoners did not finish classification on it in 9 days. From the results of Table 2 we can see, excluding ORE dataset, WSClassifier is significantly faster than the tableau-based reasoners on 7 out of 10 ontologies. For the

[1] Foundational Model of Anatomy, http://sig.biostr.washington.edu/projects/fm/index.html
[2] http://code.google.com/p/condor-reasoner/downloads/list
[3] http://www.cs.ox.ac.uk/isg/conferences/ORE2012/
[4] http://isel.cs.unb.ca/~wsong/WSClassifierExperimentOntologies.zip

Table 2. Comparison of classification performance

Ontology	Concepts	Nominals	(Hyper) tableau			Hybrid	
			HermiT	**Pellet**	**FaCT++**	**MORe**	**WSClassifier**
Wine	146	206	24.6	285.6	4.6	1.0	28.7
DOLCE	207	39	6.6	7.0	15.6	53.3	1.3
Galen-Heart-n1	3366	55	264.0	–	–	337,505	4.1
Galen-Heart-n2	3366	92	768.4	–	–	338,453	1.8
Galen-EL-n1Y	23136	739	701,822.0	–	–	–	700,985.0
Galen-EL-n2Y	23136	1113	407,427.0	–	–	–	408,188.0
Galen-EL-n1YE	23136	598	244,146.0	–	–	–	17.0
Galen-EL-n2YE	23136	712	289,637.0	–	–	–	25,630.0
Galen-Union-n	23136	598	469,274.3	–	–	–	21.1
FMA-C	41648	85	140,882.0	–	–	–	21.2
ORE-dataset (OWL DL & EL,113 ontologies) the following refers to average number							
FMA-lite	75,141	0	137,409.0	–	–	–	26.0
remaining 112 ontologies	4293	343	0.84	0.86	–*	0.24	2.10

Note: The time is measured in seconds. "–" means out of time or memory

*: Fact++ terminates unexpectedly while classifying some ontologies in the ORE-dataset

other 3 of 10 ontologies – Wine, Galen-EL-YN1 and Galen-EL-YN2, WSClassifier, incurring a relatively small cost, detected that strengthening axioms made some concepts unsatisfiable in $\mathcal{O}_s$, and so failed over to HermiT.

We see a major speedup for WSClassifier on ORE's FMA-lite. On the remaining 112 ORE ontologies, our average reasoning time is longer than other reasoners. Among these ontologies, 51 have nominals, mostly coming from ABoxes, and only 9 of them have strengthening axioms. Of the 9 ontologies, 8 did not produce any new subsumptions in $\mathcal{H}_s$ and only 1 which is the variant of Wine ontology introduced new unsatisfiable concepts and fails over to HermiT. Thus the WS approach does not incur much additional work, and most of the additional time is taken on overheads: computing normalized and strengthening axioms, and transmitting the ontology to and from ConDOR, which is necessary since ConDOR cannot be accessed directly through OWL API and consumes about 60% of the time.

WSClassifier outperforms MORe on DOLCE and all the Galen ontologies. For the Galen ontologies, MORe assigns all the classification work to a default configured HermiT; fine-tuning may improve its times. However, MORe computes only subsumptions implied by the TBox, ignoring the ABox, thus its classification result is incomplete for some ontologies with ABoxes, such as Wine.

WSClassifier seems most applicable when the ontologies are large and highly cyclic since then tableau reasoners construct large models and employ expensive blocking strategies. On the other hand consequence-based reasoners do not encounter problems on highly cyclic ontologies, and so can classify even cyclic $\mathcal{O}_w$ and $\mathcal{O}_s$ quickly. If there are no or just a few additional subsumptions derived by $\mathcal{O}_s$, AR does not need or just do a little work on the highly cyclic O_o. This improvement is observed for FMA-C which is the only real-world large and complex ontology with nominals we have access to.