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

by

Weihong Song, Bruce Spencer and Weichang Du

TR 13-222, January, 2013

Faculty of Computer Science
University of New Brunswick
Fredericton, N.B. E3B 5A3
Canada

Phone: (506) 453-4566
Fax: (506) 453-3566
E-mail: fcs@unb.ca
http://www.cs.unb.ca

Abstract

Consequence-based reasoners, which are applicable for classifying ontologies in less expressive DL languages such as $\mathcal{ALCH}$, are typically significantly faster than tableau-based reasoners. 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. In this paper, we present a weakening and strengthening approach for classification of $\mathcal{ALCHO}$ ontologies, using a hybrid of consequence- and tableau-based reasoning, with the intention of giving most of the work to the faster consequence-based reasoner. In our approach, given an $\mathcal{ALCHO}$ ontology O_{in}, we first weaken it to an $\mathcal{ALCH}$ ontology O_{wk} by removing nominal axioms. We then approximate the effect of the removed axioms to retain classification completeness by adding some $\mathcal{ALCH}$ axioms, which we call strengthening axioms, creating the O_{str} ontology, also in $\mathcal{ALCH}$. In the classification process, we first use a consequence based main reasoner to classify O_{wk}, which produces a sound but possibly incomplete subset of classification for O_{in}. We then use the same consequence based reasoner to classify O_{str}, which generates a complete but possibly not sound classification of O_{in}. Unsound subsumptions from the second-round classification are filtered out by a tableau-based assistant reasoner.

WSClassifier often takes less time than tableau-based reasoners for large and complex ontologies, but when it does not, it quickly provides an indication of the time it will require, allowing the tableau-reasoner to take over.

1 Introduction

Ontology classification is a fundamental reasoning service used in the development and application of OWL ontologies [7]. The target of classification is to calculate all the subsumption relationships between atomic concepts implied by the input ontology. (Hyper)Tableau-based and consequence-based reasoners are two of the mainstream reasoners for ontology classification. (Hyper)Tableau-based reasoners [10, 14] build models to test the satisfiability of concepts of the form $A \sqcap \neg B$ in order to see whether $A \sqsubseteq B$ holds. Current (hyper)tableau-based reasoners such as HermiT [14], FaCT++ [22], Pellet [19] and RacerPro [9], 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 classify ontologies by computing the saturation from the specifically designed inference rules that produce implied subsumptions [11, 12, 18]. They are variations of so-called completion-based approaches proposed for the OWL EL family [3, 5]. They are typically very fast but support less expressive DLs. So far the most expressive languages that are supported by consequence-based reasoners are Horn-$\mathcal{SHIQ}$ [11] and $\mathcal{ALCH}$ [18].

In this paper we introduce a hybrid reasoning approach for classification of ontologoes in the DL L, using a consequence-based *main reasoner* MR and a tableau-based *assistant reasoner* AR. MR provides sound and complete classification over the DL L_b which is less expressive than L, while AR provides sound and complete classification over L. Suppose MR reasoning is much faster than AR. We try to classify an ontology O_{in} using MR to do the major work, and AR to do auxiliary work. We produce a weakened version O_{wk} by removing from O_{in} the axioms that are beyond L_b, and a strengthened version O_{str} by adding to O_{wk} a set of strengthening axioms O_N^+ in L_b that compensate for the removed axioms. O_{str} to O_{wk} are in L_b and are classified by MR producing $\mathcal{H}_{wk}$ and $\mathcal{H}_{str}$, respectively. Subsumptions in $\mathcal{H}_{wk}$ are sound but may not be complete w.r.t , whereas subsumptions in $\mathcal{H}_{str}$ are complete but may not be sound. Unsound subsumptions in $\mathcal{H}_{str} \setminus \mathcal{H}_{wk}$ are detected by AR and filtered out. Those that remain are added to $\mathcal{H}_{wk}$ resulting in the sound and complete classification of O_{in}. We call this approach weakening and strengthening (WS); it is based on theory approximation [16]. We have implemented a prototype reasoner WSClassifier for $L = \mathcal{ALCHO}$ and $L_b = \mathcal{ALCH}$. Our empirical results of applying the WS approach to $\mathcal{ALCHOI}$ were reported previously [20].

We first introduce preliminaries in Section 2, then give an overview of our hybrid classification procedure for $\mathcal{ALCHO}$ ontologies and prove its completeness in Section 3. In Section 4, we introduce a polynomial time search to compute a set of strengthening axioms efficiently. In Section 5 we show experimental results and report the amount of extra work to do this verification. In Section 6 we introduce the related work. Finally, in Section 7, we give the concluding remarks.

2 Preliminaries

The syntax of $\mathcal{ALCHO}$ uses sets of atomic concepts, atomic roles and individuals. We use A, B, E, F for atomic concepts, C, D for concepts, R, S for roles and a, b for individuals. Complex concepts are either $\{a\}$ or recursively defined using constructors $\neg, \sqcap, \sqcup, \exists, \forall$. An ontology is a finite set of axioms in the form of concept subsumptions $C \sqsubseteq D$ or role subsumptions $R \sqsubseteq S$. A concept equivalence $C \equiv D$ is a shortcut of two subsumptions. Given an ontology $\mathcal{O}$, $\sqsubseteq^*_{\mathcal{O}}$ is the smallest reflexive transitive binary relation over roles such that $R \sqsubseteq S \in \mathcal{O}$ implies $R \sqsubseteq^*_{\mathcal{O}} S$. [1]

The semantics of $\mathcal{ALCHO}$ is defined using interpretations. An interpretation is a pair $\mathcal{I} = (\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\triangle^{\mathcal{I}}$ is a non-empty set called the domain of the interpretation and $\cdot^{\mathcal{I}}$ is the interpretation function, which assigns to each atomic concept A a set $A^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}}$, to each atomic role R a relation $R^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$, and to each individual a an element $a^{\mathcal{I}} \in \triangle^{\mathcal{I}}$. The interpretation of all the constructors are defined in Table 1. An interpretation $\mathcal{I}$ satisfies an axiom α (written as $\mathcal{I} \models \alpha$) if the respective semantic definition of the axiom in Table 1 holds. $\mathcal{I}$ is a model of an ontology $\mathcal{O}$ (written $\mathcal{I} \models \mathcal{O}$) if $\mathcal{I}$ satisfies each axiom in $\mathcal{O}$. An axiom α is implied by an ontology $\mathcal{O}$(written as $\mathcal{O} \models \alpha$) if all models of $\mathcal{O}$ satisfy α. We say an interpretation $\mathcal{I}$ satisfies a concept C if $C^{\mathcal{I}} \neq \emptyset$.

The use of nominals provides sufficient expressivity for other common axioms, e.g. concept assertions $C(a)$ can be written as $\{a\} \sqsubseteq C$, role assertions $R(a, b)$ can be written as $\{a\} \sqsubseteq \exists R.\{b\}$, and individual equivalence $a \equiv b$ can be written as $\{a\} \equiv \{b\}$.

The notations $\bigsqcap_{i=1}^{n} C_i$ and $\bigsqcup_{i=1}^{n} C_i$ denotes finite n-ary conjunctions and disjunctions with usual semantics. Ranges are omitted when irrelevant. We use letters H, K for conjunctions of atomic concepts, and M, N, M', N' for disjunctions. Conjunctions or disjunctions with different orders or multiplicity of elements are not distinguished, so we treat them as sets and use set-theoretic operators $\in, \subseteq, \cap$ on them. Empty conjunction and disjunction are identified as $\top$ and $\bot$, respectively.

3 Hybrid Classification of Ontologies

Algorithm 1 describes our hybrid procedure for classifying $\mathcal{O}_{in}$. 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 $\mathcal{O}_{wk}$ and $\mathcal{O}_{str}$ are generated and classified using the MR; and (3) a *verification stage* (lines 9 to 17) in which the subsumptions arising from just the $\mathcal{O}_{str}$ are verified using AR. The notation $\mathbb{C}$ denotes the set of atomic concepts in $\mathcal{O}_{in}$ *before normalization*, and $\mathbb{C}^{\top,\bot} = \mathbb{C} \cup \{\top, \bot\}$. $\mathcal{H}_{wk}$, $\mathcal{H}_{in}$ and $\mathcal{H}_{str}$ are classification results of $\mathcal{O}_{wk}$, $\mathcal{O}_{in}$ and $\mathcal{O}_{str}$ expressed as a set of subsumptions $A \sqsubseteq B$, where $A, B \in \mathbb{C}^{\top,\bot}$.

In the normalization stage, the $\mathcal{ALCHO}$ ontology $\mathcal{O}_{in}$ 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 $N_a \equiv \{a\}$. We assume all the ontologies are normalized in the remaining of the paper. This

[1] $\sqsubseteq^*_{\mathcal{O}}$ may not be a complete hierarchy of roles in an $\mathcal{ALCHO}$ ontology $\mathcal{O}$, but it does not affect the completeness of our approach

Table 1: Syntax and semantics of $\mathcal{ALCHO}$

Constructor	**Syntax**	**Semantics**
Concepts:		
atomic concept	A	$A^{\mathcal{I}}$
top	$\top$	$\triangle^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
negation	$\neg C$	$\triangle^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
existential restriction	$\exists R.C$	$\{x \mid R^{\mathcal{I}}(x) \cap C^{\mathcal{I}} \neq \emptyset\}$
universal restriction	$\forall R.C$	$\{x \mid R^{\mathcal{I}}(x) \subseteq C^{\mathcal{I}}\}$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
Roles:		
atomic role	R	$R^{\mathcal{I}}$
Individuals:		
individual name	a	$a^{\mathcal{I}}$
Axioms:		
concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
role inclusion	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$

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

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

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

transformation preserves subsumptions in $\mathcal{O}_{in}$ (see [21]). We call $N_a \equiv \{a\}$ an nominal axiom in the following text, and write $\mathbb{NP}$ for the set $\{N_a \mid N_a \equiv \{a\}\}$ in $\mathcal{O}_{in}$.

In the verification stage, there are some cases we hand over the classification work to AR: (1) $N_a \sqsubseteq \bot \in \mathcal{H}_{str}$; (2) $\mathcal{O}_{str}$ has a unsatisfiable concept A but $\mathcal{O}_{in} \not\models A \sqsubseteq \bot$, so that $A \sqsubseteq B \in \mathcal{H}_{str}$ for all $B \in \mathbb{C}^{\top,\bot}$, and most of them are not in $\mathcal{H}_{wk}$ and need to be checked; (3) the fraction $\parallel \mathcal{H}_{str} \setminus \mathcal{H}_{wk} \parallel / \parallel \mathbb{C} \parallel$ is greater than a threshold d. In the latter two cases, the estimated work for the stage is more than using AR to classify $\mathcal{O}_{in}$. For (3) we set $d = 1.5$ in our implementation based on the experiments in [8].

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

$\mathcal{O}_{str}$ is obtained from $\mathcal{O}_{wk}$ by adding $\mathcal{O}_N^+$, which is a set of strengthening axioms. Every model $\mathcal{I}$ of $\mathcal{O}_{str}$ satisfies all axioms in $\mathcal{O}_{in}$ 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 $\mathcal{O}_{str}$, $N_a^{\mathcal{I}}$ could have zero or multiple instances. However, if for each N_a, $N_a^{\mathcal{I}} \neq \emptyset$ and all the instances in $N_a^{\mathcal{I}}$ are "identical", i.e., they have the same label sets, these instances can be replaced by a single instance. Such a replacement is called a *condensation* – it condenses all of the differnt 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 axioms, then we can create a model for $\mathcal{O}_{in}$.

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 algorithms 2 and 3, 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, so that the condensations can occur.

We use a variant of canonical model construction approach for $\mathcal{ALCH}$ ontologies [18], which will be explained in the following subsection. Given a pair $E, F \in \mathbb{C}^{\top,\bot}$ suppose a model of O_{str} exists in which $E \sqcap \neg F$. We shall show that a canonical model of O_{str} also exists that satifies $E \sqcap \neg F$ and that this canonical model can be condensed to a model of O_{in} satisfying $E \sqcap \neg F$. Stating the contrapositive: if $O_{in} \models E \sqsubseteq F$ then $O_{str} \models E \sqsubseteq F$. Thus we know reasoning in O_{str} is complete, once we create the canonical models, done in Section 3.1, and show they can be condensed, done in Section 3.2.

3.1 Canonical Model Construction

Here we introduce the construction of canonical models. Given $E, F \in \mathbb{C}^{\top,\bot}$ such that $O_{str} \not\models E \sqsubseteq F$, a canonical model $\mathcal{I}(E, F)$ of O_{str} is constructed by first computing a *saturation* $S(E)$ of O_{str} and then defining a model based on it. $S(E)$ 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 $E \in \mathbb{C}^{\top}$ and O_{str}, the saturation $S(E)$ is initialized as

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

Then $S(E)$ is expanded by iteratively applying the inference rules in Table 2 and adding the conclusions into $S(E)$ until reaching a fixpoint. Existing axioms in $S(E)$ are used as premises and axioms in O_{str} are used as side conditions. We write $O_{str} \vdash_E \alpha$ for every α in $S(E)$ derived from O_{str}. The inference process is obviously sound, i.e. if $O_{str} \vdash_E \alpha$ then $O_{str} \models \alpha$.

(2) Definition of $\mathcal{I}(E, F)$

If $E \sqsubseteq F$ or $E \sqsubseteq \bot$ occurs in $S(E)$, then $O_{str} \models E \sqsubseteq F$ and the model satisfying $E \sqcap \neg F$ does not exist. Otherwise we define a total order $\prec_F$ over all the concepts in O_{str} such that F has the least order. If F is $\bot$, the order can be an arbitrary order. We define the domain $\triangle^{\mathcal{I}}$ of $\mathcal{I}(E, F)$ as

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

where x_H is an instance introduced for H. $\triangle^{\mathcal{I}}$ is nonempty because $\mathsf{init}(E) \in S(E)$ and $E \sqsubseteq \bot \notin S(E)$, and so x_E exists.

To define the interpretation for atomic concepts, we first construct the label set $LS(x_H, \mathcal{I})$ for each instance x_H. For simplicity, we write LS_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

Table 2: Complete Inference Rules for Normalized $\mathcal{ALCH}$ ontologies

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

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

$$\mathbf{R}^-_\exists\ \frac{H \sqsubseteq M \sqcup \exists R.K \quad K \sqsubseteq N \sqcup A}{H \sqsubseteq M \sqcup B \sqcup \exists R.(K \sqcap \neg A)} : \begin{array}{l}\exists S.A \sqsubseteq B \in O_{str}\\ R \sqsubseteq^*_O S\end{array} \qquad \mathbf{R}^\bot_\exists\ \frac{H \sqsubseteq M \sqcup \exists R.K \quad K \sqsubseteq \bot}{H \sqsubseteq M}$$

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

$\prec_F$. For convenience we write $M \prec_F A_i$ if for each disjunct A in M, $A \prec_F A_i$. Let LS^i_H be a sequence where $LS^0_H := \emptyset$, and LS^i_H is defined as

$$LS^i_H := \begin{cases} LS^i_H \cup \{A_i\} \text{ if there exists } M \prec_F A_i \text{ such that} \\ \qquad O_{str} \vdash H \sqsubseteq M \sqcup A_i \text{ and } M \cap LS^{i-1}_H = \emptyset \\ LS^{i-1}_H \text{ otherwise} \end{cases}$$

The last element of the sequence is defined as LS_H. With the LS_H defined, the interpretation of an atomic concept A is defined as

$$A^\mathcal{I} := \{x_H \mid A \in LS_H\}$$

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

$$LS^R_H := \{K \mid \exists M : O_{str} \vdash H \sqsubseteq M \sqcup \exists R.K, M \cap LS_H = \emptyset\}$$

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

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

The inference rules in Table 2 is modified from Table 3 in [18] by using $\mathbf{R}^+_\mathbf{A}$ and $\mathbf{R_{init}}$ 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 previous work [18]. So $\mathcal{I}$ is a model of the $\mathcal{ALCH}$ ontology O_{str}.

3.2 Condensing Labels and Completeness

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 $LS(x, \mathcal{I})$.*

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 an 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 *If there is a concept $E \in \mathbb{C}^{\top}$ such that $H \sqsubseteq M \sqcup N_a \in S(E)$, then H is an* ***occurring context*** *of N_a. We write H_{N_a} for all* ***occurring contexts*** *of N_a.*

Definition 5 *A concept X is called a major coexisting label of a concept Y in $\mathcal{O}_{str}$ if either X or $\neg X$ is a conjunct of an occurring context H of Y in $\mathcal{O}_{str}$. We denote the set of all* ***major coexisting labels*** *of Y as $MC(Y)$.*

X is called an atom of H if X or $\neg X$ is a conjunct of H. An instance's label set does not include the atom of any negative literal it is interpreted to belong to. However a concept's MC label may include the atom of a negative literal.

Lemma 6 *If for each $X \in \mathsf{MC}_{\mathsf{N_a}}$, either $N_a \sqcap X \sqsubseteq \bot$ or $N_a \sqsubseteq X$ holds, then for each occurring context H of N_a, either $N_a \sqcap H \sqsubseteq \bot$ or $N_a \sqsubseteq H$ holds.*

Proof. Assume $H = \sqcap_i^n X_i$. (1) If $N_a \sqsubseteq X_i$ for each X_i, $1 \leq i \leq n$, then $N_a \sqsubseteq \sqcap_i^n X_i \sqsubseteq H$. (2) If there exists some X_i, $1 \leq i \leq n$ s.t. $N_a \sqcap X_i \sqsubseteq \bot$, then $N_a \sqcap H \sqsubseteq N_a \sqcap X_i \sqsubseteq \bot$. □

Lemma 7 *Given the ontology $\mathcal{O}_{str}$, $E \in \mathbb{C}^{\top}$ and $N_a \in \mathbb{NP}$, if (1) N_a is satisfiable and (2) for each H s.t. $H \sqsubseteq M \sqcup N_a \in S(E)$, either $\mathcal{O}_{str} \models N_a \sqsubseteq H$ or $\mathcal{O}_{str} \models N_a \sqcap H \sqsubseteq \bot$ holds, then for any $F \in \mathbb{C}^{\top,\bot}$ such that $\mathcal{O}_{str} \not\models E \sqsubseteq F$, N_a is a condensing label in $\mathcal{I}(E, F)$.*

Proof. Since N_a is satisfiable, and $\mathsf{init}(N_a) \in S(E)$, so x_{N_a} exists, and according to the model construction $x_{N_a} \in N_a^{\mathcal{I}}$. So we need to prove that for each $x_H \in N_a^{\mathcal{I}}$, $LS_H = LS_{N_a}$ in any model $\mathcal{I}(E, F)$. Since $x_H \in N_a^{\mathcal{I}}$, we know $\mathcal{O}_{str} \not\models N_a \sqcap H \sqsubseteq \bot$ so that the choice from (2) is $\mathcal{O}_{str} \models N_a \sqsubseteq H$, and $x_{N_a} \in H^{\mathcal{I}}$.

We first prove $LS_{N_a} \subseteq LS_H$ by contradiction. Assume $LS_{N_a} \setminus LS_H \neq \emptyset$, let X be the concept in $LS_{N_a} \setminus LS_H$ with the smallest order. Since $X \in LS_{N_a}$, there exists $N \prec_F X$ such that $\mathcal{O}_{str} \vdash N_a \sqsubseteq N \sqcup X$ and $N \cap LS_{N_a} = \emptyset$.

$$\because \mathcal{O}_{str} \vdash N_a \sqsubseteq N \sqcup X \;\; \therefore \mathcal{O}_{str} \models N_a \sqsubseteq N \sqcup X$$

$$\because x_H \in N_a^{\mathcal{I}} \wedge x_H \notin X^{\mathcal{I}} \;\; \therefore x_H \in N^{\mathcal{I}} \;\; \therefore LS_H \cap N \neq \emptyset$$

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

$$\because \mathcal{O}_{str} \vdash H \sqsubseteq N' \sqcup Y \text{ and } x_{N_a} \in H^{\mathcal{I}} \;\; \therefore x_{N_a} \in (N' \sqcup Y)^{\mathcal{I}}$$

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

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

Next we prove $LS_H \subseteq LS_{N_a}$. For each $X \in LS_H$, there exists $N \prec_F X$ such that $\mathcal{O}_{str} \vdash H \sqsubseteq N \sqcup X$ and $N \cap LS_H = \emptyset$.

$$\because LS_{N_a} \subseteq LS_H \;\; \therefore N \cap LS_{N_a} = \emptyset \;\; \therefore x_{N_a} \notin N^{\mathcal{I}}$$

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

Thus we conclude $X \in LS_{N_a}$. □

Theorem 8 *Let $\mathcal{I}$ be a model of an $\mathcal{ALCHO}$ ontology $\mathcal{O}$ 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 $\mathcal{O} \cup \{L = \{x_L\}\}$ 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 tr, we have $L^{\mathcal{I}'} = \{x_L^{\mathcal{I}'}\}$, so the axiom $\{L = \{x_L\}\}$ 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 analyze all possible form of α case by case:

- $\alpha = \bigsqcap A_i \sqsubseteq \bigsqcup B_j$ 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 $(E \sqcap \neg F)^{\mathcal{I}'} \neq \emptyset$. □

Definition 9 *(Strengthening Axioms $\mathcal{O}_N^+$ and Strengthened Ontology $\mathcal{O}_{str}$) Let* $\mathsf{OMC}_{\mathsf{N_a}}$ *be an arbitrarily chosen set of atoms which is a superset of* $\mathsf{MC}_{\mathsf{N_a}}$. *$\mathcal{O}_N^+$ contains all such axioms as: $\forall N_a \in \mathbb{NP}, \forall X \in \mathsf{OMC}_{\mathsf{N_a}}$, $N_a \sqsubseteq X$ or $N_a \sqcap X \sqsubseteq \bot$. $\mathcal{O}_{str} = \mathcal{O}_{wk} \cup \mathcal{O}_N^+$.*

Theorem 10 *If $\mathcal{I}$ is a model of $\mathcal{O}_{str}$ for $E \sqcap \neg F$, $E, F \in \mathbb{C}^{\top,\bot}$, and all $\mathbb{NP}$ are condensing label in $\mathcal{I}$, then there exists a model of $\mathcal{O}_{in}$ for $E \sqcap \neg F$.*

Proof. Let the set of nominal axioms in $\mathcal{O}_{in}$ be $\{L_i \equiv \{x_{L_i}\}\}_{i=1}^n$. We prove $\mathcal{I}(E, F)$ can be transformed to a model $\mathcal{I}_n$ of $\mathcal{O}_{str}^n = \mathcal{O}_{str} \cup \{L_i \equiv \{x_{L_i}\}\}_{i=1}^n$ such that $(E \sqcap \neg F)^{\mathcal{I}_n} \neq \emptyset$ by induction on n.

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

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

Theorem 11 *Let $\mathcal{O}_{in}$ be an $\mathcal{ALCHO}$ ontology. Suppose there is a set of atoms* $\mathsf{OMC}_{\mathsf{N_a}}$ *such that* $\mathsf{OMC}_{\mathsf{N_a}} \supseteq \mathsf{MC}_{\mathsf{N_a}}$, $\forall N_a \in \mathbb{NP}$. *Then classification by $\mathcal{O}_{str}$ is complete w.r.t. $\mathcal{O}_{in}$.*

Proof. Because $\mathsf{OMC}_{\mathsf{N_a}} \supseteq \mathsf{MC}_{\mathsf{N_a}}$, $\forall N_a \in \mathbb{NP}$, by Lemma 6, 7, we know that for each occurring context H of each N_a, either $N_a \sqcap H \sqsubseteq \bot$ or $N_a \sqsubseteq H$ holds. Let $E, F \in \mathbb{C}^{\top,\bot}$ and $\mathcal{O}_{str} \not\models E \sqsubseteq F$, by Lemma 7, all $N_a \in \mathbb{NP}$ are condensing labels in the canonical model $\mathcal{I}(E, F)$. By Theorem 10, the model can be condensed to a model $\mathcal{I}'$ of $\mathcal{O}_{in}$ for $E \sqcap \neg F$, proving $\mathcal{O}_{in} \not\models E \sqsubseteq F$. So the classification result of $\mathcal{O}_{str}$ is complete w.r.t. $\mathcal{O}_{in}$. □

Thus, our next task is to compute $\mathsf{OMC}_{\mathsf{N_a}}$, and ensure $\mathsf{OMC}_{\mathsf{N_a}}$ is a superset of $\mathsf{MC}_{\mathsf{N_a}}$, and then calculate strengthening axioms $\mathcal{O}_N^+$ and obtain $\mathcal{O}_{str}$.

4 Computing the Strengthened Ontology

The primary task in strengthening is to calculate the set $\mathsf{OMC}_{\mathsf{N_a}}$. Since strengthening axioms participate in the saturation procedure, they may influence the canonical model construction of O_{str} and consequently $\mathsf{MC}_{\mathsf{N_a}}$. In order to ensure $\mathsf{MC}_{\mathsf{N_a}} \subseteq \mathsf{OMC}_{\mathsf{N_a}}$ for O_{str}, our algorithm is divided into two stages. In the first stage, we find the atoms of H where the derivation of axioms $H \sqsubseteq M \sqcup N_a$ does not rely on the strengthening axioms O^+_N. In the second phase, we enlarge the set computed from the first stage to address the influences of O^+_N to each $\mathsf{MC}_{\mathsf{N_a}}$.

Algorithm 2 gives the details of the first stage. The input of the algorithm is a concept X for which we want to find the set $\mathsf{OMC_X}$ such that for each $H \sqsubseteq M \sqcup X \in S(E)$, the atoms of H are in $\mathsf{OMC_X}$. To achieve this goal, we need to take a closer look at how $H \sqsubseteq M \sqcup X$ is derived in the saturation process. We divide the derivation process into two parts: **(1)** the initialization of H, i.e. how $\mathsf{init}(H)$ is derived, and **(2)** the derivation of $H \sqsubseteq M \sqcup X$ starting from $\mathsf{init}(H)$. We start with (2) and will address (1) later.

Algorithm 2: getPartialOMC

Input: Normalized $\mathcal{ALCHO}$ ontology O_{in} and a concept $X \in \mathbb{C}$, a set of atomic classes U

Output: Major coexisting label set $\mathsf{OMC_X}$, $Cset_X$

ToProcess $\leftarrow \{X\}$; $Cset_X \leftarrow \emptyset$; $exists_X \leftarrow \emptyset$;
repeat
    take out a label W from ToProcess;
    if $W \notin Cset_X$ **then**
        add W to $Cset_X$;
        if $\top \sqsubseteq M \bigsqcup W \in O_{in}$ **then** return and hand over the classification work to AR;
        foreach $\bigsqcap A_i \sqsubseteq M \bigsqcup W \in O_{in}$ **do** select one A_i and add it into ToProcess;
        foreach $\exists S.Y \sqsubseteq W \in O_{in}$ *and* $R \sqsubseteq^*_O S$ *and* $B \sqsubseteq \exists R.Z \in O_{in}$ **do**
            add B into ToProcess;
until ToProcess *is empty*;
foreach $W \in Cset_X$ **do**
    if $W \in U$ *or* $W \in \mathbb{NP}$ **then** add W to $\mathsf{OMC_X}$;
    foreach $Y \sqsubseteq \forall S.W \in O_{in}$ *and* $R \sqsubseteq^*_O S$ *and* $B \sqsubseteq \exists R.Z$ **do** add $\exists R.Z$ to $exists_X$;
    foreach $B \sqsubseteq \exists R.W \in O_{in}$ **do** add $\exists R.W$ to $exists_X$;
foreach $\exists R.W \in exists_X$ *and* $R \sqsubseteq^*_O S$ **do**
    add W to $\mathsf{OMC_X}$;
    foreach $Y \sqsubseteq \forall S.Z \in O_{in}$ **do** add Z to $\mathsf{OMC_X}$;
    foreach $\exists S.Z \sqsubseteq Y \in O_{in}$ **do** add Z to $\mathsf{OMC_X}$;
return $\langle \mathsf{OMC_X}, Cset_X \rangle$

To explain the algorithm, we first introduce some terminologies and notations.

Definition 12 *An **inference step*** IS *in a saturation process is one application of the inference rule. Each* IS *associates with an inference rule* $\mathsf{IS.rule}$*, a set of premises* $\mathsf{IS.prem}$*, a set of conclusions* $\mathsf{IS.conc}$ *and a set of side conditions* $\mathsf{IS.sc}$*. The conjunction H that occurs in* $\mathsf{IS.prem}$*, shown in Table 2, is called the context of* IS*. We write* $\mathsf{IS_H}$ *if the **inference step** is in the context H.*

Definition 13 *In a saturation $S(E)$, 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 $\mathsf{IS_H^1}, \ldots, \mathsf{IS_H^m}$ in the context H, where: (1) $\alpha \in \mathsf{IS_H^m.conc}$, and (2) for any $n < m$, $\mathsf{IS_H^n}$ occurs before $\mathsf{IS_H^{n+1}}$ in the saturation process.*

A derivation path contains all the key inference steps for deriving a conclusion from $\mathsf{init}(H)$. Note that $\mathsf{init}(H)$ can only be used as premises for $\mathbf{R_A^+}$ rule, and its conclusion is of the form $H \sqsubseteq A$. Hence any conclusion in context H is derived from one or more such axioms. In Algorithm 2, we first conduct a search in the converse direction of a derivation path of $H \sqsubseteq M \sqcup X$ to find one concept A where $H \sqsubseteq A$ is one of the axioms from which $H \sqsubseteq M \sqcup X$ is derived. We maintain a set $\mathsf{ToProcess}$ in which each concept W corresponds to some potential intermediate conclusion $H \sqsubseteq M' \sqcup W$ in the derivation path of $H \sqsubseteq M \sqcup X$. In an inference step $\mathsf{IS_H}$ of the derivation path, if W is a new disjunct in the right hand side of the conclusion axiom, then $\mathsf{IS_H.rule}$ can only be $\mathbf{R_\sqcap^n}$ or $\mathbf{R_\exists^-}$. Line 7 deals with the case where $\mathsf{IS_H.rule} = \mathbf{R_\sqcap^n}$ and $\bigsqcap A_i \sqsubseteq M \sqcup W \in \mathsf{IS_H.sc}$. The premises for this rule are all of the form $H \sqsubseteq N \sqcup A_i$. These are intermediate conclusions prior to $H \sqsubseteq M' \sqcup W$, and we select and add one A_i into $\mathsf{ToProcess}$. Lines 8 and 9 deal with the case where $\mathsf{IS_H.rule} = \mathbf{R_\exists^-}$ and $\mathsf{IS_H.sc} = \{\exists S.Y \sqsubseteq W, R \sqsubseteq_O^* S\}$. The premise is $H \sqsubseteq M' \sqcup R.K$, and there is another inference step $\mathsf{IS'_H}$ in which R is new to the conclusion. There must be $\mathsf{IS'_H.rule} = \mathbf{R_\exists^+}$, $B \sqsubseteq R.Z \in \mathsf{IS'_H.sc}$, and $\mathsf{IS'_H.prem} = \{H \sqsubseteq M'' \sqcup B\}$, and so we add B into $\mathsf{ToProcess}$. We initialize $\mathsf{ToProcess}$ with $\{X\}$, and keep adding into it all the possible labels that the processed label "comes from". This process covers all the potential derivation paths of $H \sqsubseteq M \sqcup X$ for all possible Hs. For each potential path, there are two possibilities that end the search:

(2.a) a conclusion $H \sqsubseteq B$ derived by $\mathbf{R_A^+}$ is reached,

(2.b) a conclusion $H \sqsubseteq M \sqcup B$ is derived using a strengthening axiom $N_b \sqsubseteq B$.

In case (2.a), B is a positive conjunct of H. Case (2.b) will be dealt with in Algorithm 3.

According to the discussion above, for those occurring contexts H which are in case (2.a), at least one of its positive conjuncts B is processed and added into $Cset_X$.

We return to the question of how $\mathsf{init}(H)$ was derived, and how to find all the conjuncts of H from B. There are two cases for $\mathsf{init}(H)$ to be derived:

(1.a) $\mathsf{init}(H)$ is introduced at the initialization of the saturation process;

(1.b) $\mathsf{init}(H)$ is derived using $\mathbf{R_{init}}$ rule. Let $H = \bigsqcap_{i=1}^n C_H^i$. In such case the premise is of the form $H^* \sqsubseteq M \sqcup \exists R.H$. Here is part of its derivation path:

$$H^* \sqsubseteq M_1 \sqcup A \xrightarrow[A\sqsubseteq\exists R.C_H^1]{\mathbf{R_\exists^+}} H^* \sqsubseteq M_1 \sqcup \exists R.C_H^1 \cdots \xrightarrow[R\sqsubseteq_O^* S \;\; Y\sqsubseteq\forall S.Z/\exists S.Z\sqsubseteq Y]{\mathbf{R_\forall}/\mathbf{R_\exists^-}} H^* \sqsubseteq M_n \sqcup \exists R.(\bigsqcap_{i=1}^n C_H^i)$$

We can see the first conjunct C_H^1 is added by $\mathbf{R}_\exists^+$ rule while the others are added either by $\mathbf{R}_\forall$ or $\mathbf{R}_\exists^-$. $\mathbf{R}_\forall$ adds only positive conjuncts, and $\mathbf{R}_\exists^-$ adds negative ones.

In lines 11 to 14, we check whether each concept W can be a potential conjunct of some occurring context H of the input X. For case (1.a), W is the only conjunct of H and is added to $\mathsf{OMC_X}$ in line 12. For case (1.b), $\exists R.C_H^1$ is first added into $exists_X$ in lines 13 and 14, and then all the conjuncts are added in the loop from lines 15 to 18.

Algorithm 3: getNominalStrAx (Calculate strengthening axioms for nominals)

Input: Normalized $\mathcal{ALCHO}$ ontology O_{in}
Output: Strengthening axioms O_N^+
1 **foreach** $N_a \in \mathbb{NP}$ **do**
2     $\langle \mathsf{OMC_{N_a}}, Cset_{N_a} \rangle \leftarrow$ getPartialOMC(N_a);
3   create a group g with $g.nominals = \{N_a\}$, $g.omc = \mathsf{OMC_{N_a}}$, $g.cset = Cset_{N_a}$;
4   add g into groups;
5 **repeat**
6   **if** *there exists* $g_i, g_j \in$ groups *such that* $g_i.omc \cap g_j.cset \neq \emptyset$ **then**
7   merge g_i, g_j into one group g, whose properties are unions of corresponding properties of g_i and g_j;
8   remove g_i, g_j from groups and add g;
9 **until** *no such* g_i, g_j *exists*;
10 **foreach** $g \in$ groups, $N_a \in g.nominals$ *and* $X \in g.omc$ **do**
11   add $N_a \sqsubseteq X$ or $N_a \sqcap X \sqsubseteq \bot$ to O_N^+;
12 **return** O_N^+

For the any occurring context H of N_a in case (2.b), the search ends at some inference step where $N_b \sqsubseteq B$ is the side condition, and $B \in Cset_{N_a}$. Since $N_b \sqsubseteq B \in O_{str}$, we have $B \in \mathsf{OMC_{N_b}}$, and so $B \in Cset_{N_a} \cap \mathsf{OMC_{N_b}}$. So in lines 5 to 9 of Algorithm 3, we merge $\mathsf{OMC_{N_a}}$ and $\mathsf{OMC_{N_b}}$ if $Cset_{N_a} \cap \mathsf{OMC_{N_b}}$ is not empty. Once merged, $\mathsf{OMC_{N_a}}$ and $\mathsf{OMC_{N_b}}$ are equal, which ensures that for each context H which $H \sqsubseteq M \sqcup N_a$ is derived from $H \sqsubseteq N \sqcup N_b$, the atoms of H are added into $\mathsf{OMC_{N_a}}$.

Algorithms 2 and 3 are polynomial in the size of ontology, which we measure by the number of axioms n_{ax} and the number of concepts n_c in the normalized ontology O_{in}. In Algorithm 2, the size of $Cset_X$ and $exists_X$ are bounded by the n_c and n_{ax}, respectively. Line 2 to 10 takes $O(n_c \cdot n_{ax})$ time, since each concept in $Cset_X$ is processed once in the outer loop, and each foreach-loop inside takes at most $O(n_{ax})$ time. Similarly, lines 11 to 14 also takes $O(n_c \cdot n_{ax})$ time, and lines 17 to 18 is $O(n_{ax}^2)$. So Algorithm 2 is $O(n_{ax}^2 + n_c \cdot n_{ax})$. In Algorithm 3, Algorithm 2 is invoked $O(n_c)$ time, and the merging process takes at most n_c^2 if all the nominals are merged into one group. So the whole procedure is polynomial.

4.1 Proof of $\mathsf{MC_{N_a}} \subseteq \mathsf{OMC_{N_a}}$

Lemma 14 *Let α be an axiom of the form $H \sqsubseteq M \sqcup A$ or $H \sqsubseteq M \sqcup A \sqcup \exists R.K$ in $S(E)$, and $\mathsf{IS_H^1}, \ldots, \mathsf{IS_H^m}$ is the derivation path of α, then there exists $B \in Cset_A$ and $n \leq m$*

such that (1) $\mathsf{IS}_\mathsf{H}^1, \ldots, \mathsf{IS}_\mathsf{H}^\mathsf{n}$ *is a derivation path of some* $H \sqsubseteq N \sqcup B \in S(E)$*; (2) either* $\mathsf{IS}_\mathsf{H}^\mathsf{n}.\mathsf{rule} = \mathbf{R}_\mathbf{A}^+$ *or* $\mathsf{IS}_\mathsf{H}^\mathsf{n}.\mathsf{sc}$ *contains a strengthening axiom of the form* $N_b \sqsubseteq B$*.*

Proof. According to line 1 of Algorithm 2, $A \in Cset_A$. We prove the lemma by induction over m.

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

Case 1 $\mathsf{IS}_\mathsf{H}^\mathsf{p}.\mathsf{rule}{=}\mathbf{R}_\mathbf{A}^+$ In this case we can choose $B = A$ and $n = p$ to prove the lemma.

Case 2 $\mathsf{IS}_\mathsf{H}^\mathsf{p}.\mathsf{rule}{=}\mathbf{R}_\sqcap^\mathbf{n}$ In this case $\mathsf{IS}_\mathsf{H}^\mathsf{p}.\mathsf{sc}$ has a single axiom α of the form $\bigsqcap A_i \sqsubseteq \bigsqcup B_j$. Note that by our assumption, A appears as a disjunct in the right hand side of a derived axiom for the first time in the conclusion, so there must be some B_j which is A, and so α cannot be of the form $\bigsqcap A_i \sqsubseteq \bot$. Hence if α is a strengthening axiom, then it can only be of the form $N_a \sqsubseteq A$, and we can choose $B = A$ and $n = p$ to make the lemma hold. Otherwise, by line 7 there exists some $A_i \in Cset_A$. Since $H \sqsubseteq N_i \sqcup A_i \in \mathsf{IS}_\mathsf{H}^\mathsf{p}.\mathsf{prem}$, its derivation path $\mathsf{IS}_\mathsf{H}^1, \ldots, \mathsf{IS}_\mathsf{H}^{\mathsf{p}'}$ must satisfy $p' < p \leq k$. By applying the inductive hypothesis to $m = p'$ and $H \sqsubseteq N_i \sqcup A_i$, there exists $B \in Cset_{A_i}$ and $n \leq p'$ such that conditions (1) and (2) hold. Since $A_i \in Cset_A$, according to the algorithm, we can see that $Cset_{A_i} \subseteq Cset_A$. So $B \in Cset_A$, and the lemma is proved.

Case 3 $\mathsf{IS}_\mathsf{H}^\mathsf{p}.\mathsf{rule}{=}\mathbf{R}_\exists^-$ In this case $\mathsf{IS}_\mathsf{H}^\mathsf{p}.\mathsf{sc}$ has axioms of the forms $R \sqsubseteq_O^* S$ and $\exists S.Y \sqsubseteq A$, and one of the premises $\mathsf{IS}_\mathsf{H}^\mathsf{p}.\mathsf{prem}$ is of the form $H \sqsubseteq M' \sqcup \exists R.K'$. The derivation process of $H \sqsubseteq M' \sqcup \exists R.K'$ is the same as (*), where H^* and H in (*) are replaced by H and K', respectively. The first inference step in (*) has a side condition of the form $A' \sqsubseteq \exists R.C_{K'}^1$ and a premise of the form $H \sqsubseteq M_1 \sqcup A'$. By line 8 to 9, A' is added to $Cset_A$ where $W = A$ and $Z = C_{K'}^1$. Let $\mathsf{IS}_\mathsf{H}^1, \ldots, \mathsf{IS}_\mathsf{H}^{\mathsf{p}'}$ 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 $B \in Cset_{A'}$ and $n \leq p'$ such that conditions (1) and (2) hold. Since $A' \in Cset_A$, $Cset_{A'} \subseteq Cset_A$. So $B \in Cset_A$, and the lemma is proved. □

Lemma 15 *Given* $H \sqsubseteq M \sqcup A \in S(E)$*, if a conjunct* B *of* H *is in* $Cset_A$*, then all atoms in* H *belongs to* $\mathsf{OMC_A}$*.*

Proof. Since all the conjuncts of H are added in the derivation of $\mathsf{init}(H)$, we discuss the two cases how $\mathsf{init}(H)$ is derived and 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 it is added to $\mathsf{OMC_A}$ in line 12 where $W = B$ and $U = \mathbb{C}^{\top,\bot}$.

- If $\mathsf{init}(H)$ is introduced by $\mathbf{R_{init}}$ rule, the derivation process is (*). The side condition of the first step is $A \sqsubseteq \exists R.C_H^1$. We first prove $\exists R.C_H^1 \in exists_A$. If B is C_H^1, then $\exists R.C_H^1$ is added to $exists_A$ in line 14 where $W = B$. If B is a conjunct of H other than C_H^1, 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 $\exists R.C_H^1$ is added to $exists_A$ in line 13 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 $C = C_H^1$ and is added to $\mathsf{OMC_A}$ in line 16.
  2. If C is added to the conjuncts of H by $\mathbf{R}_\forall$ rule, then C is added to $\mathsf{OMC_A}$ in line 17.
  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 $\mathsf{OMC_A}$ in line 18.

Hence the lemma is proved. □

Lemma 16 *In O_{str}, for each $N_a \in \mathbb{NP}$,* $\mathsf{MC}_{\mathsf{N_a}} \subseteq \mathsf{OMC}_{\mathsf{N_a}}$.

Proof. We prove the following equivalent statement:

> For each H and $N_a \in \mathbb{NP}$ such that $H \sqsubseteq M \sqcup N_a \in S(E)$,
> if X or $\neg X$ is a conjunct of H, then $X \in \mathsf{OMC}_{\mathsf{N_a}}$. (**)

Let the derivation path of $H \sqsubseteq M \sqcup N_a$ be $\mathsf{IS}_\mathsf{H}^1, \ldots, \mathsf{IS}_\mathsf{H}^\mathsf{m}$. We prove by induction over the m.

If $m = 1$, there must be $\mathsf{IS}_\mathsf{H}^\mathsf{m}.\mathsf{rule} = \mathbf{R_A^+}$. Since N_a is a conjunct of H, by lemma 15 all atoms of H are in $\mathsf{OMC}_{\mathsf{N_a}}$.

Next we prove statement (**) holds when $m = k$ if it holds for all N_a when $m < k$.

By apply Lemma 14 where $A = N_a$ and $m = k$, there exists $B \in Cset_{N_a}$ and $n \leq k$ such that (1) $\mathsf{IS}_\mathsf{H}^1, \ldots, \mathsf{IS}_\mathsf{H}^\mathsf{n}$ is a derivation path of some $H \sqsubseteq N \sqcup B \in S(E)$; (2) either $\mathsf{IS}_\mathsf{H}^\mathsf{n}.\mathsf{rule} = \mathbf{R_A^+}$ or $\mathsf{IS}_\mathsf{H}^\mathsf{n}.\mathsf{sc}$ contains a strengthening axiom of the form $N_b \sqsubseteq B$. Next we discuss the two cases.

Case (a) If $\mathsf{IS}_\mathsf{H}^\mathsf{n}.\mathsf{rule} = \mathbf{R_A^+}$, B is a conjunct of H, by lemma 15 all atoms of H are in $\mathsf{OMC}_{\mathsf{N_a}}$.

Case (b) If $\mathsf{IS}_\mathsf{H}^\mathsf{n}.\mathsf{sc}$ contains a strengthening axiom of the form $N_b \sqsubseteq B$, then $\mathsf{IS}_\mathsf{H}^\mathsf{n}.\mathsf{prem}$ is of the form $H \sqsubseteq M' \sqcup N_b$. Since the length of the derivation path of $H \sqsubseteq M' \sqcup N_b$ is less than k, by the inductive hypothesis we know all atoms of H are in $\mathsf{OMC}_{\mathsf{N_b}}$. Since $B \in \mathsf{OMC}_{\mathsf{N_b}} \cap Cset_{N_a}$, by lines 5 to 9 we have $\mathsf{OMC}_{\mathsf{N_b}} = \mathsf{OMC}_{\mathsf{N_a}}$. So all atoms are in $\mathsf{OMC}_{\mathsf{N_a}}$.

Hence the lemma is proved. □

4.2 Optimization

For each $X \in \mathsf{OMC}_{\mathsf{N_a}}$, if $N_a \sqsubseteq X$ or $N_a \sqcap X \sqsubseteq \bot$ is implied by O_{wk}, we do not add any axiom for X into O_N^+. Concretely, we run Algorithm 2 before the first round classification, introduce a new axiom $X_a \sqsubseteq N_a \sqcap X$ where X_a is a fresh concept. After the first round classification, if $X_a \sqsubseteq \bot$ is found, then $N_a \sqcap X \sqsubseteq \bot$ is implied. $N_a \sqsubseteq X$ can also be known from $\mathcal{H}_{wk}$.

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 in the original ontology, and $N_a \sqsubseteq X$ is not implied, then we add $N_a \sqcap X \sqsubseteq \bot$. For other cases, we add $N_a \sqsubseteq X$.

In Algorithm 3, lines 5 to 9 is to merge the OMCs of two nominal placeholders under some condition. Such operation may greatly increase the labels of the nominal placeholders. To reduce the number of labels, we can remove some labels from each $\mathsf{OMC}_{\mathsf{N_a}}$ before executing lines 5 to 9. We call this operation **optimization1**. The removed labels are from the positive labels $X \in \mathsf{OMC}_{\mathsf{N_a}}$ in line 2 of Algorithm 3, and these Xs are those we choose to add $N_a \sqcap X \sqsubseteq \bot$ to O_N^+(we add $N_a \sqcap X \sqsubseteq \bot$ to O_N^+ before removing X from $\mathsf{OMC}_{\mathsf{N_a}}$). It can be removed because if $N_a \sqcap X \sqsubseteq \bot$ is chosen and X is a disjunct of H, we can prove that in $S(E)$, any conclusion $H \sqsubseteq M \sqcup N_a$ becomes *redundant conclusions* and can be removed.

Definition 17 *(Redundant conclusion) An axiom $\alpha \in S(E)$ is called a redundant conclusion if there exists an axiom $\beta \in S(E)$, α, β are of the form $H \sqsubseteq M$ or $H \sqsubseteq M \sqcup \exists R.K$, and all the disjuncts in the right hand side of β are disjuncts in the right hand side of α. The set of redundant conclusions are denoted by $R_S(E)$.*

Remark 18 *According to section 5.5 of František* et.al. *[18], in the saturation procedure, we can safely delete any conclusion immediately from the saturation once it is found to be redundant, and the models constructed for the saturation will not change.*

Lemma 19 *In the strengthened ontology produced with **optimization1**, we have $\mathsf{MC}_{\mathsf{N_a}} \subseteq \mathsf{OMC}_{\mathsf{N_a}}$ for each nominal placeholder N_a.*

Proof. To prove the conclusion, we need to prove the optimization operation does not affect the completeness if we remove redundant conclusions during the saturation procedure.

Note that the optimization operation only affects the execution result of lines 5 to 9 of Algorithm 3. This affects only the case (b) of Lemma 16, which requires $N_b \sqsubseteq B \in \mathsf{IS}^{\mathsf{n}}_{\mathsf{H}}.\mathsf{sc}$. When $N_b \sqsubseteq B \in \mathsf{IS}^{\mathsf{n}}_{\mathsf{H}}.\mathsf{sc}$, $\mathsf{IS}^{\mathsf{n}}_{\mathsf{H}}.\mathsf{rule}{=}\mathbf{R}^{\mathbf{n}}_{\sqcap}$ and $H \sqsubseteq N \sqcup N_b \in \mathsf{IS}^{\mathsf{n}}_{\mathsf{H}}.\mathsf{prem}$ must hold.

Next we prove in the scenario such that: (i) H contains a positive conjunct X; (ii) $N_b \sqcap X \sqsubseteq \bot \in O_{str}$, case (b) will not happen. From (i) we know $H \sqsubseteq X$ is derived by $\mathbf{R}^+_{\mathbf{A}}$ rule. If there is an axiom of the form $H \sqsubseteq N \sqcup N_b$ derived in the saturation process, we will apply $\mathbf{R}^{\mathbf{n}}_{\sqcap}$ rule to these two axioms with a side condition $N_b \sqcap X \sqsubseteq \bot$, and $H \sqsubseteq N$ is derived. After this step, $H \sqsubseteq N \sqcup N_b$ becomes redundant and is deleted immediately, and then it will not participate as a premise for other derivations in the saturation procedure. That means it is impossible that $\mathsf{IS}^{\mathsf{n}}_{\mathsf{H}}.\mathsf{rule}{=}\mathbf{R}^{\mathbf{n}}_{\sqcap}$ and $H \sqsubseteq N \sqcup N_b \in \mathsf{IS}^{\mathsf{n}}_{\mathsf{H}}.\mathsf{prem}$, and so case (b) of Lemma 16 is impossible. That is to say, such Hs cannot get $H \sqsubseteq M \sqcup N_a$ through $H \sqsubseteq N \sqcup N_b$, thus we do not need to add X into $\mathsf{OMC}_{\mathsf{N_a}}$.

Hence if (i) and (ii) hold, we remove X from $\mathsf{OMC}_{\mathsf{N_b}}$, and X will not be added into $\mathsf{OMC}_{\mathsf{N_a}}$ in lines 5 to 9, and the conclusion of Lemma 16 still holds. □

5 Experiments and Evaluation

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 [11], 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 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 except the recent release of MORe, for which we set the time limit the same as classification time of HermiT for Galen and FMA 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)[2] 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[3]. Galen-EL-n1Y and Galen-EL-n2Y were provided [13]. Galen-Heart-n1 and Galen-Heart-n2 are subontologies, respectively, referring to the human heart. 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 smaller complex ontologies – Wine and DOLCE. We use the ORE dataset,[4] where 2 ontologies without axioms are removed. In all cases, we reduce the language to $\mathcal{SHO}$. The ontologies are available from our website.[5].

The results are shown in Table 3. We found that the optimal configurations for HermiT when running the large and complex ontologies were simple core blocking and individual reuse. Excluding ORE, WSClassifier achieves better efficiency than the tableau-based reasoners on 7 out of 10 ontologies. For Wine, Galen-EL-YN1 and Galen-EL-YN2, WSClassifier, incurring a relatively small cost, detected that strengthening axioms made some concepts unsatisfiable in O_{str}, and so failed over to HermiT.

We see a major speedup for WSClassifier on ORE's FMA-lite. On the other 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 $\mathcal{H}_{str}$ and only 1 introduced new unsatisfiable concepts and fails over to HermiT. Thus the WS approach does not incur much additional work, and most of the addition-

---

[2] Foundational Model of Anatomy, http://sig.biostr.washington.edu/projects/fm/index.html

[3] http://code.google.com/p/condor-reasoner/downloads/list

[4] http://www.cs.ox.ac.uk/isg/conferences/ORE2012/

[5] http://isel.cs.unb.ca/~wsong/WSClassifierExperimentOntologies.zip

Table 3: 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	–	–	–	4.1
Galen-Heart-n2	3366	92	768.4	–	–	–	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	148	140,882.0	–	–	–	21.2
ORE-dataset (OWL DL & EL,113 ontologies) the following refers to average number							
other 112 ontologies	4293	343	0.84	0.86	–*	0.24	2.10
FMA-lite	75,141	0	137,409.0	–	–	–	26.0

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

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

al 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.

Table 4 shows some statistics of WSClassifier on different phases. For FMA-C, there are no strengthening axiom added, and only one round of classification by MR is needed. For DOLCE, Galen-Heart-n2, Galen-EL-n1YE and Galen-Union-n, there are no new subsumptions derived from O_{str}, and so the verification phase by AR is not needed. For Galen-EL-n1Y, Galen-EL-n2Y and Wine, our strengthening produces incorrect unsatisfiable concepts, so the classification fails over to HermiT. The number of strengthening axioms for these ontologies is large which increases the risk of concepts in $\mathbb{NP}$ and other atomic concepts $A \in \mathbb{C}$ becoming unstaisfiable. For Galen-EL-n1Y and Galen-EL-n2Y, difficulty arises from axioms of the form $A \equiv B \sqcap \exists R.C$ where B is a general concept and R is a frequently occurring role. Two of the normalized axioms coming from this are $B \sqcap A_{\exists R.C} \sqsubseteq A$ and $\exists R.A_C \sqsubseteq A_{\exists R.C}$. Once A is added to some $Cset_{N_a}$ in Algorithm 2, then either B or $A_{\exists R.C}$ needs to be added to $Cset_{N_a}$ in line 7. We choose to add B to $Cset_{N_a}$ in line 7, which causes a large number of its subconcepts to be added to $\mathsf{OMC}_{\mathsf{N_a}}$. In the merge process from lines 5 to 9 in Algorithm 3, this $\mathsf{OMC}_{\mathsf{N_a}}$ will create a large nominal group g creating many strengthening axioms. For the Wine ontology and its variant food ontology in ORE-dataset, a large $\mathsf{OMC}_{\mathsf{N_a}}$ is caused by roles occurred in line 15 of Algorithm 2 creating many labels in line 17 and 18.

WSClassifier seems most applicable when the ontologies are large and highly cyclic

Table 4: Statistics of WSClassifier

Ontology	**unsatisfiable concepts**	**strengthening axioms added**	**potential subsumptions**	**confirmed subsumptions**	**checking time(s)**
Wine	166	982	/	/	/
Dolce	0	14	0	0	0
Galen-Heart-n1	0	2	2	2	1.1
Galen-Heart-n2	0	13	0	0	0
Galen-EL-n1Y	25,386	236,069	/	/	/
Galen-EL-n2Y	24,682	18,916	/	/	/
Galen-EL-n1YE	0	17	0	0	0
Galen-EL-n2YE	0	209	2,039	202	25,613
Galen-Union-n	0	2,624	0	0	0
FMA-C	0	0	0	0	0
ORE-dataset (OWL DL & EL,113 ontologies)					
food (variant of Wine)	48	791	/	/	/
other 112 ontologies	0	102 (total)	0	0	0

Note: "/" entry means that WSClassifier fails over to HermiT and the number is not applicable

since then tableau reasoners construct large models and employ expensive blocking s-trategies. On the other hand consequence-based reasoners do not encounter problems on highly cyclic ontologies, and so can classify even cyclic $\mathcal{O}_{wk}$ and $\mathcal{O}_{str}$ quickly. If there are no or just a few additional subsumptions derived by $\mathcal{O}_{str}$, AR does not need or just do a little work on the highly cyclic $\mathcal{O}_{in}$. This improvement is observed for FMA-C which is the only real-world large and complex ontologies with nominals we have. Of the 51 ORE ontologes with nominals, only one has additional subsumptions. This evidence suggests nominals in real world ontologies seldom produce new subsumptions; this suggests our approach is valuable.

6 Related Work

Optimization techniques for ontology classification have been extensively studied in the literature [4, 17, 8, 13]. For tableau-based reasoners, Enhanced Traversal (ET) [4] and KP [17, 8] are the most widely used techniques. Optimizations for consequence-based classification of $\mathcal{ELO}$ ontologies were also studied [13], 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 $\mathcal{O}_{in}$, they find a set of classes $\Sigma^{\mathcal{L}}$ whose superclasses in $\mathcal{O}_{in}$ can be computed by classifying a subset $\mathcal{M}^{\mathcal{L}}$ of $\mathcal{O}_{in}$ 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 [13] use of nominals in the Galen ontologies, so R has to do all the work. In contrast, our approach supports a weaker language $\mathcal{ALCHO}$, combines the two reasoners differently, handles nominals, and improves on its full reasoner more often for complex and highly cyclic ontologies.

Knowledge approximation [16] has been applied to encode $\mathcal{SROIQ}$ ontologies into $\mathcal{EL}^{++}$ with additional data structures, and classified by a tractable, sound but incomplete algorithm [15]. A strengthened approximation of $\mathcal{SROIQ}$ TBoxes with the OWL 2 RL profile [23] is used for query answering.

7 Concluding Remarks

We have presented a hybrid reasoning technique for sound and complete 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_{wk} and one strengthened O_{str}, which are classified by a fast consequence-based reasoner. The subsumptions of O_{wk} and O_{str} are a subset and a superset of the subsumptions of the original ontology, respectively. Subsumptions implied by O_{str} but not by O_{wk} are further checked by a (slower) $\mathcal{ALCHO}$ reasoner. This general approach can be applied to different language classes, each requiring different strengthening axioms. The implementation can be improved with heuristics for selecting a tighter OMC 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] Cuenca Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. J. Web Semantics 6(4), 309–322 (2008)

[8] Glimm, B., Horrocks, I., Motik, B., Shearer, R., Stoilos, G.: A novel approach to ontology classification. J. Web Semantics 14, 84–101 (2012)

[9] Haarslev, V., Möller, R.: RACER system description. In: Proc. of IJCAR. pp. 701–705 (2001)

[10] Horrocks, I., Sattler, U.: A Tableau decision procedure for $\mathcal{SHOIQ}$. J. Automated Reasoning 39(3), 249–276 (2007)

[11] Kazakov, Y.: Consequence-driven reasoning for Horn $\mathcal{SHIQ}$ ontologies. In: Proc. of IJCAI. pp. 2040–2045 (2009)

[12] Kazakov, Y., Krötzsch, M., Simančík, F.: Concurrent classification of $\mathcal{EL}$ ontologies. In: Proc. of ISWC. pp. 305–320 (2011)

[13] 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)

[14] Motik, B., Shearer, R., Horrocks, I.: Hypertableau reasoning for description logics. J. Artificial Intelligence Research 36(1), 165–228 (2009)

[15] Ren, Y., Pan, J.Z., Zhao, Y.: Soundness preserving approximation for TBox reasoning. In: Proc. of AAAI (2010)

[16] Selman, B., Kautz, H.: Knowledge compilation and theory approximation. J. ACM 43(2), 193–224 (1996)

[17] Shearer, R., Horrocks, I.: Exploiting partial information in taxonomy construction. In: Proc. of ISWC. pp. 569–584 (2009)

[18] Simančík, F., Kazakov, Y., Horrocks, I.: Consequence-based reasoning beyond Horn ontologies. In: Proc. of IJCAI. pp. 1093–1098 (2011)

[19] 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)

[20] 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)

[21] Song, W., Spencer, B., Du, W.: Technical details of a complete classification of complex $\mathcal{ALCHO}$ ontologies using a hybrid reasoning approach. Tech. rep. (2013), http://www.cs.unb.ca/tech-reports/documents/TR13-222.pdf

[22] Tsarkov, D., Horrocks, I.: FaCT++ description logic reasoner: System description. In: Proc. of IJCAR. pp. 292–297 (2006)

[23] Zhou, Y., Cuenca Grau, B., Horrocks, I.: Efficient upper bound computation of query answers in expressive description logics. In: Proc. of DL (2012)