Soundness Preserving Approximation for TBox Reasoning

Yuan Ren and **Jeff Z. Pan** and **Yuting Zhao**
Department of Computing Science
University of Aberdeen
Aberdeen, UK

Abstract

Large scale ontology applications require efficient and robust description logic (DL) reasoning services. Expressive DLs usually have very high worst case complexity while tractable DLs are restricted in terms of expressive power. This brings a new challenge: can users use expressive DLs to build their ontologies and still enjoy the efficient services as in tractable languages. In this paper, we present a soundness preserving approximate reasoning framework for TBox reasoning in OWL2-DL. The ontologies are encoded into $\mathcal{EL}^{++}$ with additional data structures. A tractable algorithm is presented to classify such approximation by realizing more and more inference patterns. Preliminary evaluation shows that our approach can classify existing benchmarks in large scale efficiently with a high recall.

1 Introduction

Ontologies have been so phenomenally successful, as a machine-understandable compilation of human knowledge, that OWL2 (the second version of OWL) is recently standardized by W3C. As more and more large ontologies become available (HermiT-Benchmark 2009), there is a pressing need for efficient and robust reasoning services.

Expressive Description Logics (DLs) (Baader et al. 2003) have high worst case computational complexity. For example, classification in the DL $\mathcal{SROIQ}$ (Horrocks, Kutz, and Sattler 2006), the adjacent logic of OWL2-DL, is N2EXPTIME-complete (Kazakov 2008). Mainstream reasoners for expressive DLs provide reasoning services based on tableau (Horrocks, Kutz, and Sattler 2006) and hyper-tableau (Motik, Shearer, and Horrocks 2009) algorithms. Such *model constructing* algorithms classify an ontology, in general, by iterating all necessary pairs of concepts, and trying to construct a model of the ontology that violates the subsumption relation between them (Kazakov 2009). On the other hand, light-weight DLs can have very efficient reasoning algorithms. For example, TBox reasoning in $\mathcal{EL}^{++}$ (Baader, Brandt, and Lutz 2005), the logic underpinning of an OWL2 tractable profile OWL2-EL, is PTIME-complete. However, their expressive power is limited.

This brings a new challenge: can users use OWL2-DL to build their ontologies and still enjoy the efficient reasoning as in tractable profiles? For example, the Foundational Model of Anatomy ontology (FMA) , which is built in $\mathcal{ALCOIF}$, beyond any tractable DLs, can hardly be classified by any mainstream DL reasoners (Motik, Shearer, and Horrocks 2009). Given the current efforts of ontology construction, it might not take long before many other FMA-like (or even larger and more complicated) ontologies appear and go beyond the capability of existing DL reasoners.

Approximation (Stuckenschmidt and van Harmelen 2002; Groot, Stuckenschmidt, and Wache 2005; Hitzler and Vrandecic 2005; Wache, Groot, and Stuckenschmidt 2005; Pan and Thomas 2007) has been identified as a potential way to reduce the complexity of ontology reasoning. However, many of these approximation approaches still rely on the reasoners of the more expressive DLs (Groot, Stuckenschmidt, and Wache 2005; Pan and Thomas 2007). Furthermore, most of the above approaches are on ABox reasoning and query answering. To the best of our knowledge, the only approach on TBox reasoning is (Groot, Stuckenschmidt, and Wache 2005), which presents an overview of approximation approaches, including language weakening, knowledge compilation and approximate deduction, as well as investigating and reporting negative results of the approximate deduction approach – the collapsing of concept expressions leads to many unnecessary approximation steps.

In this paper, we propose to combine the idea of language weakening and approximate deduction into soundness preserving approximation for TBox reasoning of very expressive DLs. Our contributions are the following:

1. After an informative discussion of the technical challenges (Sec.2), we propose a syntactic language weakening approach (Sec.3) to approximating an arbitrary $\mathcal{SROIQ}$ TBox with a corresponding $\mathcal{EL}^{++}$ TBox and additional data structures maintaining the complement and cardinality information. It is shown that the proposed approximation is in linear time (Lemmas 1, 2 and 3).

2. We present soundness-guaranteed approximate deduction rules to classify the approximated TBox (Sec.3). In contrast to the twisted trade-off between tractability and expressiveness, our approach compromises the completeness of reasoning to yield large portion of logical con-

sequences in polynomial time while imposing no restrictions on expressivity of the language used.

3. We present our implementations and preliminary evaluations (Sec.4). Evaluation against a set of real world ontologies (HermiT-Benchmark 2009) suggested that, our approach can (i) outperform existing OWL2-DL reasoners, and (ii) provide rather complete results with high recall (over 95% when complement approximated and over 99% when cardinality approximated).

All the proofs can be found in our technical report available at http://www.box.net/shared/nm913g22ie.

2 Technical Motivations

In DL $\mathcal{SROIQ}$, concepts C, D can be inductively composed with the following constructs: $\top \mid \bot \mid A \mid C \sqcap D \mid \exists R.C \mid \{a\} \mid \neg C \mid \ \geq nR.C \mid \exists R.Self$, where $\top$ is the top concept, $\bot$ the bottom concept, A an atomic concept, n an integer number, a an individual, $\exists R.Self$ the self-restriction and R a role that can be either an atomic role r or the inverse of another role (R^-). Conventionally, $C \sqcup D, \forall R.C$ and $\leq nR.C$ are used to abbreviate $\neg(\neg C \sqcap \neg D), \neg\exists R.\neg C$ and $\neg \geq (n+1)R.C$, respectively. $\{a_1, a_2, \ldots, a_n\}$ can be regarded as abbreviation of $\{a_1\} \sqcup \{a_2\} \sqcup \ldots \sqcup \{a_n\}$. Without loss of generality, in what follows, we assume all the concepts to be in their negation normal forms (NNF. A concept is in NNF *iff* $\neg$ is applied only to A, $\{a\}$ or $\exists R.Self$. NNF of a given concept can be computed in linear time (Hollunder, Nutt, and Schmidt-Schauß 1990).) and use $\sim C$ to denote the NNF of $\neg C$. We call $\top, \bot, A, \{a\}$ *basic concepts*. Given a TBox $\mathcal{T}$, we use $CN_\mathcal{T}$ ($RN_\mathcal{T}$) to denote the set of basic concepts (atomic roles) in $\mathcal{T}$. The $\mathcal{EL}$ family is dedicated for large TBox reasoning and has been widely applied in some of the largest ontologies. $\mathcal{EL}^{++}$ supports $\top \mid \bot \mid A \mid C \sqcap D \mid \exists r.C \mid \{a\}$.

Both $\mathcal{SROIQ}$ and $\mathcal{EL}^{++}$ support concept inclusions (CIs, e.g. $C \sqsubseteq D$) and role inclusions (RIs, e.g. $r \sqsubseteq s$, $r_1 \circ \ldots \circ r_n \sqsubseteq s$). $\mathcal{SROIQ}$ supports also other axioms such asymmetric of roles. If $C \sqsubseteq D$ and $D \sqsubseteq C$, we write $C \equiv D$. If C is non-atomic, $C \sqsubseteq D$ is a general concept inclusion (GCI). For more details about syntax and semantics of DLs, we refer the readers to (Baader et al. 2003).

A TBox is a set of concept and role axioms. TBox reasoning services include concept subsumption checking, concept satisfiability checking (to check if a given concept is instantiable) and classification (to compute the concept hierarchy). For example, given the following TBox $\mathcal{T}_1$ (in $\mathcal{ALC}$), we can infer $Koala \sqsubseteq Herbivore$.

Example 1 *An example TBox $\mathcal{T}_1$.*

- $\alpha_1 : Koala \sqsubseteq \forall eat.(\exists partof.Eucalypt)$
- $\alpha_2 : Eucalypt \sqsubseteq Plant$
- $\alpha_3 : Plant \sqcup \exists partof.Plant \sqsubseteq VegeFood$
- $\alpha_4 : \forall eat.VegeFood \sqsubseteq Herbivore$

The tableau algorithm (Horrocks, Kutz, and Sattler 2006) constructs a tableau as a graph in which each node x represents an individual and is labeled with a set of concepts it must satisfy, each edge $\langle x, y\rangle$ represents a pair of individuals satisfying a role that labels the edge. Subsumption checking $C \sqsubseteq D$ can be reduced to unsatisfiability checking $C \sqcap \neg D \sqsubseteq \bot$. To test this, a tableau is initialised with a single node labeled with $C \sqcap \neg D$, and is then expanded by repeatedly applying the completion rules. One of the major difficulties for tableau algorithms is the high degree of non-determinism introduced by GCIs. For each GCI $C \sqsubseteq D$ in the ontology, the algorithm generates a meta-constraint $\neg C \sqcup D$ for each node of the tableau. This leads to an exponential blowup of the search space. Some *Absorption* techniques (Tsarkov, Horrocks, and Patel-Schneider 2007; Tsarkov and Horrocks 2004) have been developed to deal with GCIs. However, they can only be applied to a limited pattern of GCIs; e.g., α_4 can not be dealt with by any absorption optimisation.

Reasoning with $\mathcal{EL}^{++}$ is more efficient. Baader, Brandt and Lutz (2005) present a set of completion rules (Table 1) to compute, given a normalised $\mathcal{EL}^{++}$ TBox T, for each $A \in CN_T$, a subsumer set $S(A) \subseteq CN_T \cup \{\bot\}$ in which for each $B \in S(A)$, $T \models A \sqsubseteq B$, and for each $r \in RN_T$, a relation set $R(r) \subseteq CN_T \times CN_T$ in which for each $(A, B) \in R(r)$, $T \models A \sqsubseteq \exists r.B$. Reasoning with rules **R1-R8** is tractable.

Table 1: $\mathcal{EL}^{++}$ completion rules (no datatypes)

R1	If $A \in S(X)$, $A \sqsubseteq B \in \mathcal{T}$ and $B \notin S(X)$ then $S(X) := S(X) \cup \{B\}$
R2	If $A_1, A_2, \ldots, A_n \in S(X)$, $A_1 \sqcap A_2 \sqcap \ldots \sqcap A_n \sqsubseteq B \in \mathcal{T}$ and $B \notin S(X)$ then $S(X) := S(X) \cup \{B\}$
R3	If $A \in S(X)$, $A \sqsubseteq \exists r.B \in \mathcal{T}$ and $(X, B) \notin R(r)$ then $R(r) := R(r) \cup \{(X, B)\}$
R4	If $(X, A) \in R(r)$ $A' \in S(A)$, $\exists r.A' \sqsubseteq B \in \mathcal{T}$ and $B \notin S(X)$ then $S(X) := S(X) \cup \{B\}$
R5	If $(X, A) \in R(r)$, $\bot \in S(A)$ and $\bot \notin S(X)$ then $S(X) := S(X) \cup \{\bot\}$
R6	If $\{a\} \in S(X) \cap S(A)$, $X \rightsquigarrow_R A$, $S(A) \not\subseteq S(X)$ then $S(X) := S(X) \cup S(A)$
R7	If $(X, A) \in R(r)$, $r \sqsubseteq s \in \mathcal{T}$ and $(X, A) \notin R(s)$ then $R(s) := R(s) \cup \{(X, A)\}$
R8	If $(X, A) \in R(r_1)$, $(A, B) \in R(r_2)$, $r_1 \circ r_2 \in \mathcal{T}$, and $(X, B) \notin R(r_3)$ then $R(r_3) := R(r_3) \cup \{(X, B)\}$

However, these rules cannot handle $\mathcal{T}_1$ because the ontology is in a language beyond the $\mathcal{EL}^{++}$.

Groot et al. (2005) attempt to speed up concept unsatisfiability checking via approximation. Given a concept C, they construct a sequence of $C_i^\top$ such that $C \sqsubseteq \ldots \sqsubseteq C_1^\top \sqsubseteq C_0^\top$, and a sequence of $C_i^\bot$ such that $C_0^\bot \sqsubseteq C_1^\bot \sqsubseteq \ldots C$ by replacing all existential restrictions ($\exists R.D$) after i universal quantifiers ($\forall$) inside C with $\top$ and $\bot$ respectively. Then C is unsatisfiable (satisfiable) if some $C_i^\top$ ($C_i^\bot$) is unsatisfiable (satisfiable), which is easier to check. This approach has several limitations when applied to TBox reasoning: (i) It only approximates the tested concept, but not the ontology, thus the complexity of the unsatisfiability checking is not reduced. (ii) Similar to the Tableau algorithms, one has to reduce concept subsumption $C \sqsubseteq D$ to unsatisfiability of $C \sqcap \neg D$ for each necessary pair of C, D. (iii) When the test concept subsumption contains no existential restriction, such as $Koala \sqsubseteq Herbivore$, this approach can not help. Hence, it does not help for classification (subsumption checking among named concepts).

3 Approach

Different from Groot et al.'s approach, we approximate both the ontology and the tested concept (if needed) by replacing concept sub-expressions (role expressions) that are not in the target DL, e.g. $\mathcal{EL}^{++}$, with atomic concepts (atomic roles) and rewrite axioms accordingly (Sec 3.1). Then, additional data structures and completion rules (Sec 3.2 and Sec 3.3) are used to maintain and restore some semantic relations among basic concepts, respectively.

In approximation, we only consider concepts corresponding to the particular TBox in question. We use the notion *term* to refer to these "interesting" concept expressions. More precisely, a term is: (i) a concept expression on the LHS or RHS of any CI, or (ii) the complement of a term, or (iii) the syntactic sub-expression of a term.

In order to represent terms and role expressions that will be used in $\mathcal{EL}^{++}$ reasoning, we assign names to them.

Definition 1 ***(Name Assignment)*** *Given S a set of concept expressions, E a set of role expressions, a* name assignment *fn is a function that for each $C \in S$ ($R \in E$), $fn(C) = C$ ($fn(R) = R$) if C is a basic concept (R is atomic); otherwise, $fn(C)$ ($fn(R)$) is a fresh name.*

Names of some terms in $\mathcal{T}_1$ are illustrated in Table 2.

Table 2: Name Assignment

Term	Name
$\forall eat.\exists partof.Eucalypt$	C_1
$\exists eat.\forall partof.\neg Eucalypt$	nC_1
$\forall partof.\neg Eucalypt$	C_2
$\exists partof.Eucalypt$	nC_2
$Plant \sqcup \exists partof.Plant$	C_3
$\neg Plant \sqcap \forall partof.\neg Plant$	nC_3
$\forall partof.\neg Plant$	C_4
$\exists partof.Plant$	nC_4
$\forall eat.VegeFood$	C_5
$\exists eat.\neg VegeFood$	nC_5
$\neg Plant$	$nPlant$
$\neg VegeFood$	$nVegeFood$

3.1 $\mathcal{EL}^{++}$ Approximation

Definition 2 ***($\mathcal{EL}^{++}$ Transformation)*** *Given a TBox $\mathcal{T}$ and a name assignment fn, its $\mathcal{EL}^{++}$* transformation *$A_{fn,\mathcal{EL}^{++}}(\mathcal{T})$ is a set of axiom T constructed as follows:*

1. *T is initialised as $\emptyset$.*
2. *for each $C \sqsubseteq D$ ($C \equiv D$) in $\mathcal{T}$, $T = T \cup \{fn(C) \sqsubseteq fn(D)\}$ ($T = T \cup \{fn(C) \equiv fn(D)\}$).*
3. *for each $\mathcal{EL}^{++}$ role axiom $\beta \in \mathcal{T}$, add $\beta_{[R/fn(R)]}$ into T.*
4. *for each term C in $\mathcal{T}$,*
   (a) *if C is of the form $C_1 \sqcap \ldots \sqcap C_n$, then $T = T \cup \{fn(C) \equiv fn(C_1) \sqcap \ldots \sqcap fn(C_n)\}$,*
   (b) *if C is of the form $\exists R.D$, then $T = T \cup \{fn(C) \equiv \exists fn(R).fn(D)\}$,*
   (c) *otherwise $T = T \cup \{fn(C) \sqsubseteq \top\}$.*

We call this procedure an *$\mathcal{EL}^{++}$ approximation.*

Lemma 1 *For a TBox $\mathcal{T}$ and a name assignment fn, let $A_{fn,\mathcal{EL}^{++}}(\mathcal{T}) = T$. We have T is an $\mathcal{EL}^{++}$ TBox and $|T| \leq n_{\mathcal{T}} + |\mathcal{T}|$ where $n_{\mathcal{T}}$ is the number of terms in $\mathcal{T}$ and $|T|$ ($|\mathcal{T}|$) is the number of axioms in T ($\mathcal{T}$) .*

With Table 2, some axioms from approximation of $\mathcal{T}_1$ are:

Example 2 *$T_{Koala} \supseteq \{Koala \sqsubseteq C_1$, $nC_1 \equiv \exists eat.C_2$, $nC_2 \equiv \exists partof.Eucalypt, \alpha_2, C_3 \sqsubseteq VegeFood$, $nC_3 \equiv nPlant \sqcap C_4$, $nC_4 \equiv \exists partof.Plant, C_5 \sqsubseteq Herbivore$, $nC_5 \equiv \exists eat.nVegeFood\}$.*

3.2 Complement-enriched $\mathcal{EL}^{++}_{\mathcal{C}}$ Approximation

In Example 2, $Koala \sqsubseteq Herbivore$ can not be inferred with **R1-R8** because the relations between a term and its complement, e.g. C_1 and nC_1, are lost. To solve this problem, we maintain such relations in a separate *complement table (CT)*, and apply additional completion rule in reasoning.

Definition 3 ***($\mathcal{EL}^{++}_{\mathcal{C}}$ Transformation)***
Given a TBox $\mathcal{T}$ and a name assignment fn, its complement-enriched $\mathcal{EL}^{++}_{\mathcal{C}}$ transformation *$A_{fn,\mathcal{EL}^{++}_{\mathcal{C}}}(\mathcal{T})$ is a pair (T, CT) constructed as follows:*

1. *$T = A_{fn,\mathcal{EL}^{++}}(\mathcal{T})$ (Ref. Def. 2).*
2. *CT is initialised as $\emptyset$.*
3. *for each term C in $\mathcal{T}$, $CT = CT \cup \{(fn(C), fn({\sim}C))\}$.*

We call this procedure an *$\mathcal{EL}^{++}_{\mathcal{C}}$ approximation.*

Proposition 2 ***($\mathcal{EL}^{++}_{\mathcal{C}}$ Approximation)*** *For a TBox $\mathcal{T}$, let $A_{fn,\mathcal{EL}^{++}_{\mathcal{C}}}(\mathcal{T}) = (T, CT)$, we have:*

1. *T is an $\mathcal{EL}^{++}$ TBox*
2. *for each $A \in CN_T$, there exists $(A, B) \in CT$*
3. *if $(A, B) \in CT$ then $A, B \in CN_T$ and $(B, A) \in CT$*

This indicates that, by Def.3, a TBox can be syntactically transformed into an $\mathcal{EL}^{++}$ TBox with a table maintaining complementary relations for all names in the $\mathcal{EL}^{++}$ TBox.

Example 3 *The $\mathcal{EL}^{++}_{\mathcal{C}}$ approximation of $\mathcal{T}_1$ in Example 1 is (T_{Koala}, CT_{Koala}), where T_{Koala} is the same as in Example 2, and CT_{Koala} contains pairs such as (C_1, nC_1), (C_2, nC_2), (C_3, nC_3), (C_4, nC_4), (C_5, nC_5), $(Plant, nPlant)$, $(VegeFood, nVegeFood)$, etc.*

Lemma 3 *For any TBox $\mathcal{T}$ and (T, CT) its $\mathcal{EL}_{\mathcal{C}}^{++}$ approximation, if $\mathcal{T}$ contains $n_{\mathcal{T}}$ terms, then $|T| \leq n_{\mathcal{T}} + |\mathcal{T}|$ and $|CT| = n_{\mathcal{T}}$, where $|T|(|\mathcal{T}|)$ is the number of axioms in $T(\mathcal{T})$ and $|CT|$ is the number of pairs in CT.*

Given an $\mathcal{EL}_{\mathcal{C}}^{++}$ transformation (T, CT), we normalise axioms of form $C \sqsubseteq D_1 \sqcap \ldots \sqcap D_n$ into $C \sqsubseteq D_1, \ldots, C \sqsubseteq D_n$, and recursively normalise role chain $r_1 \circ \ldots \circ r_n \sqsubseteq s$ with $n > 2$ into $r_1 \circ \ldots \circ r_{n-1} \sqsubseteq u$ and $u \circ r_n \sqsubseteq s$. Because C, D_i are basic concepts, this procedure can be done in linear time. In the following, we assume T to be always normalised. For convenience, we use a *complement function* $fc : CN_T \mapsto CN_T$ as: for each $A \in CN_T$, $fc(A) = B$ such that $(A, B) \in CT$.

To utilize the complementary relations in CT, we propose additional completion rules (Table 3) to $\mathcal{EL}^{++}$.

Table 3: Complement completion rules

R9	If $A, B \in S(X)$, $A = fc(B)$ and $\bot \notin S(X)$ then $S(X) := S(X) \cup \{\bot\}$
R10	If $A \in S(B)$ and $fc(B) \notin S(fc(A))$ then $S(fc(A)) := S(fc(A)) \cup \{fc(B)\}$
R11	If $A_1 \sqcap \ldots \sqcap A_i \sqcap \ldots \sqcap A_n \sqsubseteq \bot$, $A_1, \ldots, A_{i-1}$, $A_{i+1}, \ldots, A_n \in S(X)$ and $fc(A_i) \notin S(X)$ then $S(X) := S(X) \cup \{fc(A_i)\}$

R9 realises axiom $A \sqcap {\sim} A \sqsubseteq \bot$. **R10** realises $A \sqsubseteq B \rightarrow {\sim} A \sqsubseteq {\sim} B$. **R11** builds up the relations between conjuncts of a conjunction, *e.g.* $A \sqcap B \sqsubseteq \bot$ implies $A \sqsubseteq {\sim} B$.

Now we can infer $Koala \sqsubseteq Herbivore$ as follows:

- $\alpha_2 \rightarrow nC_2 \sqsubseteq nC_4 \rightarrow_{R10} C_4 \sqsubseteq C_2 \rightarrow nC_3 \sqsubseteq C_2$
- $C_3 \sqsubseteq VegeFood \rightarrow_{R10} nVegeFood \sqsubseteq nC_3$
- $nVegeFood \sqsubseteq nC_3$, $nC_3 \sqsubseteq C_2 \rightarrow nVegeFood \sqsubseteq C_2 \rightarrow nC_5 \sqsubseteq nC_1 \rightarrow_{R10} C_1 \sqsubseteq C_5 \rightarrow Koala \sqsubseteq Herbivore$

The inferences with $\rightarrow_{R10}$ are enabled by **R10**.

3.3 Cardinality-enriched $\mathcal{EL}_{\mathcal{CQ}}^{++}$ Approximation

In Def.3 we extend the $\mathcal{EL}^{++}$ transformation to support the $\neg$ construct. It is a natural question to ask whether it is possible to approximate even more non-$\mathcal{EL}^{++}$ constructs, e.g. cardinality, into $\mathcal{EL}^{++}$? In $\mathcal{EL}_{\mathcal{C}}^{++}$ approximation, a concept constructed by $\geq$ can only be represented as a fresh name. In this way, $X \sqsubseteq \bot$ can not be entailed from $\mathcal{T}_4$ in Example 4.

Example 4 $\mathcal{T}_4 = \{X \sqsubseteq \geq 4r.A, X \sqsubseteq \leq 2s.B, A \sqsubseteq B, r \sqsubseteq s\}$. *$X \sqsubseteq \bot$ should be entailed.*

This subsumption requires the relations among the filler concepts (e.g. A), the role (e.g. r) and the cardinality values (e.g 4). We maintain such relations in a *cardinality table* (QT) whose elements are tuples (A, r, n), where A denotes the filler , r the role and n the cardinality value.

Definition 4 ***(Cardinality-enriched $\mathcal{EL}_{\mathcal{CQ}}^{++}$ Transformation)*** *Given a TBox $\mathcal{T}$, a name assignment fn, let $A_{fn,\mathcal{EL}_{\mathcal{C}}^{++}}(\mathcal{T}) = (T', CT')$, its cardinality-enriched $\mathcal{EL}_{\mathcal{CQ}}^{++}$ transformation $A_{fn,\mathcal{EL}_{\mathcal{CQ}}^{++}}(\mathcal{T})$ is a tuple (T, CT, QT) constructed as follows:*

1. *T is initialised as T'.*
2. *$CT = CT'$.*
3. *QT is initialised as $\emptyset$.*
4. *for each term C that is of the form $\geq nR.D$ in $\mathcal{T}$,*
   (a) *if $n = 0$, $T = T \cup \{\top \sqsubseteq fn(C)\}$*
   (b) *if $n = 1$, $T = T \cup \{fn(C) \equiv \exists fn(R).fn(D)\}$*
   (c) *otherwise, $T = T \cup \{fn(C) \equiv fn(D)^{fn(R),n}\}$, and $QT = QT \cup \{(fn(C), fn(R), n)\}$.*
5. *for each pair of names A and r, if there exist $(A, r, i_1), (A, r, i_2), \ldots, (A, r, i_n) \in QT$ with $i_1 < i_2 < \ldots < i_n$, $T = T \cup \{A^{r,i_n} \sqsubseteq A^{r,i_{n-1}}, \ldots, A^{r,i_2} \sqsubseteq A^{r,i_1}, A^{r,i_1} \sqsubseteq \exists r.A\}$*

In step 4, $fn(D)^{fn(R),n}$ is a fresh name. For example, $nVegeFood^{eat,3}$ for $\geq 3eat.\neg VegeFood$. Similarly, $\leq nR.D$ will be approximated via the approximation of its complement $\geq (n+1)R.D$. In step 5, for each pair of name assignment A, r in T, a subsumption chain is added into T because $\geq i_n r.A \sqsubseteq \ldots \sqsubseteq \geq i_2 r.A \sqsubseteq \geq i_1 r.A \sqsubseteq \exists r.A$. We call this procedure an $\mathcal{EL}_{\mathcal{CQ}}^{++}$ *approximation*.

Proposition 4 ***($\mathcal{EL}_{\mathcal{CQ}}^{++}$ Approximation)*** *For a TBox $\mathcal{T}$, a name assignment fn, let $A_{fn,\mathcal{EL}_{\mathcal{CQ}}^{++}}(\mathcal{T}) = (T, CT, QT)$, we have T an $\mathcal{EL}^{++}$ TBox.*

This indicates that, by Def.4 a TBox can be syntactically transformed into a tuple of an $\mathcal{EL}^{++}$ TBox, a complement table and a cardinality table.

Now, in Example 4, $\mathcal{T}_4$ can be approximated into $T_4 \supseteq \{X, \sqsubseteq Y_1, Y_1 \equiv A^{r,4}, X \sqsubseteq Y_2, nY_2 \equiv B^{s,3}, A \sqsubseteq B, r \sqsubseteq s\}$ with $fn(\geq 4r.A) = Y_1$, $fn(\leq 2s.B) = Y_2$ and $fn(\geq 3s.B) = nY_2$, $CT_4 \supseteq \{(Y_1, nY_1), (Y_2, nY_2)\}$, $QT_4 \supseteq \{(A, r, 4), (B, s, 3)\}$.

Lemma 5 *For any TBox $\mathcal{T}$, let (T, CT, QT) its $\mathcal{EL}_{\mathcal{CQ}}^{++}$ transformation, if $\mathcal{T}$ contains $n_{\mathcal{T}}$ terms, then $|CN_T| \leq 2 \times n_{\mathcal{T}}$, $|T| \leq 3 \times n_{\mathcal{T}} + |\mathcal{T}|$, $|CT| = n_{\mathcal{T}}$ and $|QT| \leq n_{\mathcal{T}}$, where CN_T is the number of basic concepts in T, $|T|(|\mathcal{T}|)$ the number of axioms in $T(\mathcal{T})$, $|CT|$ the number of pairs in CT and $|QT|$ the number of tuples in QT.*

We further extend Table 3 with Table 4.

R12, in which $r \sqsubseteq_* s$ if $r = s$ or $r \sqsubseteq s \in T$, realises inference $A \sqsubseteq B, R \sqsubseteq S, i \geq j \rightarrow \geq iR.A \sqsubseteq \geq jS.B$. **R13** is the extension of **R4** and **R14-16** are extensions of **R8**. Now we can entail $X \sqsubseteq \bot$ in Example 4 as follows:

1. $A \sqsubseteq B, r \sqsubseteq s \rightarrow_{R12} A^{r,4} \sqsubseteq B^{s,3}$,
2. $A^{r,4} \sqsubseteq B^{s,3} \rightarrow X \sqsubseteq nY_2$
3. $X \sqsubseteq nY_2, X \sqsubseteq Y_2, (Y_2, nY_2) \in CT \rightarrow_{R9} X \sqsubseteq \bot$

Table 4: Cardinality completion rule

R12	If $B \in S(A)$, $(A, r, i), (B, s, j) \in QT$, $r \sqsubseteq_* s$, $i \geq j$ and $B^j \notin S(A^i)$ then $S(A^i) := S(A^i) \cup \{B^j\}$
R13	If $A^{r,i} \in S(X)$, $A' \in S(A)$, $\exists r.A' \sqsubseteq B \in T$ and $B \notin S(X)$ then $S(X) := S(X) \cup \{B\}$
R14	If $A^{r_1,i} \in S(X)$, $(A, B) \in R(r_2)$, $r_1 \circ r_2 \in T$, and $(X, B) \notin R(r_3)$ then $R(r_3) := R(r_3) \cup \{(X, B)\}$
R15	If $(X, A) \in R(r_1)$, $B^{r_2,i} \in S(A)$, $r_1 \circ r_2 \in \mathcal{T}$, and $(X, B) \notin R(r_3)$ then $R(r_3) := R(r_3) \cup \{(X, B)\}$
R16	If $A^{r_1,i} \in S(X)$, $B^{r_2,j} \in S(A)$, $r_1 \circ r_2 \in \mathcal{T}$, and $(X, B) \notin R(r_3)$ then $R(r_3) := R(r_3) \cup \{(X, B)\}$

3.4 Reasoning Properties

In this subsection, we analyze the complexity of our approximate reasoning approach.

Theorem 6 ***(Complexity)*** *For any $\mathcal{EL}^{++}_{\mathcal{CQ}}$ transformation (T, CT, QT) (T normalised), TBox reasoning by **R1**-**R16** will terminate in polynomial time w.r.t. $|CN_T|$+$|RN_T|$.*

Similarly, reasoning on the $\mathcal{EL}^{++}$ and $\mathcal{EL}^{++}_{\mathcal{C}}$ approximations also share the polynomial complexity. Note that, from Lemmas 1, 3 and 5, the approximation is always linear. To sum up, the approximation-reasoning approach is tractable.

With the approximation and corresponding rules, we can compute concept subsumptions in an $\mathcal{SROIQ}$ TBox:

Theorem 7 ***(Concept Subsumption Checking)*** *Given a TBox $\mathcal{T}$, its vocabulary $V_{\mathcal{T}}$ and $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}} = (T, CT, QT)$, for any two concepts C and D constructed from $V_{\mathcal{T}}$, if $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(\{C \sqsubseteq \top, D \sqsubseteq \top\}) = (T', CT', QT')$, then $\mathcal{T} \models C \sqsubseteq D$ if $fn(D) \in S(fn(C))$ can be computed by rules **R1-R16** on $(T \cup T', CT \cup CT', QT \cup QT')$.*

The theorem indicates that our $\mathcal{EL}^{++}_{\mathcal{CQ}}$ approximate reasoning approach is soundness-preserving. This conclusion holds similarly on $\mathcal{EL}^{++}$ and $\mathcal{EL}^{++}_{\mathcal{C}}$ approximate reasoning.

Furthermore, unsatisfiability checking of a concept C can be reduced to entailment checking of $C \sqsubseteq \bot$; ontology inconsistency checking can be reduced to entailment checking of $\top \sqsubseteq \bot$ or $\{a\} \sqsubseteq \bot$.

4 Evaluation

We implemented 3 versions of our approach, namely the $\mathcal{EL}^{++}$, $\mathcal{EL}^{++}_{\mathcal{C}}$ and $\mathcal{EL}^{++}_{\mathcal{CQ}}$ approximate reasoning in the REL reasoner, a component of our TrOWL reasoning infrastructure [1]. To evaluate their performance in practice, we compared with mainstream reasoners Pellet 2.0.0, FaCT++ 1.3.0.1 and HermiT 1.1. All experiments were conducted in an environment of Windows XP SP3 with 2.66 GHz CPU and 1G RAM allocated to JVM 1.6.0.07.

Following (Motik, Shearer, and Horrocks 2009), we examined the most difficult ontologies in the state-of-the-art DL benchmark (HermiT-Benchmark 2009). To focus on TBox reasoning, ABox axioms were removed with care[2]. Most of the remaining TBoxes can be classified easily by all the reasoners and completely by our $\mathcal{EL}^{++}_{\mathcal{C}}$ system. Results of the hard ones are shown in Table 5. We mainly conducted the evaluations on $\mathcal{EL}^{++}_{\mathcal{C}}$ system. To show the effects of complement-enriched approximate reasoning, we present also the $\mathcal{EL}^{++}$ recall. For those TBoxes for which the $\mathcal{EL}^{++}_{\mathcal{C}}$ approach was incomplete, we classified them again with the $\mathcal{EL}^{++}_{\mathcal{CQ}}$ system. Each reasoner was given 10 min to classify each ontology. We queried for subsumption relations between named concepts (including *owl:Thing* and *owl:Nothing*) and counted the numbers. Recall of REL was computed against others to measure the completeness. Thus the time figures include classification time, subsumption retrieval and counting time. Time unit is second.

Results illustrated in Table 5 show that, the efficiency of REL reasoners is in general better than all other reasoners. Even the slowest $\mathcal{EL}^{++}_{\mathcal{CQ}}$ system is faster than all main stream reasoners. Also, REL is the only reasoner that can return result on the FMA ontology.. With extension of the approximation, higher and higher recall can be achieved. $\mathcal{EL}^{++}$ is quite incomplete on some ontologies. $\mathcal{EL}^{++}_{\mathcal{C}}$ approximation can significantly improve the recall on ontologies such as Cyc and Tambis Full. With further extension to $\mathcal{EL}^{++}_{\mathcal{C}}$ approximation, all the recalls are over 99% (except FMA).

We were also interested in the scalability of our approach. Based on Table 5 we chose the 3 easiest ontologies and enlarged them by duplicating all the concept names (but keep the role names). Consequently, all the concept axioms were duplicated. We classified these ontologies using REL-$\mathcal{EL}^{++}_{\mathcal{C}}$ system, which has a nice balance between efficiency and completeness (Ref. Table 5). It performed quite stable when the quantity of data increased (Table 6). Due to the interactions between duplications through role axioms, REL even gained some recall on Wine ontology.

5 Discussions & Future Work

Approximate reasoning has been an important topic for ontology (KR) and AI research. On the one hand, expressive DLs (such as those underpinning the standard Semantic Web ontology languages) have high worst case computational complexity, making approximate reasoning an attractive way to provide scalable and efficient reasoning services (Pan and Thomas 2007). On the other hand, it has been argued that (Groot, Stuckenschmidt, and Wache 2005) while logic has always aimed at modelling idealized forms of reasoning under idealized circumstances, this is not what is required under the practical circumstances in knowledge-based systems where we also need to consider (i) reasoning under time-pressure, (ii) reasoning with other limited re-

[1] http://www.trowl.eu/

[2] ABox axioms involving individuals appearing in the TBox were converted, e.g. $a : C$ into $\{a\} \sqsubseteq C$, $a \neq b$ into $\{a\} \sqcap \{b\} \sqsubseteq \bot$, etc.. The others are removed.

Table 5: Classification time (sec) of mainstream reasoners

Ontology $\mathcal{T}$	FaCT++	HermiT	Pellet	$\mathcal{EL}^{++}$	$\mathcal{EL}_{\mathcal{C}}^{++}$		$\mathcal{EL}_{\mathcal{CQ}}^{++}$	
				recall	time	recall	time	recall
Biological Process	3.656	5.343	10.063	93.1%	1.11	100%	-	-
Cellular Component	5.872	8.077	16.966	91.9%	1.359	100%	-	-
GO	18.563	6.047	16.39	93.1%	4.203	100%	-	-
Cyc	25.531	16.853	142.89	1.2%	1.672	100%	-	-
FMA Constitutional	e/o	e/o	e/o	N/A	10.062	N/A	50.89	N/A
Tambis Full	0.375	1.063	1.343	7.2%	0.11	99.3%	0.203	100%
Wine	0.578	0.875	1.359	95.8%	0.078	96.8%	0.156	99.4%
DLP	0.219	61.948	98.024	100%	0.125	100%	-	-

Table 6: Comparison on duplicated TBox

Size	FaCT++	HermiT	Pellet	$\mathcal{EL}_{\mathcal{C}}^{++}$	Recall
Tambis Full					
5×	9.125	37.92	24.25	0.719	99.3%
10×	40.577	292.48	205.2	1.985	99.3%
20×	e/o	t/o	t/o	5.671	N/A
30×	e/o	t/o	t/o	11.624	N/A
Wine					
5×	13.784	56.85	86.66	0.641	97.7%
10×	33.01	t/o	t/o	2.188	97.9%
20×	243.496	t/o	t/o	10.077	98.0%
30×	t/o	t/o	t/o	27.529	N/A
DLP					
5×	t/o	e/o	e/o	3.39	N/A
10×	t/o	e/o	e/o	20.827	N/A
20×	t/o	e/o	e/o	142.305	N/A
30×	t/o	e/o	e/o	450.6	N/A

sources besides time and (iii) reasoning that is not perfect but instead good enough for given tasks.

In this paper, we address a long-lasting open problem; i.e, effective and efficient approximate TBox reasoning. With their negative results, Groot et al. concluded that traditional approximation method by Cadoli and Schaerf (1995) is not suited for ontology reasoning, and that new approximate strategy are needed. In this paper, we propose to combine the ideas of language weakening and approximate deduction to provide soundness preserving TBox reasoning for expressive DLs. We apply our idea to approximate OWL2-DL ontologies to $\mathcal{EL}^{++}$ ones; preliminary evaluation results showed that our approach performs effectively and efficiently on real world ontologies.

In the future we will investigate more approximation and reasoning patterns. On the basis of this study, we will investigate the completeness as we discussed in Sec.3.4 and possible approximation to Horn $\mathcal{SHIQ}$ (Kazakov 2009). We expect our work to build a bridge between expressive and tractable ontology languages (such as that between OWL2-DL and OWL2-EL).

Acknowledgements

This work has been partially supported by the European Project Marrying Ontologies and Software Technologies (EU ICT 2008-216691).

References

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Baader, F.; Brandt, S.; and Lutz, C. 2005. Pushing the $\mathcal{EL}$ Envelope. In *Proceedings IJCAI-05*.

Groot, P.; Stuckenschmidt, H.; and Wache, H. 2005. Approximating Description Logic Classification for Semantic Web Reasoning. In *ESWC-2005*.

HermiT-Benchmark. 2009. HermiT Benchmark. http://hermit-reasoner.com/2009/JAIR_benchmarks/.

Hitzler, P., and Vrandecic, D. 2005. Resolution-Based Approximate Reasoning for OWL DL. In *ISWC-2005*.

Hollunder, B.; Nutt, W.; and Schmidt-Schauß, M. 1990. Subsumption Algorithms for Concept Description Languages. In *ECAI-90*, 348–353. Pitman Publishing.

Horrocks, I.; Kutz, O.; and Sattler, U. 2006. The Even More Irresistible SROIQ. In *KR 2006*.

Kazakov, Y. 2008. SRIQ and SROIQ are Harder than SHOIQ. In *DL 2008*.

Kazakov, Y. 2009. Consequence-Driven Reasoning for Horn SHIQ Ontologies. In *IJCAI 2009*.

Motik, B.; Shearer, R.; and Horrocks, I. 2009. Hypertableau Reasoning for Description Logics. *Journal of Artificial Intelligence Research*.

Pan, J. Z., and Thomas, E. 2007. Approximating OWL-DL Ontologies. In *AAAI-2007*, 1434–1439.

Schaerf, M., and Cadoli, M. 1995. Tractable Reasoning via Approximation. *Artificial Intelligence* 74:249–310.

Stuckenschmidt, H., and van Harmelen, F. 2002. Approximating Terminological Queries. In *FQAS '02*.

Tsarkov, D., and Horrocks, I. 2004. Efficient Reasoning with Range and Domain Constraints. In *(DL 2004)*.

Tsarkov, D.; Horrocks, I.; and Patel-Schneider, P. F. 2007. Optimizing Terminological Reasoning for Expressive Description Logics. *J. Autom. Reason.* 39(3):277–316.

Wache, H.; Groot, P.; and Stuckenschmidt, H. 2005. Scalable Instance Retrieval for the Semantic Web by Approximation. In *WISE Workshops-05*.