Completeness Guaranteed Approximation for OWL DL Query Answering

Jeff Z. Pan, Edward Thomas and Yuting Zhao

Dept. of Computing Science, University of Aberdeen
King's College, Aberdeen AB24 3FX, UK

Abstract. How to provide scalable and quality guaranteed approximation for query answering over expressive description logics (DLs) is an important problem in knowledge representation (KR). This is a pressing issue, in particular due to the fact that, for the widely used standard Web Ontology Language OWL, whether conjunctive query answering is decidable is still an open problem. Pan and Thomas propose a soundness guaranteed approximation, which transforms an ontology in a more expressive DL to its least upper bound approximation in a tractable DL. In this paper, we investigate a completeness guaranteed approximation, based on transformations of both the source ontology and input queries. We have implemented both soundness guaranteed and completeness guaranteed approximations in our TrOWL ontology reasoning infrastructure.

1 Introduction

The growing availability of semantic annotated data requires scalable query answering in description logics. How to provide efficient querying answering service for expressive DLs has been an important open problem in KR. In fact, whether query answering in OWL DL is decidable is still an open problem. The closest available complexity results are about the $\mathcal{SHIQ}$ DL, which is OWL DL without nominals and datatypes, but allowing qualified number restrictions. Ortiz et al. [7] have shown the co-NP-complete data complexity result for query answering in $\mathcal{SHIQ}$, with a restriction that transitive properties and properties with transitive sub-properties are disallowed in queries. Glimm et al. [1] have further provided the co-NP-complete data complexity and 2EXPTIME combined complexity results for general query answering in $\mathcal{SHIQ}$.

Approximation has been identified as a potential way to reduce the complexity of reasoning over OWL DL ontologies. Many existing approaches [10, 12, 3, 2, 5] are mainly based on syntactic approximation of ontological axioms and queries. All these approaches could introduce unsound answers. Pan and Thomas [8] propose a soundness guaranteed approximation, which transforms an ontology in a more expressive DL to its least upper bound approximation in a tractable DL. To the best of our knowledge, there is no published scalable completeness preserving approximation approaches for query answering in OWL DL.

In this paper, we investigate a completeness guaranteed approximation for non-boolean query answering, based on transformations of both the source ontology and input queries It turns out that the semantic approximation constructed for soundness guaranteed query service can also be exploited to provide completeness guaranteed query service. We then provide a more fine grained approximation, which can provide potentially much smaller but still completeness guaranteed, which can be used to provide anytime querying answering for OWL DL. We have implemented both soundness guaranteed and completeness guaranteed approximations in our TrOWL ontology reasoning infrastructure. Accordingly, for each query, we provide two answer sets: one is soundness guaranteed and the other is completeness guaranteed. To evaluate our approach, we use the ontologies that Motik et al [6] used for evaluating ABox reasoning, and extend the tested queries by allowing non-distinguished variables. Our evaluation shows: (1) Both the soundness guaranteed and completeness guaranteed answer sets can be computed efficiently. (2) Our anytime algorithm effectively produces smaller completeness guaranteed answer set. (3) For every query in our evaluation, at least one of our completeness guaranteed answer sets is both sound and complete with respect to the reference answer for that query.

2 Preliminary

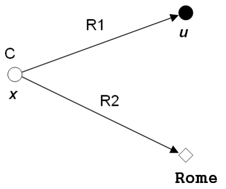
2.1 Conjunctive Query Answering

A *conjunctive query* is of the form $q(X) \leftarrow \exists U.\varphi(X,U)$, or simply $q(X) \leftarrow \varphi(X,U)$, where X and U are vectors of distinguished variables (DVs) and non-distinguished variables (NDVs) resp., and φ is a conjunction of atoms of the form $\mathsf{A}(v)$, $R(v_1, v_2)$, where A, R are *named* concepts and *named* roles resp., $v,\ v_1$ and v_2 are variables in X and U, or individual names in the given ontology. If $v,\ v_1$ and v_2 are *not* in U, then $\mathsf{A}(v)$ and $R(v_1, v_2)$ are non-distinguished variable free atoms. If X is an empty set, we say q is a boolean query; otherwise, we say q is a non-boolean query. Theoretically, allowing only named concepts and roles in atoms is not a restriction, as we can always define such named concepts and roles in ontologies. Practically, this should not be an issue as querying against *named* relations is a usual practice when people query over relational databases. In this paper, although we consider input queries with only named concepts and roles in atoms, it is still possible to have concept descriptions in atoms due to query rewriting (see Section 3). As usual, an interpretation $\mathcal{I}$ satisfies an ontology $\mathcal{O}$ if it satisfies all the axioms in $\mathcal{O}$; in this case, we say $\mathcal{I}$ is a model of $\mathcal{O}$. Given an evaluation $[X \mapsto S]$, where S is a vector of individual names, if every model $\mathcal{I}$ of $\mathcal{O}$ satisfies $q_{[X \mapsto S]}$, we say $\mathcal{O}$ entails $q_{[X \mapsto S]}$; in this case, S is called a *solution* of q.

We could consider a conjunctive query as a directed graph [4], where the nodes are variable or individual names. In addition, concept and role terms provide labels for nodes and edges respectively. Note that the direction of the edge is related to role label; namely, $R(i,j)$ (i connects to j via R) is equivalent to $R^-(j,i)$ (j connects to i via R^-). Therefore, it is enough to consider weak connectivity of query graphs. Without loss of generality, we assume input queries corresponding to *weakly connected graphs*;[1] a graph is a weakly connected if replacing all its directed edges with undirected edges produces a connected (undirected) graph. The weak connectivity degree of a node is

[1] Unconnected components do not share variables, therefore they can be considered independently to each other.

the number of nodes that weakly connect to it. A *path* in a query graph is a sequence of nodes such that each of its nodes weakly connects to the next node in the sequence. The length of the path is $n - 1$, where n is the number of nodes in the path. For the readers' convenience the non-distinguished variables are represented by filled circles (●), distinguished variables by unfilled circles (○) and individuals by diamonds (◇). For example, the query $q_1(x) \leftarrow C(x) \wedge R1(x, u) \wedge R2(x, \texttt{Rome})$ corresponds to the following graph.



A non-distinguished variable sub-graph (or NDV sub-graph) of a query q is a sub-graph of q with all its nodes being non-distinguished variables. For example, q_1 has one NDV sub-graph (containing the node u).

2.2 Knowledge Compilation

Selman and Kautz illustrated the idea of knowledge compilation in [9] by showing how a propositional theory can be compiled into a tractable form consisting of a set of Horn clauses. As a logically equivalent set of Horn clauses does not always exist, they proposed to use Horn lower-bound and Horn upper-bound to approximate the original theory.

Let Σ be a set of clauses (the original theory), the sets Σ_{lb} and Σ_{ub} of Horn clauses are respectively a Horn lower-bound and a Horn upper-bound of Σ iff $\mathcal{M}(\Sigma_{\mathrm{lb}}) \subseteq \mathcal{M}(\Sigma) \subseteq \mathcal{M}(\Sigma_{\mathrm{ub}})$, or, equivalently, $\Sigma_{\mathrm{lb}} \models \Sigma \models \Sigma_{\mathrm{ub}}$. This says the Horn lower-bound is logically stronger than the original theory and the Horn upper-bound is logically weaker than it. A Horn lower-bound Σ_{glb} is a greatest Horn lower-bound iff there is no set Σ' of Horn clauses such that $\mathcal{M}(\Sigma_{\mathrm{lb}}) \subset \mathcal{M}(\Sigma') \subseteq \mathcal{M}(\Sigma)$. A Horn upper-bound Σ_{lub} is a least Horn upper-bound iff there is no set Σ' of Horn clauses such that $\mathcal{M}(\Sigma) \subseteq \mathcal{M}(\Sigma') \subset \mathcal{M}(\Sigma_{\mathrm{lub}})$.

2.3 Semantic Approximation

Pan and Thomas [8] apply the idea of knowledge compilation on semantically approximating a source ontology $\mathcal{O}_s$ in a more expressive DL $\mathcal{L}_s$ (source language) with its (least) upper-bound $\mathcal{O}_t$ in a less expressive DL $\mathcal{L}_t$ (target language). This sub-section summaries the results from [8].

The following definition provides the notion of least upper-bound in their setting; i.e., they consider all $\mathcal{L}_t$ axioms that are entailed by $\mathcal{O}_s$.

Definition 1. ***(Entailment Set)*** *Let $\mathbf{N}_C$, $\mathbf{N}_P$ and $\mathbf{N}_I$ be the finite set of named concepts, named roles and named individuals, respectively, used in $\mathcal{O}_s$. The* entailment set *of $\mathcal{O}_s$ w.r.t. $\mathcal{L}_t$, denoted as $\mathbf{ES}(\mathcal{O}_s, \mathcal{L}_t)$, is the set which contains* all *$\mathcal{L}_t$ axioms (constructed by using only vocabulary in $\mathbf{N}_C$, $\mathbf{N}_P$ and $\mathbf{N}_I$) that are entailed by $\mathcal{O}_s$.*

Lemma 1. *$\mathbf{ES}(\mathcal{O}_s, \mathcal{L}_t)$ is the least upper bound compilation of $\mathcal{O}_s$ in $\mathcal{L}_t$.*

In order to use $\mathbf{ES}(\mathcal{O}_s, \mathcal{L}_t)$ as the target approximation ontology $\mathcal{O}_t$, we need to find some lightweight language $\mathcal{L}_t$ such that $\mathbf{ES}(\mathcal{O}_s, \mathcal{L}_t)$ is finite.

Lemma 2. *Given an OWL DL ontology $\mathcal{O}_s$, $\mathbf{ES}(\mathcal{O}_s, \mathcal{L}_{DL\text{-}LiteR})$ is a finite set.*

Accordingly, we can use DL-LiteR as a lightweight language $\mathcal{L}_t$ for approximating OWL DL ontologies in order to provide scalable query answering service. See [8] for how to compute $\mathbf{ES}(\mathcal{O}_s, \mathcal{L}_{\text{DL-LiteR}})$ from $\mathcal{O}_s$.

Given an OWL DL ontology $\mathcal{O}_s$ and an arbitrary query q, we denote the set of solutions of q over $\mathcal{O}_s$ as $\mathbf{S}_{q,\mathcal{O}_s}$ and the set of solutions of q over $\mathbf{ES}(\mathcal{O}_s, \text{DL-LiteR})$ as $\mathbf{S}_{q,\mathbf{ES}(\mathcal{O}_s,\text{DL-LiteR})}$. The following theorem shows that query answering based on the DL-LiteR entailment set is soundness guaranteed.

Theorem 1. *Given an OWL DL ontology $\mathcal{O}_s$ and an arbitrary query q, $\mathbf{S}_{q,\mathbf{ES}(\mathcal{O}_s,DL\text{-}LiteR)} \subseteq \mathbf{S}_{q,\mathcal{O}_s}$.*

For queries without non-distinguished variables, it is both soundness and completeness guaranteed.

Theorem 2. *Given an OWL DL ontology $\mathcal{O}_s$ and an arbitrary query q' that contains no non-distinguished variables, $\mathbf{S}_{q',\mathbf{ES}(\mathcal{O}_s,DL\text{-}LiteR)} = \mathbf{S}_{q',\mathcal{O}_s}$.*

3 Completeness Guaranteed Approximations

In this section, we will first show how to make use of entailment sets (introduced in Section 2.3) to provide a completeness guaranteed approximation for OWL DL query answering. Secondly, we will show how to provide smaller completeness guaranteed approximations by using some enriched entailment sets.

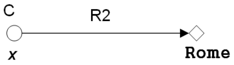
3.1 Approximation Based on Entailment Sets

In order to provide a completeness guaranteed approximation based on **entailment sets**, we first introduce the approximate function $\mathtt{F}_{C0}$, and then show that, given an OWL DL ontology $\mathcal{O}_s$ and a query q, querying $\mathtt{F}_{C0}(q)$ over $\mathbf{ES}(\mathcal{O}_s, \text{DL-LiteR})$ is completeness guaranteed, i.e. $\mathbf{S}_{q,\mathcal{O}_s} \subseteq \mathbf{S}_{\mathtt{F}_{C0}(q),\mathbf{ES}(\mathcal{O}_s,\text{DL-LiteR})}$.

Definition 2. *(**Approximation Function** $\mathtt{F}_{C0}$) Let q be an input non-boolean conjunctive query of the form $q(X) \leftarrow \varphi(X, U)$, the approximation function $\mathtt{F}_{C0}$ returns $q_{C0}(X) \leftarrow \varphi'(X, \emptyset)$, where $\varphi'(X, \emptyset)$ is a non-distinguished variable free conjunction that only contains all the non-distinguished variable free atoms in $\varphi(X, U)$.*

From the query graph point of view, this amounts to removing all NDV sub-graphs of an input query q and all edges connecting to these sub-graphs from q.

Example 1. Let us revisit the query $q1(x) \leftarrow C(x) \wedge R1(x, u) \wedge R2(x, \mathtt{Rome})$, $\mathtt{F}_{C0}(q1)$ is $q1_{C0}(x) \leftarrow C(x) \wedge R2(x, \mathtt{Rome})$, which corresponds to the following graph.



Theorem 3. *Given an OWL DL ontology $\mathcal{O}_s$ and an arbitrary non-boolean query q, $\mathbf{S}_{q,\mathcal{O}_s} \subseteq \mathbf{S}_{\mathtt{F}_{C0}(q),\mathbf{ES}(\mathcal{O}_s,\text{DL-LiteR})}$.*

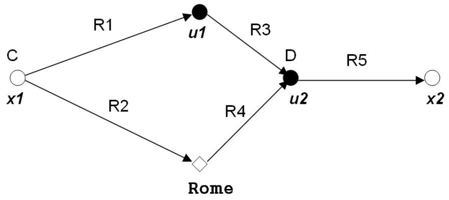
Proof: (Sketch) Immediate consequence of (i) $\mathbf{S}_{q,\mathcal{O}_s} \subseteq \mathbf{S}_{\mathtt{F}_{C0}(q),\mathcal{O}_s}$, due to the definition of $\mathtt{F}_{C0}$, and (ii) $\mathbf{S}_{\mathtt{F}_{C0}(q),\mathcal{O}_s} = \mathbf{S}_{\mathtt{F}_{C0}(q),\mathbf{ES}(\mathcal{O}_s,\text{DL-LiteR})}$, due to Theorem 2. ■

Theorems 1 and 3 suggest that, given an OWL DL ontology $\mathcal{O}_s$ and a non-boolean query q, we could now provide *both* a soundness guaranteed answer set $\mathbf{S}_{q,\mathbf{ES}(\mathcal{O}_s,\text{DL-LiteR})}$ *and* a completeness guaranteed answer set $\mathbf{S}_{\mathtt{F}_{C0}(q),\mathbf{ES}(\mathcal{O}_s,\text{DL-LiteR})}$.

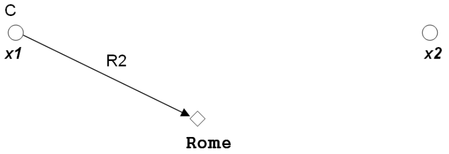
3.2 Towards Fine Grained Approximations

The approximation function $\mathtt{F}_{C0}$ removes all the non-distinguished atoms from the input query q, which could potentially introduce many unsound answers in the completeness guaranteed answer set.

Example 2. Consider the cyclic query $q2$ of the form $q2(x1, x2) \leftarrow C(x1) \wedge R1(x1, u1) \wedge R2(x1, \mathtt{Rome}) \wedge R3(u1, u2) \wedge R4(\mathtt{Rome}, u2) \wedge D(u2) \wedge R5(u2, x2)$, which corresponds to the following graph:



$\mathtt{F}_{C0}(q2)$ is $q2_{C0}(x1, x2) \leftarrow R2(x1, \mathtt{Rome}) \wedge \top(x2)$, which corresponds to the following graph:

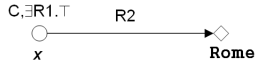


$\mathtt{F}_{C0}(q2)$ has two disconnected components that do not share any variables: $R2(x1, \mathtt{Rome})$ and $\top(x2)$. The latter one binds *all* named individuals to $x2$, thus the answer set potentially could be large and contain many unsound answers.

In this section, we investigate how to improve $\mathtt{F}_{C0}$ by keeping more information from the non-distinguished atoms. We develop a query rewriting technique based on the rolling up technique that has been used [4] to help reduce the problem of answering *boolean queries with acyclic query graphs* to the problem of knowledge base satisfiability checking.

Let us revisit the query $q1$ (with an acyclic query graph) before providing formal analysis.

Example 3. The query $q1$ is to retrieve all named individuals (x), which are instances of C, related by role $R1$ to an (possibly unnamed) individual (u) and related by role $R2$ to the individual Rome. The query can be paraphrased as the query $q1'(x) \leftarrow C(x) \wedge \exists R1.\top(x) \wedge R2(x, \mathtt{Rome})$, which corresponds to the following graph.



It should be noted that the intuition from the above example is substantiated by the fact $q1$ corresponds to the first order logic formula $\forall x(\exists u(C(x) \wedge R1(x,u) \wedge R2(x, \mathtt{Rome})))$, which can be translated to the first order logic formula $\forall x(C(x) \wedge \exists R1.\top(x) \wedge R2(x, \mathtt{Rome}))$. Accordingly, we have the the following lemma for rolling-up a path of non-distinguished variables.

Lemma 3. *Let $\mathcal{O}_s$ be an OWL DL ontology, $C_1, ...C_m$ concepts, $R_1, ...R_m$ named roles and $\mathtt{o}$ a named individual in $\mathcal{O}_s$, $x, u_1, ..., u_m$ variables. Given the following input query q and corresponding rolled up query q':*

- $q(x) \leftarrow R(x, u_1) \wedge S(\mathtt{o}, u_1) \wedge C_1(u_1) \wedge R_2(u_1, u_2) \wedge C_2(u_2) \wedge ... \wedge R_m(u_{m-1}, u_m) \wedge C_m(u_m)$
- $q'(x) \leftarrow \exists R.(C_1 \sqcap \exists R_2.(...\exists R_{m-1}.(C_{m-1} \sqcap \exists R_m.C_m)))(x) \wedge \exists S.(C_1 \sqcap \exists R_2.(...\exists R_{m-1}.(C_{m-1} \sqcap \exists R_m.C_m)))(\mathtt{o}) \wedge \exists R.\exists S^-.\{\mathtt{o}\}(x) \wedge \exists S.\exists R^-.\top(\mathtt{o})$

we have $\mathbf{S}_{q,\mathcal{O}_s} = \mathbf{S}_{q',\mathcal{O}_s}$.

A few remarks for the above lemma: (i) If we have $A_{i,1}(u_i) \wedge ... \wedge A_{i,n}(u_i)$ in the query q, we could first combine them into one atom $A_{i,1} \sqcap ... \sqcap A_{i,n}(u_i)$, before applying the lemma. Hence, C_i can be either a named concept or a conjunction of named concepts $A_{i,1} \sqcap ... \sqcap A_{i,n}$. (ii) The lemma shows that rolling up non-distinguished variables to distinguished variables (x) is the similar to rolling up to individuals ($\mathtt{o}$). For example, we could roll up $q3(x) \leftarrow R1(x, u1) \wedge R3(u1, u2)$ into $q3'(x) \leftarrow \exists R1.(\exists R3.\top)(x)$.

In order to apply Lemma 3 for completeness guaranteed approximations of **potentially cyclic** non-boolean queries, we introduce the notion of proxy nodes. Intuitively speaking, proxy nodes are nodes to which we roll up paths of non-distinguished variable.

Definition 3. ***(Proxy Node)*** *Given a non-boolean query q, a proxy node of q is a node in the query graph of q that (i) corresponds to a distinguished variable or an individual and (ii) directly connects via a role to a non-distinguished variable.*

For example, in the query graph of $q2$, proxy nodes include $x1, x2$ and Rome.

Given a non-boolean query q, a proxy node p of q, we call the *acyclic* path $p, n_1, ...n_h$ (where $n_1, ..., n_h$ are nodes in q) a **proxy path** w.r.t. p. Note that the above definition does not take into account the direction of edges, since $R(u_i, u_j)$ is equivalent to $R^-(u_j, u_i)$. We use $\mathtt{Path}(p, n)$ to denote all proxy paths from the proxy node p to the node n. For example, in the query graph of $q2$, $\mathtt{Path}(x1, \mathtt{Rome})$ contains one path: $x1, u1, u2, \mathtt{Rome}$ w.r.t. $x1$.

The following lemma deals with enriching the labels of proxy nodes based on their non-distinguished variable paths.

Lemma 4. *Let $\mathcal{O}_s$ be an OWL DL ontology, $C_1, ...C_m$ concepts and $R_1, ...R_m$ named roles in $\mathcal{O}_s$, X the set of distinguished variables, p a proxy node in q and $n_1, ..., n_m$ nodes in q. Given the following input query q and corresponding enriched query q' based on the rolling up of the proxy path $p, n_1, ..., n_m$:*

- $q(X) \leftarrow D(p) \wedge R_1(p, n_1) \wedge C_1(n_1) \wedge R_2(n_1, n_2) \wedge C_2(n_2) \wedge ... \wedge R_m(n_{m-1}, n_m) \wedge C_m(n_m)$
- $q'(X) \leftarrow D(p) \wedge R_1(p, n_1) \wedge C_1(n_1) \wedge R_2(n_1, n_2) \wedge C_2(n_2) \wedge ... \wedge R_m(n_{m-1}, n_m) \wedge C_m(n_m) \wedge \exists R_1.(C_1 \sqcap \exists R_2.(...\exists R_{m-1}.(C_{m-1} \sqcap \exists R_m.C_m)))(p)$

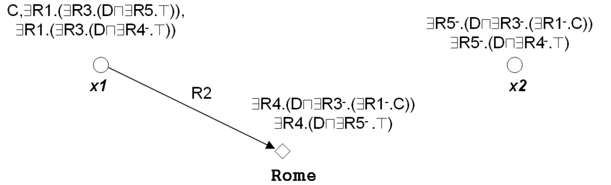
we have $\mathbf{S}_{q,\mathcal{O}_s} = \mathbf{S}_{q',\mathcal{O}_s}$.

It should be noted that q' contains all atoms of q, while adding an atom for p based on the rolling up.

Definition 4. ***(Proxy Node Based Rolling Up Function*** $\mathtt{F}_R$*) Let q be an input non-boolean query of the form $q(X) \leftarrow \varphi(X, U)$, $p_1, ..., p_n$ proxy nodes in q, the proxy node based rolling up function $\mathtt{F}_R$ rewrite q as follows:*

1. *Normalise q into q': transform concept atoms of the form $A_1(i) \wedge ... \wedge A_k(i)$ into $A_1 \sqcap ... \sqcap A_k(i)$.*
2. *Enrich q' into q'': For each NDV-subgraph g of q',*
   - *if g weakly connects to at least two proxy nodes $p_1, ...p_k (k \geq 2)$, then for each pair $\langle p_i, p_j \rangle$ $(1 \leq i < j \leq k)$ of the weakly connected proxy nodes, enrich q' based on all paths in $\mathtt{Path}(p_i, p_j)$ according to Lemma 4;*
   - *if there is only one proxy node p connected to g, let $n_1, ...n_s$ be the set of nodes in g that has the lowest weak connectivity degree. For each n_h $(1 \leq h \leq s)$, enrich q' based on all paths in $\mathtt{Path}(p, n_h)$ according to Lemma 4.*
3. *Returns $\mathtt{F}_{C0}(q'')$.*

Example 4. $q2$ contains only one NDV sub-graph, which connects to three proxy nodes. $\mathtt{F}_R$(q2) is $q2_R(x1, x2) \leftarrow C(x1) \wedge R2(x1, \mathtt{Rome}) \wedge \exists R1.(\exists R3.(D \sqcap \exists R5.\top))(x1) \wedge \exists R5^-.(D \sqcap \exists R3^-.(\exists R1^-.C))(x2) \wedge \exists R1.(\exists R3.(D \sqcap \exists R4^-.\top))(x1) \wedge \exists R4.(D \sqcap \exists R3^-.(\exists R1^-.C))(\mathtt{Rome}) \wedge \exists R5^-.(D \sqcap \exists R4^-.\top)(x2) \wedge \exists R4.(D \sqcap \exists R5^-.\top)(\mathtt{Rome})$, which corresponds to the following graph.



Theorem 4. *Given an OWL DL ontology $\mathcal{O}_s$ and an arbitrary non-boolean query q,* $\mathbf{S}_{q,\mathcal{O}_s} \subseteq \mathbf{S}_{\mathtt{F}_R(q),\mathcal{O}_s} \subseteq \mathbf{S}_{\mathtt{F}_{C0}(q),\mathbf{ES}(\mathcal{O}_s,\textit{DL-LiteR})}$.

Theorem 4 shows that $\mathbf{S}_{\mathtt{F}_R(q),\mathcal{O}_s}$ is a more fine grained completeness guaranteed answer set.

3.3 Approximation Based on Enriched Entailment Sets

Given an OWL DL ontology $\mathcal{O}_s$ and a non-boolean query q, this sub-section investigates how to to answer $\mathtt{F}_R(q)$ over $\mathcal{O}_s$. In particular, we will show this can be done based on enriched entailment sets of $\mathcal{O}_s$.

According to Def 4, these descriptions are of the form

$$\exists R_1.(C_1 \sqcap \exists R_2.(...\exists R_{m-1}.(C_{m-1} \sqcap \exists R_m.C_m))) \quad (1)$$

where $C_1, ...C_m$ are conjunctions of named concepts, and $R_1, ...R_m$ are either named roles or their inverse. Intuitively, we first introduce fresh named concepts to represent concept descriptions in the labels of proxy nodes, then query against the extended ontology. Formally, given an ontology $\mathcal{O}_s$ and a non-boolean query q, we can extend $\mathcal{O}_s$ to *q-enriched ontology* $\mathcal{O}_s^q$ as follows: for each proxy node concept description P of the form (1) in $\mathtt{F}_R(q)$, add an axiom $A_P \equiv P$ into $\mathcal{O}_s$, where A_P is a fresh named concept. We use $\mathtt{F}_N(\mathtt{F}_R(q))$ to denote the resulted query of rewriting $\mathtt{F}_R(q)$ by replacing all concept descriptions P of the form (1) with the corresponding named concepts A_P.

Lemma 5. *Let $\mathcal{O}_s$ be an ontology, q a non-boolean query.* $\mathbf{S}_{\mathtt{F}_R(q),\mathcal{O}_s} = \mathbf{S}_{\mathtt{F}_N(\mathtt{F}_R(q)),\mathcal{O}_s^q} = \mathbf{S}_{\mathtt{F}_N(\mathtt{F}_R(q)),\mathbf{ES}(\mathcal{O}_s^q,\textit{DL-LiteR})}$.

Proof: (Sketch) The first equivalence is trivial. The second equivalence is due to Theorem 2 and the fact that $\mathtt{F}_R$(q) does not contain any non-distinguished variables. ■

We call $\mathbf{ES}(\mathcal{O}_s^q, \text{DL-LiteR})$ an enriched entailment set of $\mathcal{O}_s$ w.r.t. the query q. In the next section, we will investigate query independent enriched entailment sets.

4 Anytime Reasoning Approximation

In this section, we introduce a strategy to incrementally construct enriched entailment sets. For a concept description P of the form (1), we use $depth(P)$ to denote the maximal number of serial existential quantifiers in it; e.g., $depth(\exists R_1.(C_1 \sqcap \exists R_2.\top)) = 2$. We define the depth of a query q as the maximal length of proxy paths in it.

For an OWL DL ontology $\mathcal{O}_s$, let $\mathcal{E}_i = \{e_{i1}, \cdots, e_{ik}\}$ $(i \geq 1)$ be the set of DL-LiteR axioms that a) contain some of the representative named concepts for the concept descriptions of the form (1) that are with depth i, and b) are entailed by $\mathcal{O}_s$. Accordingly, a serial of *i-entailment set* Π_i $(i \geq 1)$, are defined as followings: $\Pi_0 = \mathbf{ES}(\mathcal{O}_s, \text{DL-LiteR})$, Π_1 = $\Pi_0 \cup \mathcal{E}_1$, ..., Π_i = $\Pi_{i-1} \cup \mathcal{E}_i$. Let $0 \leq i \leq m$ be an integer, we define the function $\mathtt{F}_i$ to rewrite $\mathtt{F}_R(q)$ by a) replacing all concept descriptions $\exists R_1.(C_1 \sqcap \exists R_2.(...\exists R_{m-1}.(C_{m-1} \sqcap \exists R_m.C_m)))$ with the representative named concept (of depth i) for $\exists R_1.(C_1 \sqcap \exists R_2.(...\exists R_{i-1}.(C_{i-1} \sqcap \exists R_i.\top)))$. For example, $\mathtt{F}_1(\mathtt{F}_R(q2))$ is $q2_R(x1, x2) \leftarrow C(x1) \wedge R2(x1, \mathtt{Rome}) \wedge A_{\exists R1.\top}(x1) \wedge A_{\exists R5^-.\top}(x2) \wedge A_{\exists R4.\top}(\mathtt{Rome})$.

Theorem 5. *Given an OWL DL ontology $\mathcal{O}_s$ and a query q over $\mathcal{O}_s$. Let $j \geq i \geq 0$, we have:*

1 $\mathbf{S}_{\mathtt{F}_0(\mathtt{F}_R(q)),\Pi_0} = \mathbf{S}_{\mathtt{F}_{C0}(q),\Pi_0}$;

Table 1. Ontologies used and time required to calculate C_i (in minutes)

Ontology	C_1	C_2	C_3
SEMINTEC_1	11	47	119
VICODI_1	16	94	422
WINE_4	14	46	342
LUBM_1	17	57	421

2 $\mathbf{S}_{\mathtt{F}_i(\mathtt{F}_R(q)),\Pi_i} \supseteq \mathbf{S}_{\mathtt{F}_j(\mathtt{F}_R(q)),\Pi_j} \supseteq \mathbf{S}_{\mathtt{F}_R(q),\mathcal{O}_s} \supseteq \mathbf{S}_{q,\mathcal{O}_s}$.

Theorem 6. *Let $\mathcal{O}_s$ be an ontology, q a non-boolean query of depth i, $\mathbf{S}_{\mathtt{F}_R(q),\mathcal{O}_s} = \mathbf{S}_{\mathtt{F}_i(\mathtt{F}_R(q)),\Pi_i}$.*

It should be noted that the above theorems are for arbitrary queries. For OWL DL ontology $\mathcal{O}_s$ and a query q, an *anytime reasoning for query answering* is a set of computing jobs $\{\mathbf{S}_{\mathtt{F}_0(\mathtt{F}_R(q)),\Pi_0}, \cdots, \mathbf{S}_{\mathtt{F}_i(\mathtt{F}_R(q)),\Pi_i}, \cdots\}$. Fig. 1 shows the intuitive hierarchy of the set of solutions in anytime reasoning for query q on $\mathcal{O}_s$. We note when i increases, the curve is approaching to the middle red line, which is $\mathbf{S}_{q,\mathcal{O}_s}$.

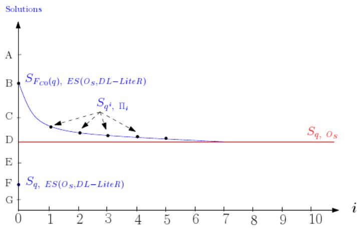


Fig. 1. Hierarchy of the set of solutions for query q on $\mathcal{O}_s$.

5 Implementation and Evaluation

We have implemented both soundness guaranteed and completeness guaranteed approximations in our TrOWL ontology reasoning infrastructure, which is based on the infrastructure used in the ONTOSEARCH2 system. As Theorem 5 indicates, soundness guaranteed and completeness guaranteed semantic approximations can be implemented in a similar manner. The main difference is that, the form one is based on Π_0, while the latter one is based on $\Pi_i(\geq 1)$. All these (i-)entailment sets are stored in the database. As it could be quite time consuming to compute $\Pi_i(\geq 1)$, we apply some optimisations and produce some relaxed completeness guaranteed approximations C_i (instead of Π_i). All the tests were done on an Apple Macbook, with 2.0 Ghz Dual Core and 2Gb ram.

In [6], Motik et al used four ontologies to evaluate the performance of ABox answering across several different reasoning systems. To evaluate the performance of our

Table 2. Source Queries

Ontology	Query
SEMINTEC	$q(x, y, z) \leftarrow Man(x) \wedge isCreditCardOf(x, y) \wedge Gold(y) \wedge livesIn(x, z) \wedge Region(z)$
VICODI	$q(x, y, z) \leftarrow Military - Person(x) \wedge hasRole(y, x) \wedge related(x, z)$
WINE	$q(x, y, z) \leftarrow Winery(x) \wedge producesWine(x, y) \wedge locatedIn(y, z)$
LUBM	$q(x, y, z) \leftarrow Student(x) \wedge Faculty(y) \wedge Course(z) \wedge advisor(x, y) \wedge takesCourse(x, z) \wedge teacherOf(y, z)$

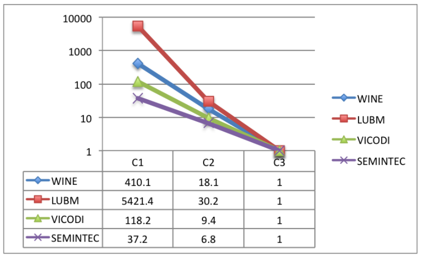
Table 3. Results of Queries

		Pellet		$\mathit{\Pi}_0$		C_1		C_2		C_3	
Ontology	NDVs	cd	t (ms)	cd	t (ms)	cd	t (ms)	cd	t (ms)	cd	t (ms)
WINE	2	0.45	981	0.57	121	410.1	388	18.1	297	1.0	288
	1	0.86	922	0.86	132	22.8	224	1.0	224	1.0	236
	0	1.0	1015	1.0	127	1.0	152	1.0	163	1.0	174
LUBM	2	0.68	1593	1.0	189	5421.4	1068	30.2	401	1.0	396
	1	0.94	1499	1.0	210	168.1	521	1.0	282	1.0	348
	0	1.0	1464	1.0	193	1.0	243	1.0	264	1.0	288
VICODI	2	0.31	1202	0.8	225	118.2	294	9.4	358	1.0	423
	1	0.97	1320	1.0	222	14.4	258	1.0	301	1.0	387
	0	1.0	1278	1.0	237	1.0	252	1.0	287	1.0	305
SEMINTEC	2	0.45	2022	0.69	182	37.2	389	6.8	356	1.0	410
	1	0.67	2121	1.0	180	4.5	286	1.0	266	1.0	339
	0	1.0	2221	1.0	195	1.0	212	1.0	234	1.0	258

query tool we will use the same ontologies, but we change the queries to include non-distinguished variables. Table 1 lists the four ontologies, together with the time spent on computing $C1 - C3$, which only needs to be computed once. In [6], each ontology is queried twice, a simple instance retrieval query, and a more complex query containing three distinguished variables. We have modified these complex queries to produce queries with zero, one, or two non-distinguished variables, so that each combination of distinguished and non-distinguished variable can be tested.

Table 3 presents the average completeness degrees (cd, with cd $= 1.0$ indicating being the exact answer set) and average time (ms) spent on querying each set of queries (one with zero NDVs, three with one, and three with two) of over $\mathit{\Pi}_0, C_1, ..., C_3$. Completeness degree is defined as $\frac{|ta|}{|ea|}$, where ta is the total answers returned, while ea is the exact answer set. Since all the queries were tree-shape queries, the exact result set was calculated semi-automatically. For comparison, we also list the average completeness degrees and average time (querying over the original ontology) spent for Pellet, which treats all variables in the query as distinguished when query answering over ontologies.

Our evaluation shows (see also Table 3): (1) Query with Pellet and over $\mathit{\Pi}_0$ are both soundness preserving; moveover, our soundness preserving approximation based on $\mathit{\Pi}_0$ is more complete than Pellet. (2) Querying over $\mathit{\Pi}_0$ is much more efficient than Pellet. (3) Querying over C_1, C_2, C_3 is also more efficient than over Pellet, while slightly less efficient than over $\mathit{\Pi}_0$. (4) The refinement from of the completeness preserving approximations is effective. The completeness preserving answer sets over C_3 are much smaller than those over C_2, which in turn are much smaller than those over C_1. In fact,



	C1	C2	C3
WINE	410.1	18.1	1
LUBM	5421.4	30.2	1
VICODI	118.2	9.4	1
SEMINTEC	37.2	6.8	1

Fig. 2. Graph of results for each complete entailment set for queries with two non-distinguished variables

for all the tested queries, answer sets over C_3 are both sound and complete. Figure 2 shows the convergence of each complete entailment set, as the complexity of the set increases. This shows the result for each query using two non-distinguished variables. It can clearly be seen how using anytime reasoning over increasingly large entailment sets can improve the precision of a query, as the additional entailment sets are calculated from the source ontology.

6 Discussion and Outlook

In this paper, we investigate a completeness guaranteed approximation, based on transformations of both the source ontology and input queries. Based on this approach and the results from [8], we have implemented both soundness preserving and completeness preserving approximations in our TrOWL ontology infrastructure. To the best of our knowledge, this is the first scalable OWL DL query engine supporting both soundness preserving and completeness preserving approximations.

From the literature, the closest approach is SCREECH-ALL [3, 11], which provides complete approximations for ABox reasoning (rather than conjunctive query answering) on $\mathcal{SHIQ}$ ontologies, based on the KAON2 reasoner. While SCREECH-ALL could produce many unsound answers for atomic and defined concepts, our approach is very precise on atomic concepts. In Fig. 1 the position of SCREECH-ALL is at point A or B or C, it depends on the ontology used. It should be noted that Fig. 1 illustrates an interesting advantage of our approach: when i is increasing our approach produces smaller completeness preserving answer sets; i.e., our anytime approach is monotonic. SCREECH-NONE [3, 11], on the other hand, simply removes all disjunctive rules and it guarantees sound by not complete reasoning. Unfortunately the result is that some instance of named concepts might be lost. The semantic approximation approach, however, is based on least upper bound approximation, i.e. the strongest weaker approximation w.r.t. DL-LiteR. Thus, it will never lose any instances of DL-LiteR basic concepts, let alone named concepts.

One immediate future work is to further test this approach on variant application ontologies. Study more optimization techniques in order to build more efficient systems will be one of the important future works.

References

1. B. Glimm, C. Lutz, I. Horrocks, and U. Sattler. Answering conjunctive queries in the $\mathcal{SHIQ}$ description logic. *Journal of Artificial Intelligence Research*, 31:150–197, 2008.
2. P. Groot, H. Stuckenschmidt, and H. Wache. Approximating Description Logic Classification for Semantic Web Reasoning. In *Proc. of ESWC2005*, 2005.
3. P. Hitzler and D. Vrandecic. Resolution-Based Approximate Reasoning for OWL DL. In *Proc. of the 4th International Semantic Web Conference (ISWC2005)*, 2005.
4. I. Horrocks and S. Tessaris. Querying the Semantic Web: a Formal Approach. In *Proc. of the 1st International Semantic Web Conference (ISWC 2002)*, pages 177–191, 2002.
5. C. Hurtado, A. Poulovassilis, and P. Wood. A Relaxed Approach to RDF Querying. In *Proc. of the 5th International Semantic Web Conference (ISWC-2006)*, 2006.
6. B. Motik and Ul. Sattler. A Comparison of Reasoning Techniques for Querying Large Description Logic ABoxes. In *Proc. of LPAR 2006*, pages 227–241, 2006.
7. M. Ortiz, D. Calvanese, and T. Eiter. Characterizing data complexity for conjunctive query answering in expressive description logics. In *In Proc. of AAAI 2006*, 2006.
8. J. Z. Pan and E. Thomas. Approximating OWL-DL Ontologies. In *the Proc. of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, pages 1434–1439, 2007.
9. Bart Selman and Henry Kautz. Knowledge Compilation and Theory Approximation. *Journal of the ACM (JACM)*, 43(2):193–224, 1996.
10. H. Stuckenschmidt and F. van Harmelen. Approximating Terminological Queries. In *Proc. of FQAS2002)*, pages 329–343, 2002.
11. Tuvshintur Tserendorj, Sebastian Rudolph, Markus Krötzsch, and Pascal Hitzler. Approximate owl-reasoning with screech. In Diego Calvanese and Georg Lausen, editors, *RR*, volume 5341 of *Lecture Notes in Computer Science*, pages 165–180. Springer, 2008.
12. H. Wache, P. Groot, and H. Stuckenschmidt. Scalable Instance Retrieval for the Semantic Web by Approximation. In *Proc. of WISE-2005 Workshop on Scalable Semantic Web Knowledge Base Systems*, 2005.