Closed World Reasoning for OWL2 with NBox*

REN Yuan, Jeff Z Pan**, ZHAO Yuting

Department of Computing Science, University of Aberdeen, Aberdeen, UK
{y.ren, jeff.z.pan, yuting.zhao}@abdn.ac.uk

Abstract. In this paper, the authors investigate the problem of doing Description Logic (DL) reasoning with partially closed world. They address this issue by extending the syntax of DL SROIQ with an NBox, which specifies the predicates to close, extending the semantics with the idea of Negation As Failure, reducing the closed world reasoning to incremental reasoning on classical DL ontologies and applying the syntactic approximation technology to improve the reasoning performance. Compared with the existing DBox approach, which corresponds to the relation database, the NBox approach supports deduction on closed concepts and roles. Also, the approximate reasoning can reduce reasoning complexity from N2EXPTIME-Complete to PTIME-Complete while preserve the correctness of reasoning for ontologies with certain properties.

Keywords: ontology; closed world; reasoning; approximation

1 Introduction

With the fast development of the Semantic Web, ontology has become one of the major knowledge formalisms in knowledge intensive systems. Today's de facto standard ontology language, the Web Ontology Language (OWL) and its various profiles [1] are based on a family of Description Logics (DLs) [2]. With its DL foundations, one of the major feature of ontology is that it imposes the Open World Assumption (OWA) in reasoning, which means a proposition is inferred to be true *iff* it is true in all possible models of the ontology.

In contrast, many other knowledge formalisms impose the Closed World Assumption (CWA), which means a proposition is inferred to be false *iff* it is not inferred to be true. In other words, unknown knowledge is regarded as false. This CWA is widely adopted by serious knowledge representations such as Logic Program and Relational Database. And it is also imposed by many real world applications in which people believe they have complete knowledge about the domain. For example, in Generation of Referring Expressions [3], if a certain relation is not entailed between two objects, then people believe these two objects have no such a relation at all. Also, CWA is important for the checking of integrity constraints [4].

* Received: 2010-09-21. Supported by the European Project Marrying Ontologies and Software Technologies (MOST, EU ICT 2008-216691).

** To whom correspondence should be addressed. Tel: +44(0)1224 274 449; E-mail: jeff.z.pan@abdn.ac.uk

In order to inter-operate knowledge bases and systems with both OWA and CWA, it is necessary to study doing ontology with the CWA. In [5] the notion of DBox is proposed. Briefly speaking, a DBox is a set of individual assertion axioms to which the interpretation of predicates appear in are restricted to. In other words, expect the explicitly asserted ones in the DBox, no further instances can be entailed for these predicates. This feature can be inconvenient or inefficient in practice because the ontology engineer has to explicitly assert all the possible instances for these predicates.

In this paper we overcome this difficulty, allowing inferences on closed predicates by extending the syntax of ontology with a Negation As Failure (NAF) Box (NBox for short). An NBox is a set of predicates that the engineer wants to close. The semantics of the NBox-closed ontology is also presented. Different from the "pre-closed" DBox approach, when doing reasoning with NBox, one should first retrieve the instances of the NBox predicates and then use these instances to close NBox predicates and proceed for further reasoning.

In order to improve the efficiency of this two-phase reasoning by reusing results of the first phase in the second phase, we apply the the syntactic approximation [6] to reduce the closed world reasoning in $\mathcal{SROIQ}$ to closed world reasoning in $\mathcal{EL}^{++}$, in which tractable incremental reasoning algorithm is available [7].

2 Syntactic Approximation of Description Logics

Approximations are techniques used to improve the performance of reasoning. They can be done in a semantic way (pre-computing the reasoning results off-line and retrieval them efficiently on-line) or in a syntactic way (reducing the ontologies from very expressive languages to less expressive languages). In this section we present the syntactic approximation. In later section, we will use it to support closed world reasoning.

Different OWL languages are underpinned by different DLs. For example, the underpinning of the OWL2-DL is DL $\mathcal{SROIQ}$. The underpinning of the OWL2-EL is DL $\mathcal{EL}^{++}$. For the sake of conciseness, we highlight the parts of syntax and semantics that are of interest in our paper. Complete specifications of these two languages can be found in [8] and [9, 10], respectively.

Let N_C, N_R and $N_I = \{a_1, a_2, \ldots, a_n\}$ be disjoint sets of concept, role and individual names, respectively. The set of $\mathcal{SROIQ}$ roles is N_R and, if R is a role, then R^- is also a role. The set of $\mathcal{SROIQ}$ concepts is the smallest set that includes the named concepts $N_C \cup \{\top\}$, and if C, D are concepts and R is a role, $a \in N_I$, and i is a non-negative integer, then $\neg C$, $C \sqcap D$, $\exists R.C$, $\{a\}$, $\geq iR.C$, $\exists R.Self$ are concepts. As usual, $\bot$, $C \cup D$, $\forall R.C$, $\{a_1, \ldots, a_m\}$ and $\leq iR.C$ are short for $\neg\top$, $\neg(\neg C \cup \neg D)$, $\neg\exists R.\neg C$, $\neg(\neg\{a_1\} \sqcap \ldots \sqcap \neg\{a_m\})$ and $\neg \leq i-1R.C$.

Similarly, the set of $\mathcal{EL}^{++}$ roles is N_R, while the $\mathcal{EL}^{++}$ concepts can only be $\top$, $\bot$, $A \in N_C$, $\{a\}$, $C \sqcap D$ or $\exists R.C$.

Without loss of generality, in what follows, we assume all the concepts in their unique *negated normal forms* (NNF. An $\mathcal{SROIQ}$ concept is in NNF *iff* negation is applied only to named concepts, nominals or Self-restriction. NNF of a given concept can be computed in linear time [11].) and use ~C to denote the NNF of $\neg C$. We assume all the roles in their unique *inverse normal form* (INF. A role is in INF *iff* inverse applies

only to role name.) by applying the transformation $(R^-)^- \rightarrow R$ and use $Inv(R)$ to denote the INF of R^-. We also call $\top, \bot, A, \{a\}$ *basic concepts* because they are not composed by other concepts or roles. Given a knowledge base Σ (an ontology or a TBox, or an ABox), we use CN_Σ (RN_Σ) to denote the set of basic concepts (named roles) in Σ. We use IN_Σ to denote the set of individuals in Σ.

For any DL, an ontology $\mathcal{O}$ is a pair $(\mathcal{T}, \mathcal{A})$. $\mathcal{T}$ is a terminology box (TBox), $\mathcal{A}$ is an assertion box (ABox). A TBox is a set of concept and role axioms. Both $\mathcal{SROIQ}$ and $\mathcal{EL}^{++}$ support concept inclusion axioms (CIs, e.g. $C \sqsubseteq D$) and role inclusion axioms (RIs, e.g. $r \sqsubseteq s$, $r_1 \circ \ldots \circ r_n \sqsubseteq s$). If $C \sqsubseteq D$ ($R \sqsubseteq S$) and $D \sqsubseteq C$ ($S \sqsubseteq R$), we write $C \equiv D$ ($R \equiv S$). If C is complex, $C \sqsubseteq D$ is a general concept inclusion axiom (GCI). An ABox is a set of assertion axioms. Both $\mathcal{SROIQ}$ and $\mathcal{EL}^{++}$ support the concept assertion axioms, e.g. $a : C$, role assertion axioms, e.g. $(a, b) : R$, individual equality, e.g. $a \doteq b$ and inequality, e.g. $a \neq b$. Furthermore, in $\mathcal{SROIQ}$, negative role assertion axioms, e.g. $(a, b) : \neg R$ is allowed.

An *interpretation* $\mathcal{I}$ is a pair $\langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ where $\Delta^\mathcal{I}$ is a non-empty set and $\cdot^\mathcal{I}$ is a function that maps named concept A to $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, named role r to $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$ and individual a to $a^I \in \Delta^\mathcal{I}$. The interpretation of complex concepts and some important forms of axioms are presented in Table 1.

Table 1. DL Semantics (# means the size of a set)

Syntax	Semantics
R^-	$\{(x, y) \mid (x^\mathcal{I}, y^\mathcal{I}) \in R^\mathcal{I}\}$
$\top$	Δ
$\bot$	$\emptyset$
$\neg C$	$\Delta \setminus C^\mathcal{I}$
$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
$C \sqcup D$	$C^\mathcal{I} \cup D^\mathcal{I}$
$\exists R.C$	$\{x \mid \exists y \in C^\mathcal{I}, (x^\mathcal{I}, y) \in R^I\}$
$\forall R.C$	$\{x \mid \forall y, (x^\mathcal{I}, y) \in R^I \rightarrow y \in C^\mathcal{I}\}$
$\{a_1, \ldots, a_m\}$	$\{a_1^I, \ldots, a_m^\mathcal{I}\}$
$\geq iR.C$	$\{x \mid \#\{y \mid y \in C^I, (x^I, y) \in R^\mathcal{I}\} \geq i\}$
$\leq iR.C$	$\{x \mid \#\{y \mid y \in C^I, (x^I, y) \in R^\mathcal{I}\} \leq i\}$
$\exists R.Self$	$\{x \mid (x^\mathcal{I}, x^\mathcal{I}) \in R^\mathcal{I}\}$
$C \sqsubseteq D$	$C^\mathcal{I} \subseteq D^\mathcal{I}$
$R \sqsubseteq S$	$R^\mathcal{I} \subseteq S^\mathcal{I}$
$a : C$	$a^\mathcal{I} \in C^\mathcal{I}$
$(a, b) : R$	$(a^\mathcal{I}, b^\mathcal{I}) \in R^\mathcal{I}$
$(a, b) : \neg R$	$(a^\mathcal{I}, b^\mathcal{I}) \notin R^\mathcal{I}$
$a \doteq b$	$a^\mathcal{I} = b^\mathcal{I}$
$a \neq b$	$a^\mathcal{I} \neq b^\mathcal{I}$

$\mathcal{I}$ is a *model* of an axiom α, denoted by $\mathcal{I} \models \alpha$ *iff* it satisfies the semantics of the axiom. Similarly, $\mathcal{I}$ is a model of $\mathcal{O}$, written $\mathcal{I} \models \mathcal{O}$, *iff* $\mathcal{I}$ is a model of all axioms in $\mathcal{O}$.

Some standard reasoning services include:

1. **Axiom Entailment Checking:** An axiom α is entailed by $\mathcal{O}$, denoted by $\mathcal{O} \models \alpha$ *iff* $\forall \mathcal{I}$, if $\mathcal{I} \models \mathcal{O}$ then $\mathcal{I} \models \alpha$. Otherwise, α is not entailed, written $\mathcal{O} \not\models \alpha$.
2. **Instance Retrieval:** An individual a is a instance of a concept C, *iff* $\mathcal{O} \models a : C$. Similarly, two individual a, b have an r relation, *iff* $\mathcal{O} \models (a, b) : r$.

Due to the different levels of expressive power, the complexity of the above reasoning services in $\mathcal{SROIQ}$ and $\mathcal{EL}^{++}$ are different. In $\mathcal{SROIQ}$, the complexity is N2EXPTIME-Complete. In $\mathcal{EL}^{++}$, the complexity is PTIME-Complete.

Because we are going to extend the syntax and semantics of ontology in later sections of this paper, we refer to the above definitions and notions as the *classical* cases. For example, classical ontology, classical semantics, etc.

Syntactic approximation [12, 6] is an approximate reasoning technique that reduces DL reasoning in $\mathcal{SROIQ}$ to $\mathcal{EL}^{++}$. The reasoning complexity is thus reduced from 2NEXPTIME-Complete to PTIME-Complete while the results are guaranteed correct. In this subsection, we briefly recall the syntactic approximation technique and its features. For more details about proofs, readers are referred to [12, 6]. In later sections, we will extend the current approach to support required closed world reasoning services.

The idea of syntactic approximation from $\mathcal{SROIQ}$ and $\mathcal{EL}^{++}$ is to first encode non-$\mathcal{EL}^{++}$ expressions with fresh names, then maintain their semantics with additional axioms and separate data structures. For example, complement relations between an named concept A and the new name, e.g. nA, assigned to its complement $\neg A$ are maintained in the *Complement Table* (CT). In reasoning phase, additional completion rules will be used to partially recover the semantics.

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 in any axiom, or (ii) a singleton of any individual, or (iii) the complement of a term, or (iv) the syntactic sub-expression of a term. In order to represent all these terms and role expressions that will be used in $\mathcal{EL}^{++}$ reasoning, we first assign names to them.

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

Now we can transform ontologies into $\mathcal{EL}^{++}$ with additional data structures by the following definition.

Definition 2. ***($\mathcal{EL}^{++}_{\mathcal{CQ}}$ Transformation)*** *Given an Ontology $\mathcal{O}$ and a name assignment fn, its $\mathcal{EL}^{++}_{\mathcal{CQ}}$* transformation
$A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(\mathcal{O})$ is a triple $(\mathcal{T}, CT, QT)$ constructed as follows:

1. *$\mathcal{T}, CT$ and QT are all initialized to $\emptyset$.*
2. *for each $C \sqsubseteq D$ ($C \equiv D$) in $\mathcal{O}$, $\mathcal{T} = \mathcal{T} \cup \{fn(C) \sqsubseteq fn(D)\}$ ($\mathcal{T} = \mathcal{T} \cup \{fn(C) \equiv fn(D)\}$).*
3. *for each $\mathcal{EL}^{++}$ role axiom $\beta \in \mathcal{O}$, add $\beta_{[R/fn(R)]}$ into $\mathcal{T}$.*
4. *for each $a : C \in \mathcal{O}$, $\mathcal{T} = \mathcal{T} \cup \{\{a\} \sqsubseteq fn(C)\}$.*

5. *for each* $(a, b) : R \in \mathcal{O}$, $\mathcal{T} = \mathcal{T} \cup \{\{a\} \sqsubseteq \exists fn(R).\{b\}, \{b\} \sqsubseteq \exists fn(Inv(R)).\{a\}\}$.
6. *for each* $a \doteq b \in \mathcal{O}$, $\mathcal{T} = \mathcal{T} \cup \{\{a\} \equiv \{b\}\}$.
7. *for each* $a \neq b \in \mathcal{O}$, $\mathcal{T} = \mathcal{T} \cup \{\{a\} \sqsubseteq \neg\{b\}\}$
8. *for each term* C *in* $\mathcal{T}_{\mathcal{O}}$, $CT = CT \cup \{(fn(C), fn(\sim C))\}$, *and*
   (a) *if* C *is the form* $C_1 \sqcap \ldots \sqcap C_n$, *then* $\mathcal{T} = \mathcal{T} \cup \{fn(C) \equiv fn(C_1) \sqcap \ldots \sqcap fn(C_n)\}$,
   (b) *if* C *is the form* $\exists r.D$, *then* $\mathcal{T} = \mathcal{T} \cup \{fn(C) \equiv \exists r.fn(D)\}$,
   (c) *if* C *is the form* $\exists R.Self$, *then* $T = T \cup \{fn(C) \equiv \exists fn(R).Self\}$
   (d) *if* C *is the form* $\geq nR.D$, *then*
      i. *if* $n = 0$, $\mathcal{T} = \mathcal{T} \cup \{\top \sqsubseteq fn(C)\}$
      ii. *if* $n = 1$, $\mathcal{T} = \mathcal{T} \cup \{fn(C) \equiv \exists fn(R).fn(D)\}$
      iii. *otherwise,* $\mathcal{T} = \mathcal{T} \cup \{fn(C) \equiv fn(D)^{fn(R),n}\}$, *and* $QT = QT \cup \{(fn(C), fn(R), n)\}$.
   (e) *otherwise* $\mathcal{T} = \mathcal{T} \cup \{fn(C) \sqsubseteq \top\}$.
9. *for each pair of names* A *and* r, *if there exist* (A, r, i_1), (A, r, i_2), ..., $(A, r, i_n) \in QT$ *with* $i_1 < i_2 < \ldots < i_n$, $\mathcal{T} = \mathcal{T} \cup \{A^{r,i_n} \sqsubseteq A^{r,i_{n-1}}, \ldots, A^{r,i_2} \sqsubseteq A^{r,i_1}, A^{r,i_1} \sqsubseteq \exists r.A\}$

Step 2 rewrites all the concept axioms; Step 3 preserves all the $\mathcal{EL}^{++}$ role axioms; Step 4 to 7 rewrite all the ABox axioms and internalize them into the approximated TBox; Step 8 defines terms and constructs the complement table CT and cardinality table QT; Particularly, in step 8.(d), $fn(D)^{fn(R),n}$ is a fresh name. For example, $A^{r,3}$ for $\geq 3r.A$. Obviously, this is unique for a given tuple of D, R and n. We call them *cardinality names*. Similarly, $\leq nR.D$ will be approximated via the approximation of its complement $\geq (n+1)R.D$. In step 9, 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*.The $\mathcal{EL}^{++}_{\mathcal{CQ}}$ approximation approximates a TBox into an $\mathcal{EL}^{++}$ ontology with a table maintaining the complementary relations and another table maintaining the cardinality relations. This approximation can be computed in linear time:

Proposition 1. *($\mathcal{EL}^{++}_{\mathcal{CQ}}$* ***Approximation)*** *For an ontology* $\mathcal{O}$, *let* $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(\mathcal{O}) = (\mathcal{T}, CT, QT)$, *then* $\mathcal{T}$ *is an* $\mathcal{EL}^{++}$ *TBox.*

Proposition 2. *For any ontology* $\mathcal{O}$ *and* $(\mathcal{T}, CT, QT)$ *its* $\mathcal{EL}^{++}_{\mathcal{CQ}}$ *transformation, if* $\mathcal{O}$ *contains* $n_{\mathcal{O}}$ *terms, then* $CN_{\mathcal{T}}, \mathcal{T}, CT, QT$ *are all of linear size of* $|\mathcal{O}| + n_{\mathcal{O}}$.

Given an $\mathcal{EL}^{++}_{\mathcal{CQ}}$ transformation $(\mathcal{T}, CT, QT)$, we normalize axioms of form $C \sqsubseteq D_1 \sqcap \ldots \sqcap D_n$ into $C \sqsubseteq D_1, \ldots, C \sqsubseteq D_n$, and recursively normalize 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 $\mathcal{T}$ to be always normalized. For convenience, we use a *complement function* $fc : CN_{\mathcal{T}} \mapsto CN_{\mathcal{T}}$ as: for each $A \in CN_{\mathcal{T}}$, $fc(A) = B$ if $(A, B) \in CT$. Note that if A is a cardinality name, then it does not have a complement. In what follows, when applying $fc(A)$ we always assume that A is not a cardinality name but a assigned name.

With the normalized approximation, the reasoning can be realized by extending $\mathcal{EL}^{++}$ completion rules with support for the CT and QT. Given an ontology $\mathcal{EL}^{++}_{\mathcal{CQ}}$

transformation $(\mathcal{T}, CT, QT)$, the completion rules will compute, for each basic concept A, a subsumer set $S(A) \subseteq CN_{\mathcal{T}}$ such that if $B \in S(A)$ then $A \sqsubseteq B$, and for each named role r, a relation set $R(r) \subseteq CN_{\mathcal{T}} \times CN_{\mathcal{T}}$. For each basic concept A, $S(A)$ is initialized to $\{A, \top\}$ and for each named role r, $R(r)$ is initialized to $\emptyset$. Then, the rules in Table 2 are repeatedly applied until no new results can be derived.

Table 2. $\mathcal{EL}^{++}_{\mathcal{CQ}}$ completion rules

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 $A_1, \ldots, A_n \in S(X)$, $A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B \in \mathcal{T}$ 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 $R(r_3) := R(r_3) \cup \{(X, B)\}$
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 $A_1 \sqcap \ldots \sqcap A_n \sqsubseteq \bot$, $A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_n \in S(X)$ and $fc(A_i) \notin S(X)$ then $S(X) := S(X) \cup \{fc(A_i)\}$
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 $S(A^{r,i}) := S(A^{r,i}) \cup \{B^{s,j}\}$
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 $R(r_3) := R(r_3) \cup \{(X, B)\}$

In Table 2, the rules **R1-8** are the original $\mathcal{EL}^{++}$ completion rules [9]. Especially, in **R6** $X \rightsquigarrow_R A$ *iff* there exists $C_1, \ldots, C_k \in CN_T$ s.t. $C_1 = X$ or $C_1 = \{b\}$, $(C_j, C_{j+1}) \in R(r_j)$ for some $r_j \in RN_T (1 \leq j \leq k)$ and $C_k = A$. This *reachability* can be computed in polynomial time. Rules **R9-R16** are devised to make use of information maintained in CT and QT. For example, **R9** realizes axiom $A \sqcap \sim A \sqsubseteq \bot$.

R10 asserts the reverse subsumption between concepts to supplement the absence of negation, i.e. $A \sqsubseteq B \rightarrow \sim A \sqsubseteq \sim B$. **R11** builds up the relations between conjuncts of a conjunction, *e.g.* $A \sqcap B \sqsubseteq \bot$ implies $A \sqsubseteq \sim B$. **R12**, in which $r \sqsubseteq_* s$ if $r = s$ or $r \sqsubseteq s \in T$, realizes 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**. Reasoning with rules **R1-16** is tractable and sound:

Theorem 1. ***(Complexity)*** *For any $\mathcal{EL}^{++}_{\mathcal{CQ}}$ transformation $(\mathcal{T}, CT, QT)$ ($\mathcal{T}$ normalized), TBox reasoning by completion rules **R1-R16** will terminate in polynomial time w.r.t. $|CN_{\mathcal{T}}|+|RN_{\mathcal{T}}|$.*

Theorem 2. ***(Concept Subsumption Checking)*** *Given an ontology $\mathcal{O}$, its vocabulary $V_{\mathcal{O}}$ and $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(\mathcal{O}) = (\mathcal{T}, CT, QT)$, for any two concepts C and D constructed from $V_{\mathcal{O}}$, if $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(\{C \sqsubseteq \top, D \sqsubseteq \top\}) = (\mathcal{T}', CT', QT')$, then $\mathcal{O} \models C \sqsubseteq D$ if $fn(D) \in S(fn(C))$ can be computed by rules **R1-R16** on $(\mathcal{T} \cup \mathcal{T}', CT \cup CT', QT \cup QT')$.*

As in classical reasoning, 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 $\{a\} \sqsubseteq \bot$.

3 Technical Motivation

In this section, we introduce the notion of Open World Assumption and Closed World Assumption and motivate our work by discussion of existing works.

3.1 Open World Assumption & Closed World Assumption

Classical DL reasoning imposes the Open World Assumption (OWA), which means the truth-value of a statement is unknown if its value varies in different interpretations of the ontology. According to the definition of entailment in DL, the truth-value of an entailment is true *iff* the entailment holds in all interpretations of the ontology. It is false *iff* the entailment does not hold in any interpretation. Otherwise, it is unknown. For example, in ontology $\{a : A, b : B\}$ whether $a : B$ is unknown, so is $b : A$. DL reasoning can be done with OWA. For example, in the classic example $\{Oedipus : Patricide, Thersandros : \neg Patricide, (Iokaste, Oedipus) : hasChild, (Iokaste, Polyneikes) : hasChild, (Oedipus, Polyneikes) : hasChild, (Polyneikes, Thersandros) : hasChild\}$, we can inferred that $Iokaste : \exists hasChild.(Patricide \sqcap \exists hasChild.\neg Patricide)$ is true in all interpretations, no matter $Polyneikes : Patricide$ or not.

However, in certain applications, as we mentioned in the Sec.1, it is required to partially or completely close the world or domain. In this case, the truth-value of a statement that is not known to be true, is false. Thus in the above example $\{a : A, b : B\}$, if A, B are closed, then $a : \neg B$ and $b : \neg A$ can be entailed. The CWA has been widely applied in relational database and logic programming.

3.2 Closing the Domain with DBox Approach

CWA can be realized in different ways. [5] presented the DBox approach. In addition to the TBox $\mathcal{T}$ and ABox $\mathcal{A}$, a DBox $\mathcal{D}$ is specified to close the domain. $\mathcal{D}$ is syntactically similar to $\mathcal{A}$, except that in $\mathcal{D}$ only named concepts and named roles are allowed. Furthermore, the interpretations of predicates appearing in $\mathcal{D}$ is fixed by $\mathcal{D}$. That is to say, in any model of the ontology $\mathcal{I}$, if $A \in CN_D$, then $A^{\mathcal{I}} = \{a^{\mathcal{I}} | a : A \in \mathcal{D}\}$. If $r \in RN_D$, then $r^{\mathcal{I}} = \{(a^{\mathcal{I}}, b^{\mathcal{I}}) | (a, b) : r \in \mathcal{D}\}$. For predicates not appearing in $\mathcal{D}$, their interpretations are the same as we introduced in Sec.2, especially in Table 1.

The DBox approach strongly corresponds to relational database. Actually, the data tables in a relation database can be regarded as a DBox. This resemblance makes it easy to reduce query answering with ontologies over DBox to relational database query answering [5].

The use of DBox is based on the assumption that a user has complete knowledge about the predicates he or she wants to close. In this case, a user can confidently encode such knowledge with the DBox. However, the DBox actually prohibits inferences on predicates appearing in it. In certain cases this can be inconvenient and unscalable. For example, Given an ontology $\mathcal{T} = \{PhDStudent \sqsubseteq Student\}, \mathcal{A} = \{Emily : Student, David : PhDStudent, Emily \neq David\}$ one should infer $David : Student$. This kind of inferences are also witnessed by deductive database and DataLog. However, if we close $Student$ by putting only $Emily : Student$ into $\mathcal{D}$, this ontology becomes inconsistent, because $David$, which is inferred to be a $Student$, is different from the only instance of $Student$.

From this example, we can see that the DBox approach does not only require a user to have complete knowledge about certain domain, but also requires him or her to explicitly assert this knowledge. This will introduce a lot of redundancies and undermines DL's advantages of reasoning.

It is nature to ask, is that possible to close certain predicates but still allow inferences of implicit knowledge about them? To answer this question, we present a realization of CWA with Negation As Failure.

4 Introducing Negation As Failure Box (NBox)

In this section, we present our realization of CWA. We start from the notion of Negation As Failure (NAF). Then we present the syntax and semantics of an ontology with a Negation As Failure Box (NBox).

4.1 Negation As Failure

As we mentioned earlier, the idea of CWA is that the truth-value of a statement that is not known to be true, is false. However, the meaning of "known" here is unclear. In knowledge representation, people tend to distinguish the **Explicit Knowledge** and **Implicit Knowledge**. Formally speaking, given a knowledge base Σ, a proposition P is explicit knowledge *iff* $P \in \Sigma$. A proposition P is implicit knowledge *iff* $P \notin \Sigma$ and $\Sigma \models P$.

Obviously, the DBox approach closes the domain w.r.t. the explicit knowledge encoded in $\mathcal{D}$, leaving no space for implicit knowledge of predicates in the DBox. If we want to close the domain w.r.t. both the explicit and implicit knowledge, it should be done by the following non-monotonic inference rule:

Definition 3. ***(Negation As Failure (NAF))*** *For a knowledge base Σ and a proposition P, $\Sigma \models \bar{P}$* iff *$\Sigma \not\models P$, where $\Sigma \models \bar{P}$ means Σ entails that P is not true.*

Particularly, when $P = a : C$, $\Sigma \models \bar{P}$ *iff* $\Sigma \models a : \neg C$. When $P = (a, b) : r$, $\Sigma \models \bar{P}$ *iff* $\Sigma \models (a, b) : \neg r$.

In ontology reasoning, not all the predicates need to be closed. Therefore, NAF should be applied on a pre-specified set of predicates. We call this set of predicates the *Negation As Failure Box* (NBox).

4.2 NBox-closed Ontology

The syntax of NBox and NBox-closed Ontology is defined as follows:

Definition 4. ***(NBox-closed Ontology)*** *An NBox-closed Ontology (N-Ontology for short) $\mathcal{O}$ is a triple $(\mathcal{T}, \mathcal{A}, \mathcal{N})$, in which $\mathcal{T}$ is a TBox, $\mathcal{A}$ is an ABox and $\mathcal{N}$ is a subset of the set containing $\mathcal{T}$, all the named concepts and named roles in $\mathcal{T}$ and $\mathcal{A}$.*

In order to distinguish the classical ontology and N-Ontology, in what follows, we use $\mathcal{O}$ to denote a classical ontology and $\mathcal{O}^{\mathcal{N}}$ to denote an N-Ontology.

The semantics of an N-Ontology can be extended by the NBox as follows:

Definition 5. ***(N-Ontology Semantics)*** *Given an N-Ontology $\mathcal{O}^{\mathcal{N}} = (\mathcal{T}, \mathcal{A}, \mathcal{N})$, let $\mathcal{O} = (\mathcal{T}, \mathcal{A})$, an* interpretation *$\mathcal{I}$ of $\mathcal{O}^{\mathcal{N}}$ is a pair $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ following the classical notion of interpretation except that:*

1. *If a concept $A \in \mathcal{N}$, then $A^{\mathcal{I}} = \{a^{\mathcal{I}} | \mathcal{O} \models a : A\}$.*
2. *If a role $r \in \mathcal{N}$, then $r^{\mathcal{I}} = \{(a^{\mathcal{I}}, b^{\mathcal{I}}) | \mathcal{O} \models (a, b) : A\}$*

In other words, the interpretation of a predicate in $\mathcal{N}$ is restricted to the maximal common subset of its interpretations over $\mathcal{O}$.

With the above semantics, we have the similar definition for entailment checking as in classical ontology: an axiom α is entailed by an N-Ontology $\mathcal{O}^{\mathcal{N}}$ *iff* it is entailed by all interpretations of $\mathcal{O}^{\mathcal{N}}$, denoted by $\mathcal{O}^{\mathcal{N}} \models \alpha$.

We show that the NBox predicates satisfy the NAF inference in Def.3.

Theorem 3. ***(NBox for NAF)*** *For any N-Ontology $\mathcal{O}^{\mathcal{N}} = (\mathcal{T}, \mathcal{A}, \mathcal{N})$, the following holds:*

1. *If concept $A \in \mathcal{N}$, then $\mathcal{O}^{\mathcal{N}} \models a : \neg A$* iff *$(\mathcal{T}, \mathcal{A}) \not\models a : A$.*
2. *If role $r \in \mathcal{N}$, then $\mathcal{O}^{\mathcal{N}} \models (a, b) : \neg r$* iff *$(\mathcal{T}, \mathcal{A}) \not\models (a, b) : r$.*

Proof. 1. By definition of N-Ontology entailment, $\mathcal{O}^{\mathcal{N}} \models a : \neg A$ *iff* $\forall \mathcal{I}$ of $\mathcal{O}^{\mathcal{N}}$, $\mathcal{I} \models a : \neg A$, i.e. $a^{\mathcal{I}} \notin A^{\mathcal{I}}$.
According to Def.5, because $A \in \mathcal{N}$, $A^{\mathcal{I}} = \{a^{\mathcal{I}} | (\mathcal{T}, \mathcal{A}) \models a : A\}$. Therefore, due to the definition of classical entailment checking, $a^{\mathcal{I}} \notin A^{\mathcal{I}}$ *iff* there exists some interpretation of $(\mathcal{T}, \mathcal{A})$ that does not satisfies $a : A$, which means $(\mathcal{T}, \mathcal{A}) \not\models a : A$. Together we have $\mathcal{O}^{\mathcal{N}} \models a : \neg A$ *iff* $(\mathcal{T}, \mathcal{A}) \not\models a : A$.

2. The proof of role entailment is similar.
   By definition of N-Ontology entailment, $\mathcal{O}^{\mathcal{N}} \models (a,b) : \neg r$ *iff* $\forall \mathcal{I}$ of $\mathcal{O}^{\mathcal{N}}$, $\mathcal{I} \models (a,b) : \neg r$, i.e. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \notin r^{\mathcal{I}}$.
   According to Def.5, because $r \in \mathcal{N}$, $r^{\mathcal{I}} = \{(a^{\mathcal{I}}, b^{\mathcal{I}}) | (\mathcal{T}, \mathcal{A}) \models (a,b) : r\}$. Therefore, due to the definition of classical entailment checking, $(a^{\mathcal{I}}, b^{\mathcal{I}}) \notin r^{\mathcal{I}}$ *iff* there exists some interpretation of $(\mathcal{T}, \mathcal{A})$ that does not satisfies $(a,b) : r$, which means $(\mathcal{T}, \mathcal{A}) \not\models (a,b) : r$.
   Together we have $\mathcal{O}^{\mathcal{N}} \models (a,b) : \neg r$ *iff* $(\mathcal{T}, \mathcal{A}) \not\models (a,b) : r$. □

5 Closed World Reasoning with N-Ontology

In this section, we discuss how to do closed world reasoning with NBox.

5.1 Reasoning Reduction

We first show that N-Ontology reasoning can be reduced to classical ontology reasoning.

Definition 6. ***(NBox Internalization)*** *Given an N-Ontology $\mathcal{O}^{\mathcal{N}} = (\mathcal{T}, \mathcal{A}, \mathcal{N})$, its* NBox Internalization *$NI(\mathcal{O}^{\mathcal{N}})$ is an ontology $\mathcal{O} = (\mathcal{T}', \mathcal{A}')$ constructed as follows:*

1. $\mathcal{A}' = \mathcal{A}$, $\mathcal{T}' = \mathcal{T}$.
2. $\mathcal{T}' = \mathcal{T}' \cup \{A \equiv \{a | (\mathcal{T}, \mathcal{A}) \models a : A\} | A \in \mathcal{N}\}$.
3. $\mathcal{T}' = \mathcal{T}' \cup \{\exists r.\top \sqsubseteq \{a | \exists b, (\mathcal{T}, \mathcal{A}) \models (a,b) : r\} | r \in \mathcal{N}\}$.
4. $\mathcal{T}' = \mathcal{T}' \cup \{\exists r^{-}.\top \sqsubseteq \{b | \exists a, (\mathcal{T}, \mathcal{A}) \models (a,b) : r\} | r \in \mathcal{N}\}$.
5. $\mathcal{T}' = \mathcal{T}' \cup \{\{a\} \sqsubseteq \forall r.\{b | (\mathcal{T}, \mathcal{A}) \models (a,b) : r\} | r \in \mathcal{N}\}$.
6. $\mathcal{T}' = \mathcal{T}' \cup \{\{b\} \sqsubseteq \forall r^{-}.\{a | (\mathcal{T}, \mathcal{A}) \models (a,b) : r\} | r \in \mathcal{N}\}$.

The above Step-2 closes the concepts in $\mathcal{N}$. Step-3 and 4 close the global domains and ranges of roles in $\mathcal{N}$. Step-5 and 6 close the local domains and ranges of roles in $\mathcal{N}$.

The following theorem shows that N-Ontology reasoning can be reduced to its NBox internalization reasoning:

Theorem 4. ***(Reasoning Reduction)*** *For any N-Ontology $\mathcal{O}^{\mathcal{N}} = (\mathcal{T}, \mathcal{A}, \mathcal{N})$ and its NBox internalization $NI(\mathcal{O}^{\mathcal{N}}) = \mathcal{O} = (\mathcal{T}', \mathcal{A}')$, let α an axiom, then we have $\mathcal{O}^{\mathcal{N}} \models \alpha$* iff *$\mathcal{O} \models \alpha$.*

Proof. According to the definition of N-Ontology entailment and Def.5, it is sufficient to prove that for any interpretation $\mathcal{I}$ of $\mathcal{O}'$, the following holds

1. If concept $A \in \mathcal{N}$, $A^{\mathcal{I}} = \{a^{\mathcal{I}} | (\mathcal{T}, \mathcal{A}) \models a : A\}$
2. If role $r \in \mathcal{N}$, $r^{\mathcal{I}} = \{(a^{\mathcal{I}}, b^{\mathcal{I}}) | (\mathcal{T}, \mathcal{A}) \models (a,b) : r\}$

We show that these two propositions are both true:

1. If concept $A \in \mathcal{N}$, according to Step-2 of Def.6, $\mathcal{O}' \models A \equiv \{a | (\mathcal{T}, \mathcal{A}) \models a : A\}$. Due to the definition of entailment we have $A^{\mathcal{I}} = \{a^{\mathcal{I}} | (\mathcal{T}, \mathcal{A}) \models a : A\}$.

2. If role $r \in \mathcal{N}$, according to Step-3 and 4 of Def.6, we have $r^{\mathcal{I}} \subseteq \{a^{\mathcal{I}} | \exists b, (\mathcal{T}, \mathcal{A}) \models (a, b) : r\} \times \{b^{\mathcal{I}} | \exists a, (\mathcal{T}, \mathcal{A}) \models (a, b) : r\}$.
   Together with Step-5 and 6 of Def.6, we have $r^{\mathcal{I}} = \{(a^{\mathcal{I}}, b^{\mathcal{I}}) | (\mathcal{T}, \mathcal{A}) \models (a, b) : r\}$.

All together, the theorem can be proved. □

According to Def.6, the NBox reasoning by NBox internalization requires the following steps:

1. Retrieving the instances of the NBox predicates.
2. Internalizing the NBox with the materialization from Step-1.
3. Reasoning over the NBox internalization from Step-2.

This procedure actually forms an incremental reasoning over the TBox and ABox in the original N-Ontology with two phases. This is usually difficult in expressive DLs such as $\mathcal{SROIQ}$ because the results in the first phase, i.e. the above Step-1 can not be reused in the second phase, i.e. the above Step-3. Alternatively, we can approximate this to $\mathcal{EL}^{++}$, which has tractable incremental reasoning service [7].

5.2 Closed World Reasoning by Approximation

The first step of closed world reasoning with N-Ontologies is instance retrieval of the NBox predicates. As we mentioned earlier, the $\mathcal{EL}^{++}_{\mathcal{CQ}}$ approximation completion rules **R1-16**, extended from the $\mathcal{EL}^{++}$ completion rules, are devised to compute the subsumption relations between concepts. Therefore, the instance retrieval of named concepts A can be reduced to computing all the singleton subconcepts of A. This can be done simultaneously in one go for all the named concepts.

The instance retrieval of named roles r is trickier. It requires, for each pair of individuals a, b, checking whether $\{a\} \sqsubseteq \exists r.\{b\}$ can be entailed. This can not be trivially done by checking whether $(\{a\}, \{b\}) \in R(r)$ because the original $\mathcal{EL}^{++}$ completion rules omitted many such relations. For example, from $\{A \sqsubseteq \exists r.B, B \sqsubseteq C\}$ the system can infer $(A, B) \in R(r)$ but not $(A, C) \in R(r)$. Therefore it is necessary to extend the $\mathcal{EL}^{++}_{\mathcal{CQ}}$ completion rules to efficiently compute all these entailments instead of checking entailements one by one. This can be realized by the following additional completion rule (Table 3).

Table 3. $\mathcal{EL}^{++}_{\mathcal{CQ}}$ additional role completion rule

R17	If $(\{a\}, A) \in R(r)$, $\{b\} \in S(A)$ and $(\{a\}, \{b\}) \notin R(r)$ then $R(r) := R(r) \cup \{(\{a\}, \{b\})\}$

Obviously, this additional rule has no impact on the computation of S sets:

Proposition 3. *For any $\mathcal{EL}^{++}_{\mathcal{CQ}}$ transformation $(\mathcal{T}, CT, QT)$, $A \in S(X)$ can be entailed by rules **R1-16** iff $A \in S(X)$ can be entailed by rules **R1-17**.*

Proof. This proposition follows from the facts that: (i) The additional **R17** only extends the results of R sets. (ii) Only **R4** and **R13** use results of R sets to derive results of S sets. (iii) No matter with **R17** or not, the application results of **R4** is the same. So does **R13**. □

Which means this rule will not affect the concept instance retrieval. We show that this rule will "complete" the role instance retrieval:

Theorem 5. *For any $\mathcal{EL}^{++}_{\mathcal{CQ}}$ transformation $(\mathcal{T}, CT, QT)$, $\{a\} \sqsubseteq \exists r.\{b\}$ can be entailed by rules **R1-16** as in Theorem 2* iff *$(\{a\}, \{b\}) \in R(r)$ can be entailed by rules **R1-17**.*

Proof. According to Theorem 2, to check $\{a\} \sqsubseteq r.\{b\}$ with rules **R1-16**, we first approximate $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(\{\{a\} \sqsubseteq \top, \exists r.\{b\} \sqsubseteq \top\}) = \{\mathcal{T}', CT', QT'\}$. Then we do reasoning with **R1-R16** on $(\mathcal{T} \cup \mathcal{T}', CT \cup CT', QT \cup QT')$. Suppose $fn(\exists r.\{b\}) = X$, obviously we have $\mathcal{T}' = \{\{a\} \sqsubseteq \top, X \sqsubseteq \top, X \sqsubseteq \exists r.\{b\}, \exists r.\{b\} \sqsubseteq X\}$. And $\{a\} \sqsubseteq r.\{b\}$ *iff* $X \in S(\{a\})$ is derived.

When doing reasoning with **R1-16**, rules **R1,2,4,6,10,13** can be applied to derive $X \in S(\{a\})$. We show that, no matter which rule is used to derived $X \in S(\{a\})$ in $\mathcal{T} \cup \mathcal{T}'$, **R1-17** can be used to derive $(\{a\}, \{b\}) \in R(r)$ in $\mathcal{T}$. For convenience, we use $\in_{\mathcal{T}\cup\mathcal{T}'}$ and $\in_{\mathcal{T}}$ to distinguish the results derived by **R1-16** on $\mathcal{T} \cup \mathcal{T}'$ and those derived by **R1-17** on $\mathcal{T}$:

1. **R1:** This rule requires some $Y \in_{\mathcal{T}\cup\mathcal{T}'} S(\{a\})$ s.t. $Y \sqsubseteq X \in \mathcal{T} \cup \mathcal{T}'$. Obviously, there's no such axiom in $\mathcal{T}'$, thus there exists $Y \in S_{\mathcal{T}\cup\mathcal{T}'}(\{a\})$ s.t. $Y \sqsubseteq X \in \mathcal{T}$. According to Proposition 3 we can see that $Y \in S_{\mathcal{T}\cup\mathcal{T}'}(\{a\})$ *iff* $Y \in_{\mathcal{T}} S(\{a\})$. Thus with **R1** we can derive $X \in_{\mathcal{T}} S(\{a\})$. According to Def. 2 there should also exists $X \sqsubseteq \exists r.\{b\} \in \mathcal{T}$. With **R3** we can derive $(\{a\}, \{b\}) \in_{\mathcal{T}} R(r)$.
2. **R2:** Similar as **R1:**, if **R2** is applied to derive $X \in_{\mathcal{T}\cup\mathcal{T}'} (\{a\})$, we will have $X \in_{\mathcal{T}} S(\{a\})$. And further with **R3** we have $(\{a\}, \{b\}) \in_{\mathcal{T}} R(r)$.
3. **R10:** Also similar as **R1:**, $fc(\{a\}) \in_{\mathcal{T}\cup\mathcal{T}'} S(fc(X))$ *iff* $fc(\{a\}) \in_{\mathcal{T}} S(fc(X))$. So does $X \in S(\{a\})$. And then we have $(\{a\}, \{b\}) \in_{\mathcal{T}} R(r)$.
4. **R6:** This rule requires some $\{b\} \in_{\mathcal{T}\cup\mathcal{T}'} S(\{a\}) \cap S(X)$ and $\{a\} \rightsquigarrow_{R,\mathcal{T}\cup\mathcal{T}'} X$. Obviously $\{b\} \in_{\mathcal{T}\cup\mathcal{T}'} S(\{a\}) \cap S(X)$ *iff* $\{b\} \in_{\mathcal{T}} S(\{a\}) \cap S(X)$. It is sufficient to prove $\{a\} \rightsquigarrow_{R,\mathcal{T}\cup\mathcal{T}'} X$ *iff* $\{a\} \rightsquigarrow_{R,\mathcal{T}} X$:
   - The $\rightarrow$ direction is obvious.
   - Regarding the $\leftarrow$ direction, according to the meaning of $\rightsquigarrow_R$, $\{a\} \rightsquigarrow_{R,\mathcal{T}} X$ *iff* there exist $C_1, \ldots, C_k \in CN_{\mathcal{T}}$ s.t. $C_1 = \{a\}$ or $C_1 = \{y\}$, $(C_j, C_{j+1}) \in_{\mathcal{T}} R(r_j)$ for some $r_j \in RN_{\mathcal{T}} (1 \leq j \leq k)$ and $C_k = X$. Comparing **R1-16** and **R1-17** we know that for each two C_j, C_{j+1}, either $(C_j, C_{j+1}) \in_{\mathcal{T}\cup\mathcal{T}'} R(r_j)$ or $(C_j, Y) \in_{\mathcal{T}\cup\mathcal{T}'} R(r_j), C_{j+1} = \{y\} \in_{\mathcal{T}\cup\mathcal{T}'} S(Y)$. In the former case, we have $\{a\} \rightsquigarrow_{R,\mathcal{T}\cup\mathcal{T}'} X$. In the later case, there exist some $\{y\}, C_{j+2}, \ldots, C_{k-1}, X$, which also makes $\{a\} \rightsquigarrow_{R,\mathcal{T}\cup\mathcal{T}'} X$.

   Thus, with **R6**, we also have $X \in_{\mathcal{T}\cup\mathcal{T}'} S(\{a\})$ *iff* $X \in_{\mathcal{T}} S(\{a\})$. Similarly, we can derive $(\{a\}, \{b\}) \in_{\mathcal{T}} R(r)$.

5. **R4:** This rule requires $(\{a\}, A) \in_{\mathcal{T}\cup\mathcal{T}'} R(r), B \in_{\mathcal{T}\cup\mathcal{T}'} S(A), \exists r.B \sqsubseteq X \in \mathcal{T} \cup \mathcal{T}'$. Obviously, $(\{a\}, A) \in_{\mathcal{T}\cup\mathcal{T}'} R(r)$ *iff* $(\{a\}, A) \in_{\mathcal{T}} R(r)$ because $\mathcal{T}'$ does not introduce any new relation regarding $\{a\}$; $B \in_{\mathcal{T}\cup\mathcal{T}'} S(A)$ *iff* $B \in_{\mathcal{T}} S(A)$ because $\mathcal{T}'$ does not introduce any new concept subsumption:
   - If $\exists r.B \sqsubseteq X \in \mathcal{T}$, then applying **R4** on $\mathcal{T}$ we will also get $X \in_{\mathcal{T}} (\{a\})$. Similarly applying **R3** yields $(\{a\}, \{b\}) \in_{\mathcal{T}} R(r)$.
   - If $\exists r.B \sqsubseteq X \in \mathcal{T}'$, then obviously $B = \{b\}$. Now we have $(\{a\}, A) \in_{\mathcal{T}} (R, r), \{b\} \in_{\mathcal{T}} S(A)$. Applying **R17** yields $(\{a\}, \{b\}) \in_{\mathcal{T}} R(r)$.
6. **R13:** This rule can be proved similar as **R4**.

Altogether, this theorem can be proved. □

It's worth mentioning that this theorem is not claiming the reasoning to be complete. It shows that, instead of checking relations between one by one pairs of individuals, the relations between individuals can be computed simultaneously, just like the concept instances.

With the extended rule sets, we can compute the instances for all the named concepts and named roles in one go of the reasoning. Given an $\mathcal{EL}^{++}_{\mathcal{CQ}}$ transformation $(\mathcal{T}, CT, QT)$, if $A \in S(\{a\})$ can be derived by **R1-17**, we say that $a : A$ can be entailed, denoted by $(\mathcal{T}, CT, QT) \models a : A$. If $(\{a\}, \{b\}) \in R(r)$ can be derived by **R1-17**, we say that $(a, b) : r$ can be entailed, denoted by $(\mathcal{T}, CT, QT) \models (a, b) : r$.

We approximate the N-Ontology reasoning with the approximation of the NBox Internalization.

Definition 7. ***(Approximate NBox Internalization)*** *Given an N-Ontology $\mathcal{O}^{\mathcal{N}} = (\mathcal{T}^{\mathcal{N}}, \mathcal{A}^{\mathcal{N}}, \mathcal{N})$, let $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}((\mathcal{T}^{\mathcal{N}}, \mathcal{A}^{\mathcal{N}})) = (\mathcal{T}, CT, QT)$, its* Approximate NBox Internalization $ANI(\mathcal{O}^{\mathcal{N}})$ *is an $\mathcal{EL}^{++}_{\mathcal{CQ}}$ transformation $(\mathcal{T}', CT', QT')$ constructed as follows:*

1. *Let $\mathcal{T}^T = \emptyset$.*
2. $\mathcal{T}^T = \mathcal{T}^T \cup \{A \equiv \{a | (\mathcal{T}, CT, QT) \models a : A\} | A \in \mathcal{N}\}$.
3. $\mathcal{T}^T = \mathcal{T}^T \cup \{\exists f.\top \sqsubseteq \{a | \exists b, (\mathcal{T}, CT, QT) \models (a, b) : r\} | r \in \mathcal{N}\}$.
4. $\mathcal{T}^T = \mathcal{T}^T \cup \{\exists r^-.\top \sqsubseteq \{b | \exists a, (\mathcal{T}, CT, QT) \models (a, b) : r\} | r \in \mathcal{N}\}$.
5. $\mathcal{T}^T = \mathcal{T}^T \cup \{\{a\} \sqsubseteq \forall r.\{b | (\mathcal{T}, CT, QT) \models (a, b) : r\} | r \in \mathcal{N}\}$.
6. $\mathcal{T}^T = \mathcal{T}^T \cup \{\{b\} \sqsubseteq \forall r^-.\{a | (\mathcal{T}, CT, QT) \models (a, b) : r\} | r \in \mathcal{N}\}$.
7. *Let $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(\mathcal{T}^T) = (\mathcal{T}'', CT'', QT'')$*
8. $\mathcal{T}' = \mathcal{T} \cup \mathcal{T}''$, $CT' = CT \cup CT''$, $QT' = QT \cup QT''$.

From Def. 7 the incremental pattern is quite obvious: the $(\mathcal{T}, CT, QT)$ is the permanent $\mathcal{EL}^{++}_{\mathcal{CQ}}$ transformation, while the $(\mathcal{T}'', CT'', QT'')$ is the temporary $\mathcal{EL}^{++}_{\mathcal{CQ}}$ transformation.

The following theorem describes the quality of approximate closed world reasoning with approximate NBox Internalization.

Theorem 6. ***(Soundness)*** *For any N-Ontology $\mathcal{O}^{\mathcal{N}} = (\mathcal{T}, \mathcal{A}, \mathcal{N})$, if for any concept $A \in \mathcal{N}$ and any individual a,*
$A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}((\mathcal{T}, \mathcal{A})) \models a : A$ iff $(\mathcal{T}, \mathcal{A}) \models a : A$ *and for any role $r \in \mathcal{N}$ and any two individual a, b, $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}((\mathcal{T}, \mathcal{A})) \models (a, b) : r$, then we have, for any two basic concepts A and B, $\mathcal{O}^{\mathcal{N}} \models A \sqsubseteq B$ if $ANI(\mathcal{O}^{\mathcal{N}}) \models A \sqsubseteq B$.*

Proof. According to Theorem 4, $\mathcal{O}^{\mathcal{N}} \models A \sqsubseteq B$ *iff* $NI(\mathcal{O}^{\mathcal{N}}) \models A \sqsubseteq B$. According to Theorem 2, $NI(\mathcal{O}^{\mathcal{N}}) \models A \sqsubseteq B$ if $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(NI(\mathcal{O}^{\mathcal{N}})) \models A \sqsubseteq B$. Then it is sufficient to prove that when the condition of the theorem holds, $ANI(\mathcal{O}^{\mathcal{N}}) = A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(NI(\mathcal{O}^{\mathcal{N}}))$.

We construct a TBox $\mathcal{T}^T$ as follows:

1. $\mathcal{T}^T$ is initialized to $\emptyset$.
2. $\mathcal{T}^T = \mathcal{T}^T \cup \{A \equiv \{a|(\mathcal{T},\mathcal{A}) \models a : A\}|A \in \mathcal{N}\}$.
3. $\mathcal{T}^T = \mathcal{T}^T \cup \{\exists r.\top \sqsubseteq \{a|\exists b, (\mathcal{T},\mathcal{A}) \models (a,b) : r\}|r \in \mathcal{N}\}$.
4. $\mathcal{T}^T = \mathcal{T}^T \cup \{\exists r^-.\top \sqsubseteq \{b|\exists a, (\mathcal{T},\mathcal{A}) \models (a,b) : r\}|r \in \mathcal{N}\}$.
5. $\mathcal{T}^T = \mathcal{T}^T \cup \{\{a\} \sqsubseteq \forall r.\{b|(\mathcal{T},\mathcal{A}) \models (a,b) : r\}|r \in \mathcal{N}\}$.
6. $\mathcal{T}^T = \mathcal{T}^T \cup \{\{b\} \sqsubseteq \forall r^-.\{a|(\mathcal{T},\mathcal{A}) \models (a,b) : r\}|r \in \mathcal{N}\}$.

According to Def.6, we have $NI(\mathcal{O}^{\mathcal{N}}) = (\mathcal{T} \cup \mathcal{T}^T, \mathcal{A})$.

Suppose $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}((\mathcal{T},\mathcal{A})) = (\mathcal{T}_1, CT_1, QT_1)$ and $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(\mathcal{T}^T) = (\mathcal{T}_2, CT_2, QT_2)$. From Def.2 it is easy to see that $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(NI(\mathcal{O}^{\mathcal{N}})) = A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}((\mathcal{T} \cup \mathcal{T}^T, \mathcal{A})) = (\mathcal{T}_1 \cup \mathcal{T}_2, CT_1 \cup CT_2, QT_1 \cup QT_2)$.

If for any concept $A \in \mathcal{N}$ and any individual a, $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}((\mathcal{T},\mathcal{A})) \models a : A$ *iff* $(\mathcal{T},\mathcal{A}) \models a : A$ and for any role $r \in \mathcal{N}$ and any two individual a, b, $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}((\mathcal{T},\mathcal{A})) \models (a,b) : r$, according to Def. 7 we have $ANI(\mathcal{O}^{\mathcal{N}}) = (\mathcal{T}_1 \cup \mathcal{T}_2, CT_1 \cup CT_2, QT_1 \cup QT_2)$ as well.

Altogether, we have $ANI(\mathcal{O}^{\mathcal{N}}) = A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(NI(\mathcal{O}^{\mathcal{N}}))$. Thus the theorem can be proved. □

This theorem indicates that, when the instance retrieval of NBox predicates is complete and sound, the approximate closed world reasoning is always soundness-guaranteed.

When we know that the approximate reasoning is sound and complete for such instance retrieval we can use approximate reasoning to infer them. This is useful in many applications where the checking of complete and soundness of NBox predicates instance retrieval is much easier than closed world reasoning on them. For example, when the NBox predicates only appear as the left-hand-side of some axioms, their instances are solely determinted by the ABox. Thus approximate reasoning is always sound and complete for them. Also, this is useful in applications where the closed world reasoning is more frequent and dynamic than the reasoning on the original ontology. In this case, one can verify the soundness and completeness on the NBox predicates in offline time and provide closed world reasoning online.

Thanks for the efficient incremental reasoning provided by $\mathcal{EL}^{++}$ [7], subsequent reasoning on $ANI(\mathcal{O}^{\mathcal{N}})$ can benefit from the reusing of existing results on $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(\mathcal{O}^{\mathcal{N}})$. Also, the overall approximate reasoning is still tractable.

When we don't know whether the approximate reasoning is complete (it is always sound), we can use a complete and sound reasoner such as Pellet or HermiT, to compute $NI(\mathcal{O}^{\mathcal{N}})$, then doing approximate reasoning on $A_{fn,\mathcal{EL}^{++}_{\mathcal{CQ}}}(NI(\mathcal{O}^{\mathcal{N}}))$.

6 Conclusion

In this paper we tried to address the issue of doing ontology reasoning with CWA. Different from the existing DBox approach, our solution follows from the idea of Negation As Failure and extends the syntax and semantics of classical ontology with a Negation As Failure Box (NBox). We showed that closed world reasoning over NBox-closed ontology can be reduced to reasoning over a classical ontology. In order to speed up this two-phase reduction-reasoning paradigm, we extended the existing syntactic approximation to support approximate closed world reasoning.

In the future, we will work on implementation and evaluation of our approach, as well as extending the existing completion rules to support more reasoning patterns.

References

1. Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language: Profiles. W3c working draft, W3C (October 2008)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
3. Ren, Y., van Deemter, K., jeff Z. Pan: Charting the Potential of Description Logic for the Generation of Referring Expressions. In: Proc. of 6th Internaltional Conference on Natural Langauge Generation(INLG-2010). (2010)
4. Tao, J., Sirin, E., Bao, J., McGuinness, D.: Integrity constraints in owl. In: AAAI2010: Proceedings of the AAAI Conference on Artificial Intelligence. (2010)
5. Seylan, I., Franconi, E., De Bruijn, J.: Effective query rewriting with ontologies over dboxes. In: IJCAI'09: Proceedings of the 21st international joint conference on Artifical intelligence, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2009) 923–929
6. Ren, Y., Pan, J.Z., Zhao, Y.: Towards Soundness Preserving Approximation for ABox Reasoning of OWL2. In: the Proc. of the International Description Logic Workshop (DL2010). (2010)
7. Suntisrivaraporn, B.: Module Extraction and Incremental Classification: A Pragmatic Approach for $\mathcal{EL}^+$ Ontologies. In Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M., eds.: Proceedings of the 5th European Semantic Web Conference (ESWC'08). Volume 5021 of Lecture Notes in Computer Science., Springer-Verlag (2008) 230–244
8. Horrocks, I., Kutz, O., Sattler, U.: The Even More Irresistible SROIQ. In: KR 2006. (2006)
9. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ Envelope. In: Proceedings IJCAI-05. (2005)
10. Baader, F., Brandt, S., Lutz, C.: Pushing the EL Envelope Further. In Clark, K., Patel-Schneider, P.F., eds.: In OWLED-2008. (2008)
11. Hollunder, B., Nutt, W., Schmidt-Schauß, M.: Subsumption Algorithms for Concept Description Languages. In: ECAI-90, Pitman Publishing (1990) 348–353
12. Ren, Y., Pan, J.Z., Zhao, Y.: Soundness Preserving Approximation for TBox Reasoning. In: the Proc. of the 25th AAAI Conference Conference (AAAI2010). (2010)