Secrecy-Preserving Query Answering in $\mathcal{ELH}$ Knowledge Bases

Gopalakrishnan Krishnasamy Sivaprakasam and Giora Slutzki
Department of Computer Science, Iowa State University, Ames, U.S.A.

Keywords: Knowledge Representation and Reasoning, Ontologies, Privacy and security, Semantic Web.

Abstract: In this paper we study Secrecy-Preserving Query Answering problem under Open World Assumption (OWA) for $\mathcal{ELH}$ Knowledge Bases (KBs). We employ two tableau procedures designed to compute some consequences of ABox ($\mathcal{A}$) and TBox ($\mathcal{T}$) denoted by $\mathcal{A}^*$ and $\mathcal{T}^*$ respectively. A secrecy set of a querying agent is subset $\mathbb{S}$ of $\mathcal{A}^* \cup \mathcal{T}^*$ which the agent is not allowed to access. An envelope is a superset of the secrecy set which provides logical protection to the secrecy set against the reasoning of the querying agent. Once envelopes are computed, they are used to efficiently answer assertional and GCI queries without compromising the secret information in $\mathbb{S}$. Answering GCI queries while preserving secrecy has not been studied in the current literature. When the querying agent asks a query q, the reasoner answers "Yes" if KB $\models q$ and q does not belong to the envelopes; otherwise, the reasoner answers "Unknown". Being able to answer "Unknown" plays a key role in protecting secrecy under OWA. Since we are not computing all the consequences of the KB, answers to the queries based on just $\mathcal{A}^*$ and $\mathcal{T}^*$ could be erroneous. To fix this problem, we further augment our algorithms to make the query answering procedure foolproof.

1 INTRODUCTION

The explosive growth in online banking activities, social networks, web based travel services and other internet based business and homeland security applications contain massive amounts of private details of users, administrators, service providers and governmental agencies. This contributes, on one hand, to unprecedented levels of information sharing and, on the other hand, to grave concerns about privacy and confidentiality of communication between WWW users. It will be an indispensable aspect of future web based service industry that private information while being shared must remain inviolate. In literature, most of the approaches dealing with "information protection" are based on access control mechanisms. For semantic web applications, the authors of (Kagal et al., 2003) have proposed policy languages to represent obligation and delegation policies based on access control approach. Biskup et al. in (Biskup and Weibert, 2008; Biskup and Tadros, 2012) studied secrecy in incomplete databases using controlled query evaluation (CQE). Since description logics (DLs) underlie web ontology languages (OWLs), recently researchers have shown an interest in studying secrecy-preserving reasoning in DL knowledge bases (KBs).

In (Bao et al., 2007; Tao et al., 2010; Tao et al., 2014), the authors have developed a secrecy framework that attempts to satisfy the following competing goals: (a) it protects secret information and (b) queries are answered as informatively as possible (subject to satisfying property (a)). The notion of an *envelope* to hide secret information against logical inference was first defined and used in (Tao et al., 2010). Further, in (Tao et al., 2014), Tao et al., introduced a more elaborate conceptual framework for secrecy-preserving query answering (SPQA) under Open World Assumption (OWA) with multiple querying agents. This approach is based on OWA and (so far) it has been restricted to instance-checking queries. Specifically, in (Bao et al., 2007; Tao et al., 2010; Tao et al., 2014) the main idea was to utilize the secret information within the reasoning process, but then answering "Unknown" whenever the answer is truly unknown or in case the true answer could compromise confidentiality.

The motivation for this work is that popular ontologies like GALEN, GO and SNOMED that can be viewed as KBs defined in languages belong to $\mathcal{EL}$ family. In addition, a number of studies were reported in conjunctive query answering, reasoning and classifications in $\mathcal{ELH}$ and its extensions, see (Bienvenu et al., 2013; Delaitre and Kazakov, 2009).

In this paper we extend the work of Tao et al., reported in (Tao et al., 2010), to the $\mathcal{ELH}$ language. In addition to the extension, we make several new con-

Sivaprakasam, G. and Slutzki, G.
Secrecy-Preserving Query Answering in $\mathcal{ELH}$ Knowledge Bases.
DOI: 10.5220/0005709701490159
In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART 2016) - Volume 2*, pages 149-159
ISBN: 978-989-758-172-4

tributions. First, we study secrecy in the context of assertions as well as general concept inclusions (GCIs). To the best of our knowledge, secrecy-preserving reasoning for GCIs has not been studied before. As a first step in constructing SPQA system, we design two tableau algorithms to compute finite sets $\mathcal{T}^*$ and then $\mathcal{A}^*$, of consequences of the TBox $\mathcal{T} \cup \mathcal{R}^*$ and the KB $\langle \mathcal{A}, \mathcal{T}^*, \mathcal{R}^* \rangle$ respectively, restricted to individuals and concepts that actually occur in the given KB $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ and an extra "auxiliary" set of concepts defined over the signature of Σ. The approach to constructing SPQA system presented in this paper is quite different from (Tao et al., 2010). In (Tao et al., 2010), the KB and envelope are expanded with new queries. This makes the subsequent query answering step more and more complicated. In general, the sets of all assertional consequences and GCI consequences of a given $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ may be infinite. By forcing the tableau algorithms to compute the consequences (both assertions and GCIs) of KB restricted to individuals and subconcepts that occur in a given prescribed set, we obtain finite $\mathcal{A}^*$ and $\mathcal{T}^*$ that in fact can be computed efficiently in polynomial time. These sets, once computed, remain fixed and are not modified. The two tableau algorithms are sound and complete under the restrictions stated above, see sections 3.1 and 3.2. Since the sets $\mathcal{A}^*$ and $\mathcal{T}^*$ do not contain all the consequences of the KB, in order to answer user queries we have designed recursive algorithms which break the queries into smaller assertions or GCIs all the way until the information in the sets $\mathcal{A}^*$ and $\mathcal{T}^*$ can be used. In effect, we have split the task of query answering into two parts: in the first part we compute all the consequences of Σ restricted to concepts and individuals that occur in Σ, in the second part we use a recursive algorithm to evaluate more complex queries with the base case that has been computed in the first part.

In more detail, starting from the secrecy sets $\mathbb{S}_{\mathcal{A}}$ (of assertions) and $\mathbb{S}_{\mathcal{T}}$ (of GCIs), we compute finite sets of assertions and GCIs, viz., the *envelopes* $\mathbb{E}_{\mathcal{A}} \subseteq \mathcal{A}^*$ of $\mathbb{S}_{\mathcal{A}}$ and $\mathbb{E}_{\mathcal{T}} \subseteq \mathcal{T}^*$ of $\mathbb{S}_{\mathcal{T}}$ respectively. These envelopes are computed by two tableau algorithms based on the idea of inverting the expansion rules of two tableau algorithms listed in Figures 1 and 2. The idea behind the envelope concept is that no expression in the envelope can be logically deduced from information outside the envelope. Once such envelopes are computed, the answers to the queries are censored whenever the queries belong to the envelopes. Since, generally, an envelope for a given secrecy set is not unique, the developer can force the algorithm to output a specific envelope from the available choices satisfying the needs of application domain, company policy, social obligations and user preferences.

Next, we discuss query answering procedures which allow us answer queries without revealing secrets. Usually in SPQA framework queries are answered by checking their membership (a) in $\mathcal{A}^* \setminus \mathbb{E}_{\mathcal{A}}$ if the query is an assertion; and (b) in $\mathcal{T}^* \setminus \mathbb{E}_{\mathcal{T}}$ if the query is a GCI. Since $\mathcal{A}^*$ and $\mathcal{T}^*$ do not contain all the statements entailed by Σ, we need to extend the query answering procedure from just membership checking. Towards that end we designed two recursive algorithms to answer more complicated assertion and GCI queries. To answer an assertion query q, the algorithm first checks if $q \in \mathcal{A}^* \setminus \mathbb{E}_{\mathcal{A}}$ in which case the answer is "Yes"; otherwise, the given query is broken into subqueries based on the constructors, and the algorithm is applied recursively on the subqueries, see section 5. This query answering procedure runs in polynomial time in the size of the KB and the query q. Similar approach is used to answer GCI queries.

2 SYNTAX AND SEMANTICS

A vocabulary of $\mathcal{ELH}$ is a triple $< N_O, N_C, N_R >$ of countably infinite, pairwise disjoint sets. The elements of N_O are called *object* (or *individual) names*, the elements of N_C are called *concept names* and the elements of N_R are called *role names*. The set of $\mathcal{ELH}$ *concepts* is denoted by $\mathcal{C}$ and is defined by the following rules

$$C ::= A \mid \top \mid C \sqcap D \mid \exists r.C$$

where $A \in N_C$, $r \in N_R$, $\top$ denotes the "*top concept*", and $C, D \in \mathcal{C}$. *Assertions* are expressions of the form $C(a)$ or $r(a,b)$, *general concept inclusions (GCIs)* are expressions of the form $C \sqsubseteq D$ and *role inclusions* are expressions of the form $r \sqsubseteq s$ where $C, D \in \mathcal{C}$, $r, s \in N_R$ and $a, b \in N_O$. The semantics of $\mathcal{ELH}$ concepts is specified, as usual, by an *interpretation* $\mathcal{I} = \left\langle \Delta, \cdot^{\mathcal{I}} \right\rangle$ where Δ is the *domain* of the interpretation, and $\cdot^{\mathcal{I}}$ is an *interpretation function* mapping each $a \in N_O$ to an element $a^{\mathcal{I}} \in \Delta$, each $A \in N_C$ to a subset $A^{\mathcal{I}} \subseteq \Delta$, and each $r \in N_R$ to a binary relation $r^{\mathcal{I}} \subseteq \Delta \times \Delta$. The interpretation function $\cdot^{\mathcal{I}}$ is extended inductively to all $\mathcal{ELH}$ concepts in the usual manner:

$$\top^{\mathcal{I}} = \Delta; \;\; (C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}};$$
$$(\exists r.C)^{\mathcal{I}} = \{d \in \Delta \mid \exists e \in C^{\mathcal{I}} : (d,e) \in r^{\mathcal{I}}\}.$$

An *Abox* $\mathcal{A}$ is a finite, non-empty set of assertions. A *TBox* $\mathcal{T}$ is a finite set of GCIs and an *RBox* $\mathcal{R}$ is a finite set of role inclusions. An $\mathcal{ELH}$ KB is a triple $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ where $\mathcal{A}$ is an ABox, $\mathcal{T}$ is a TBox and

$\mathcal{R}$ is an RBox. Let $\mathcal{I} = \left\langle \Delta, \cdot^{\mathcal{I}} \right\rangle$ be an interpretation, $C, D \in \mathcal{C}$, $r, s \in N_R$ and $a, b \in N_O$. We say that $\mathcal{I}$ *satisfies* $C(a)$, $r(a,b)$, $C \sqsubseteq D$ or $r \sqsubseteq s$, notation $\mathcal{I} \models C(a)$, $\mathcal{I} \models r(a,b)$, $\mathcal{I} \models C \sqsubseteq D$ or $\mathcal{I} \models r \sqsubseteq s$ if, respectively, $a^{\mathcal{I}} \in C^{\mathcal{I}}$, $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ or $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$. $\mathcal{I}$ is a *model* of Σ, notation $\mathcal{I} \models \Sigma$, if $\mathcal{I}$ satisfies all the assertions in $\mathcal{A}$, all the GCIs in $\mathcal{T}$ and all the role inclusions in $\mathcal{R}$. Let α be an assertion, a GCI or a role inclusion. We say that Σ *entails* α, notation $\Sigma \models \alpha$, if all models of Σ satisfy α.

3 COMPUTATION OF $\mathcal{A}^*$ AND $\mathcal{T}^*$

Let $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ be an $\mathcal{ELH}$ KB. In this section, we give two tableau algorithms that compute $\mathcal{A}^*$, a set of assertional consequence of Σ, and $\mathcal{T}^*$ a set of GCI consequences of Σ, both restricted to concepts that occur in Σ. We assume that all RBoxes are acyclic. Before computing $\mathcal{T}^*$ and $\mathcal{A}^*$, we compute $\mathcal{R}^* = \mathcal{R}^+ \cup \mathcal{R}^\circ$, where $\mathcal{R}^+$ is the transitive closure of $\mathcal{R}$ with respect to role inclusion and $\mathcal{R}^\circ = \{r \sqsubseteq r \mid r \text{ occurs in } \Sigma\}$. As an example, consider a KB $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ where ABox $\mathcal{A} = \{A(a), \exists m.B(c)\}$, TBox $\mathcal{T} = \{A \sqsubseteq \exists n.D\}$ and RBox $\mathcal{R} = \{r \sqsubseteq s,\ p \sqsubseteq q,\ u \sqsubseteq v,\ s \sqsubseteq u\}$. Then, $\mathcal{R}^* = \mathcal{R} \cup \{s \sqsubseteq v,\ r \sqsubseteq u,\ r \sqsubseteq v\} \cup \{m \sqsubseteq m, n \sqsubseteq n, r \sqsubseteq r, s \sqsubseteq s, p \sqsubseteq p, q \sqsubseteq q, u \sqsubseteq u, v \sqsubseteq v\}$. $\mathcal{R}^*$ is easily computed in polynomial time and we omit the details.

3.1 Computation of $\mathcal{T}^*$

Denote by N_Σ the set of all concept names and role names occurring in Σ and let $\mathbb{S}$ be a finite set of concepts over the symbol set N_Σ. Let $\mathcal{C}_{\Sigma,\mathbb{S}}$ be the set of all subconcepts of concepts that occur in either $\mathbb{S}$ or Σ. Given Σ and $\mathcal{C}_{\Sigma,\mathbb{S}}$, we describe a procedure that computes $\mathcal{T}^*$, a set of GCI consequences of the given KB Σ (restricted to concepts in $\mathcal{C}_{\Sigma,\mathbb{S}}$). That is, $\mathcal{T}^* = \{C \sqsubseteq D \mid C, D \in \mathcal{C}_{\Sigma,\mathbb{S}} \text{ and } \Sigma \models C \sqsubseteq D\}$. This procedure is similar to the calculus presented in (Kazakov et al., 2014) (designed for $\mathcal{EL}^+$).

Let $AX_{\mathcal{T}} = \{C \sqsubseteq C, C \sqsubseteq \top, \top \sqsubseteq \top \mid C \in \mathcal{C}_{\Sigma,\mathbb{S}}\}$. $\mathcal{T}^*$ is initialized as $AX_{\mathcal{T}}$ and then expanded by exhaustively applying expansion rules listed in Figure 1. The $\mathrm{T}_{\sqsubseteq}$-rule derives a GCI based on transitivity of subsumption. $\mathrm{T}_{\sqcap}^-$-rule derives new GCIs by decomposing conjunction concepts into its two conjuncts. The $\mathrm{T}_{\sqcap}^+$-rule is just the "opposite" of the $\mathrm{T}_{\sqcap}^-$-rule. Finally, T_H^+-rule derives GCIs based on concept and role inclusions.

A TBox is *completed* if no expansion rule in Figure 1 is applicable to it. We denote by $\Lambda_{\mathcal{T}}$ the *algorithm* which, given Σ, $\mathcal{C}_{\Sigma,\mathbb{S}}$ and $\mathcal{R}^*$, nondeterministically applies expansion rules in Figure 1 until no further applications are possible. Since $\Lambda_{\mathcal{T}}$ has been restricted to derive GCIs whose left and right hand side concept expressions occur in $\mathcal{C}_{\Sigma,\mathbb{S}}$, the size of the $\mathcal{T}^*$ is at most a polynomial in the size of its input. Hence, the running time of $\Lambda_{\mathcal{T}}$ is polynomial in $|\,\Sigma\,| + |\,\mathcal{C}_{\Sigma,\mathbb{S}}\,|$. The correctness of $\Lambda_{\mathcal{T}}$ can be shown by proving soundness and completeness of $\Lambda_{\mathcal{T}}$. The soundness proof is obvious.

$\mathrm{T}_{\sqsubseteq}$ – rule : if $C \sqsubseteq D \in \mathcal{T}^*$, $D \sqsubseteq E \in \mathcal{T}$
and $C \sqsubseteq E\ \notin \mathcal{T}^*$,
then $\mathcal{T}^* := \mathcal{T}^* \cup \{C \sqsubseteq E\}$;
$\mathrm{T}_{\sqcap}^-$ – rule : if $C \sqsubseteq D \sqcap E \in \mathcal{T}^*$, and $C \sqsubseteq D \notin \mathcal{T}^*$
or $C \sqsubseteq E\ \notin \mathcal{T}^*$,
then $\mathcal{T}^* := \mathcal{T}^* \cup \{C \sqsubseteq D, C \sqsubseteq E\}$;
$\mathrm{T}_{\sqcap}^+$ – rule : if $C \sqsubseteq D$, $C \sqsubseteq E \in \mathcal{T}^*$, $D \sqcap E \in \mathcal{C}_{\Sigma,\mathbb{S}}$
and $C \sqsubseteq D \sqcap E\ \notin \mathcal{T}^*$,
then $\mathcal{T}^* := \mathcal{T}^* \cup \{C \sqsubseteq D \sqcap E\}$;
T_H^+ – rule : if $C \sqsubseteq \exists r.D$, $D \sqsubseteq E \in \mathcal{T}^*$, $r \sqsubseteq s \in \mathcal{R}^*$,
$\exists s.E \in \mathcal{C}_{\Sigma,\mathbb{S}}$ and $C \sqsubseteq \exists s.E\ \notin \mathcal{T}^*$,
then $\mathcal{T}^* := \mathcal{T}^* \cup \{C \sqsubseteq \exists s.E\}$.

Figure 1: TBox Tableau expansion rules.

Example 1. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ be a $\mathcal{ELH}$ KB, where $\mathcal{A} = \{C(a), r(b,a), \exists u.A(d)\}$, $\mathcal{T} = \{A \sqsubseteq B, C \sqsubseteq D \sqcap E, F \sqsubseteq \exists u.B\}$ and $\mathcal{R} = \{u \sqsubseteq v\}$. Then, $\mathcal{R}^* = \{r \sqsubseteq r, u \sqsubseteq u, v \sqsubseteq v, u \sqsubseteq v\}$. Thus, applying rules in Figure 1 to $\mathcal{T}$, we get $\{\top \sqsubseteq \top, A \sqsubseteq \top, C \sqsubseteq C, \exists u.A \sqsubseteq \exists u.A, \exists u.A \sqsubseteq \exists u.B, C \sqsubseteq D, C \sqsubseteq D \sqcap E\} \subseteq \mathcal{T}^*$.* □

To prove the completeness of $\Lambda_{\mathcal{T}}$, we define the *canonical interpretation* $\mathcal{J} = \left\langle \Delta, \cdot^{\mathcal{J}} \right\rangle$ for a completed TBox $\mathcal{T}^*$ and an RBox $\mathcal{R}^*$ as follows:

$$\begin{aligned} \Delta &= \{w_C \mid C \in \mathcal{C}_{\Sigma,\mathbb{S}}\}; \\ \top^{\mathcal{J}} &= \Delta; \\ \text{for } A \in N_C,\ A^{\mathcal{J}} &= \{w_C \mid C \sqsubseteq A \in \mathcal{T}^*\}; \\ \text{for } r \in N_R,\ r^{\mathcal{J}} &= \{(w_C, w_D) \mid C \sqsubseteq \exists r.D \in \mathcal{T}^*\} \cup \\ &\quad \textstyle\bigcup_{u \sqsubseteq r \in \mathcal{R}^*} u^{\mathcal{J}}. \end{aligned}$$

The interpretation function $\cdot^{\mathcal{J}}$ is extended to concept expressions as usual. To prove that $\mathcal{J}$ is a model of $\mathcal{T}^*$, we need the following definition and technical lemma.

Definition 1. *Let $\mathcal{J}$ be the canonical interpretation and u a role name that occurs in Σ. u is said to be minimal with respect to $(w_G, w_H) \in \Delta \times \Delta$ if*

1) $(w_G, w_H) \in u^{\mathcal{J}}$ and

2) there is no v that occurs in $\mathcal{R}$ such that $v \neq u$, $(w_G, w_H) \in v^{\mathcal{J}}$ and $v \sqsubseteq u \in \mathcal{R}^*$.

Lemma 1. *Let B, $C \in \mathcal{C}_{\Sigma,\mathbb{S}}$. Then,*

(a) $w_C \in C^{\mathcal{J}}$.
(b) $w_C \in B^{\mathcal{J}}$ *if and only if* $C \sqsubseteq B \in \mathcal{T}^*$.

Proof. (a) By induction on the structure of C.

- $C = A \in N_C$ or $C = \top$, the claim follows from the definition of $\mathcal{J}$.
- $C = D \sqcap E$. Then, $D \sqcap E \sqsubseteq D \sqcap E \in \mathcal{T}^*$ and by the $\mathrm{T}^{-}_{\sqcap}$-rule, we have $D \sqcap E \sqsubseteq D, D \sqcap E \sqsubseteq E \in \mathcal{T}^*$, whence $w_{D \sqcap E} \in D^{\mathcal{J}}$ and $w_{D \sqcap E} \in E^{\mathcal{J}}$, by inductive hypothesis. By the semantics of $\sqcap$, $w_{D \sqcap E} \in D^{\mathcal{J}} \cap E^{\mathcal{J}} = (D \sqcap E)^{\mathcal{J}}$.
- $C = \exists r.D$. Then, $\exists r.D \sqsubseteq \exists r.D \in \mathcal{T}^*$ and by the definition of $\mathcal{J}$, $(w_{\exists r.D}, w_D) \in r^{\mathcal{J}}$; also, by the inductive hypothesis, $w_D \in D^{\mathcal{J}}$. By the semantics of $\exists$, $w_{\exists r.D} \in (\exists r.D)^{\mathcal{J}}$.

(b) ($\Leftarrow$) By induction on the structure of B.

- $B \in N_C$. Then, $C \sqsubseteq B \in \mathcal{T}^*$ whence $w_C \in B^{\mathcal{J}}$, by the definition of $\mathcal{J}$.
- $B = \top$, the claim follows from the definition of $\mathcal{J}$.
- $B = D \sqcap E$. Then, $C \sqsubseteq D \sqcap E \in \mathcal{T}^*$. By $\mathrm{T}^{-}_{\sqcap}$-rule, $C \sqsubseteq D$, $C \sqsubseteq E \in \mathcal{T}^*$ implies $w_C \in D^{\mathcal{J}}$ and $w_C \in E^{\mathcal{J}}$, and by the inductive hypothesis whence $w_C \in (D \sqcap E)^{\mathcal{J}} = B^{\mathcal{J}}$, by the semantics of $\sqcap$.
- $B = \exists r.D$. We assume, $C \sqsubseteq \exists r.D \in \mathcal{T}^*$. Since $C, D \in \mathcal{C}_{\Sigma,\mathbb{S}}$, we have $w_C, w_D \in \Delta$. By the definition of $\mathcal{J}$, $(w_C, w_D) \in r^{\mathcal{J}}$. By part (a), $w_D \in D^{\mathcal{J}}$ hence $w_C \in (\exists r.D)^{\mathcal{J}} = B^{\mathcal{J}}$, by the semantics of $\exists$.

($\Rightarrow$) By induction on the structure of B.

- When $B \in N_C$, the claim follows from the definition of $\mathcal{J}$.
- $B = \top$, the claim follows from the definition of $AX_{\mathcal{T}}$.
- $B = D \sqcap E$. Then, $w_C \in (D \sqcap E)^{\mathcal{J}} \Rightarrow w_C \in D^{\mathcal{J}}$ and $w_C \in E^{\mathcal{J}} \Rightarrow C \sqsubseteq D, C \sqsubseteq E \in \mathcal{T}^*$, by inductive hypothesis. Since $D \sqcap E$ occurs in $\mathcal{C}_{\Sigma,\mathbb{S}}$, by the $\mathrm{T}^{+}_{\sqcap}$-rule, we have $C \sqsubseteq D \sqcap E = B \in \mathcal{T}^*$.
- $B = \exists r.D$. Then, $w_C \in (\exists r.D)^{\mathcal{J}} \Rightarrow$ there is an element $w_E \in \Delta$ such that $(w_C, w_E) \in r^{\mathcal{J}}$, $w_E \in D^{\mathcal{J}}$. By inductive hypothesis, $E \sqsubseteq D \in \mathcal{T}^*$. Now, we have two subcases depending on a "manner" in which (w_C, w_E) entered $r^{\mathcal{J}}$.
  - If r is minimal with respect to (w_C, w_E), then, by the definition of $\mathcal{J}$ and Definition 1, $C \sqsubseteq \exists r.E \in \mathcal{T}^*$. Since $r \sqsubseteq r \in \mathcal{R}^*$, by the T^{+}_{H}-rule, we have $C \sqsubseteq \exists r.D \in \mathcal{T}^*$. Hence, $C \sqsubseteq B \in \mathcal{T}^*$.
  - If r is not minimal, then $(w_C, w_E) \in u^{\mathcal{J}}$, $u \neq r$ and $u \sqsubseteq r \in \mathcal{R}^*$ for some u that occurs in $\mathcal{R}$. If u is minimal with respect to (w_C, w_E), then by previous case $C \sqsubseteq \exists u.E \in \mathcal{T}^*$ and by the T^{+}_{H}-rule, we have $C \sqsubseteq \exists r.D \in \mathcal{T}^*$. Hence, $C \sqsubseteq B \in \mathcal{T}^*$. If u is not minimal with respect to (w_C, w_E), since RBox $\mathcal{R}$ is acyclic, there exists a chain $v \sqsubseteq v_1 \sqsubseteq v_2 ...... \sqsubseteq v_k \sqsubseteq u$ in $\mathcal{R}$ such that v is minimal with respect to (w_C, w_E). Since $\mathcal{R}^*$ is the transitive closure of $\mathcal{R}$, $v \sqsubseteq r \in \mathcal{R}^*$. Again by the previous case, $C \sqsubseteq \exists v.E \in \mathcal{T}^*$. By T^{+}_{H}-rule, we have $C \sqsubseteq \exists r.D \in \mathcal{T}^*$. Hence, $C \sqsubseteq B \in \mathcal{T}^*$.

□

The following lemma claims that $\mathcal{J}$ satisfies $\mathcal{T}^*$ and $\mathcal{R}^*$. The proof is a consequence of Lemma 1

Lemma 2. $\mathcal{J} \models \mathcal{T}^* \cup \mathcal{R}^*$.

The completeness of $\Lambda_{\mathcal{T}}$ now follows by an easy argument.

Theorem 1. *Let Σ be a $\mathcal{ELH}$ KB and let $\mathcal{T}^*$ be the completed TBox. For any C, D $\in \mathcal{C}_{\Sigma,\mathbb{S}}$, if $\Sigma \models C \sqsubseteq D$ then $C \sqsubseteq D \in \mathcal{T}^*$.*

Proof. Suppose $C \sqsubseteq D \notin \mathcal{T}^*$, i.e., by part (b) of Lemma 1, $w_C \notin D^{\mathcal{J}}$. On the other hand by part (a) of Lemma 1, $w_C \in C^{\mathcal{J}}$ and this implies that $\mathcal{J} \not\models C \sqsubseteq D$. Since by Lemma 2, $\mathcal{J} \models \mathcal{T}^*$, and since $\mathcal{T} \subseteq \mathcal{T}^*$, we obtain $\Sigma \not\models C \sqsubseteq D$. □

3.2 Computation of $\mathcal{A}^*$

Let $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ be an $\mathcal{ELH}$ KB, $\mathcal{R}^*$ be defined as at the beginning of this section and $\mathcal{T}^*$ be the completed TBox as computed in Section 3.1. Also, let $\mathcal{O}_{\Sigma}$ be the set of individual names that occur in Σ and define $AX_{\mathcal{A}} = \{\top(a) \mid a \in \mathcal{O}_{\Sigma}\}$.

We outline the procedure that computes $\mathcal{A}^*$, the set of assertional consequences of Σ^* where $\Sigma^* = \langle \mathcal{A}, \mathcal{T}^*, \mathcal{R}^* \rangle$, restricted to the concepts and role names that occur in $\mathcal{C}_{\Sigma,\mathbb{S}}$ and Σ respectively.

$$\text{That is } \mathcal{A}^* = \{C(a) \mid C \in \mathcal{C}_{\Sigma,\mathbb{S}} \text{ and } \Sigma^* \models C(a)\} \cup \{r(a,b) \mid r \text{ occurs in } \Sigma \text{ and } \Sigma^* \models r(a,b)\}.$$

$\mathcal{A}^*$ is initialized as $\mathcal{A} \cup AX_{\mathcal{A}}$ and is expanded by exhaustively applying rules listed in Figure 2. $\mathrm{A}^{-}_{\sqcap}$-rule decomposes conjunctions, and the $\mathrm{A}_{\sqsubseteq}$-rule derives assertions based on the GCIs present in $\mathcal{T}^*$. To build new concept assertions whose concept expressions already occur in $\mathcal{C}_{\Sigma,\mathbb{S}}$, we use the $\mathrm{A}^{+}_{\sqcap}$ and $\mathrm{A}^{+}_{\exists}$-rules. Similarly, the $\mathrm{A}^{+}_{\exists H}$-rule derives concept assertions based on role inclusions. It is important to note that this procedure does not introduce any fresh individual names into $\mathcal{A}^*$. Thus some assertions of the form $\exists r.C(a)$ may not have "syntactic witnesses". Finally, the A_{H}-rule derives role assertions based on role inclusions.

An ABox is *completed* if no expansion rule in Figure 2 is applicable to it. We denote by $\Lambda_{\mathcal{A}}$ the *algorithm* which, given $\mathcal{A}$, $\mathcal{R}^*$, $\mathcal{T}^*$ and $\mathcal{C}_{\Sigma,\mathbb{S}}$, nondeterministic-ally applies expansion rules in Figure 2 until no further applications are possible. Since $\Lambda_{\mathcal{A}}$ derives only assertions involving concept expressions that occur in $\mathcal{C}_{\Sigma,\mathbb{S}}$, it is easy to see that the running time of $\Lambda_{\mathcal{A}}$ is polynomial in $|\,\Sigma\,| + |\,\mathcal{C}_{\Sigma,\mathbb{S}}\,|$.

$$
\begin{array}{rl}
\mathrm{A}_{\sqcap}^{-}-\text{rule}: & \text{if } C \sqcap D(a) \in \mathcal{A}^*, \text{ and} \\
 & C(a) \notin \mathcal{A}^* \text{ or } D(a) \notin \mathcal{A}^*, \\
 & \text{then } \mathcal{A}^* := \mathcal{A}^* \cup \{C(a), D(a)\}; \\
\mathrm{A}_{\sqcap}^{+}-\text{rule}: & \text{if } C(a),\ D(a) \in \mathcal{A}^*, \\
 & C \sqcap D \in \mathcal{C}_{\Sigma,\mathbb{S}} \text{ and } C \sqcap D(a) \notin \mathcal{A}^*, \\
 & \text{then } \mathcal{A}^* := \mathcal{A}^* \cup \{C \sqcap D(a)\}; \\
\mathrm{A}_{\exists}^{+}-\text{rule}: & \text{if } r(a,b),\ C(b) \in \mathcal{A}^*, \\
 & \exists r.C \in \mathcal{C}_{\Sigma,\mathbb{S}} \text{ and } \exists r.C(a) \notin \mathcal{A}^*, \\
 & \text{then } \mathcal{A}^* := \mathcal{A}^* \cup \{\exists r.C(a)\}; \\
\mathrm{A}_{\sqsubseteq}-\text{rule}: & \text{if } C(a) \in \mathcal{A}^*,\ C \sqsubseteq D \in \mathcal{T}^*, \\
 & \text{and } D(a) \notin \mathcal{A}^*, \\
 & \text{then } \mathcal{A}^* := \mathcal{A}^* \cup \{D(a)\}; \\
\mathrm{A}_{\exists H}^{+}-\text{rule}: & \text{if } \exists r.C(a) \in \mathcal{A}^*,\ r \sqsubseteq s \in \mathcal{R}^*, \\
 & C \sqsubseteq D \in \mathcal{T}^*, \\
 & \exists s.D \in \mathcal{C}_{\Sigma,\mathbb{S}} \text{ and } \exists s.D(a) \notin \mathcal{A}^*, \\
 & \text{then } \mathcal{A}^* := \mathcal{A}^* \cup \{\exists s.D(a)\}; \\
\mathrm{A}_{H}-\text{rule}: & \text{if } r(a,b) \in \mathcal{A}^*,\ r \sqsubseteq s \in \mathcal{R}^*, \text{ and} \\
 & s(a,b) \notin \mathcal{A}^*, \text{ then } \mathcal{A}^* := \mathcal{A}^* \cup \{s(a,b)\}.
\end{array}
$$

Figure 2: ABox Tableau expansion rules.

Example 2. *(Example 1 cont.) Recall that* $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ *be a* $\mathcal{ELH}$ *be the given KB,* $\mathcal{R}^*$ *the computed RBox and* $\mathcal{T}^*$ *the completed TBox. Then, by applying rules in Figure 2 to* $\mathcal{A}$ *and using* $\mathcal{T}^*$ *and* $\mathcal{R}^*$ *we get,*

$\mathcal{A}^* = \{\top(a), \top(b), \top(d), \exists u.A(d), \exists u.B(d), C(a), r(b,a), D(a), E(a), D \sqcap E(a)\}$. □

The correctness of $\Lambda_{\mathcal{A}}$ can be shown by proving its soundness and completeness. The soundness is obvious. To prove the completeness of $\Lambda_{\mathcal{A}}$, we first define the canonical interpretation $\mathcal{K} = \left\langle \Delta, \cdot^{\mathcal{K}} \right\rangle$ for a completed ABox $\mathcal{A}^*$. The definition of $\mathcal{K}$ is similar to the definition of canonical model $\mathcal{I}_{\mathcal{K}}$ presented in (Lutz et al., 2008). Define the witness set, $\mathcal{W} = \{w_C \mid C \in \mathcal{C}_{\Sigma,\mathbb{S}}\}$.

$$\Delta = \mathcal{O}_{\Sigma} \cup \mathcal{W};$$

$$a^{\mathcal{K}} = a, \text{where } a \in \mathcal{O}_{\Sigma};$$

$$\top^{\mathcal{K}} = \Delta;$$

for each $A \in N_C$,

$$A^{\mathcal{K}} = \{a \in \mathcal{O}_{\Sigma} \mid A(a) \in \mathcal{A}^*\} \cup \{w_C \in \mathcal{W} \mid C \sqsubseteq A \in \mathcal{T}^*\};$$

for each $r \in N_R, r^{\mathcal{K}} = \{(a,b) \in \mathcal{O}_{\Sigma} \times \mathcal{O}_{\Sigma} \mid r(a,b) \in \mathcal{A}^*\} \cup$

$$
\begin{array}{l}
\{(a, w_C) \in \mathcal{O}_{\Sigma} \times \mathcal{W} \mid \exists r.C(a) \in \mathcal{A}^*\} \cup \\
\{(w_C, w_D) \in \mathcal{W} \times \mathcal{W} \mid C \sqsubseteq \exists r.D \in \mathcal{T}^*\} \\
\cup \bigcup \{u^{\mathcal{K}} \mid u \sqsubseteq r \in \mathcal{R}^*\}.
\end{array}
$$

$\mathcal{K}$ is extended to compound concepts in the usual way. We argue that $\mathcal{K}$ is a model of $\mathcal{A}^*$, $\mathcal{T}^*$ and $\mathcal{R}^*$.

Lemma 3. *Let* $a, b \in \mathcal{O}_{\Sigma}$ *and suppose that the role name* r *occurs in* Σ*. If* $(a,b) \in r^{\mathcal{K}}$*, then* $r(a,b) \in \mathcal{A}^*$.

Proof. Assume the hypotheses. We prove the claim by induction on how $r(a,b)$ has been generated by $\Lambda_{\mathcal{A}}$. The base case, when $r(a,b) \in \mathcal{A}$, is trivial. Let $(a,b) \in u^{\mathcal{K}}$ with $u \sqsubseteq r \in \mathcal{R}^*$. Then by induction hypothesis, $u(a,b) \in \mathcal{A}^*$ and by applying the A_H-rule, we have $r(a,b) \in \mathcal{A}^*$. □

We state the following lemma whose proof is similar to the proof of Lemma 1.

Lemma 4. *Let* B, $C \in \mathcal{C}_{\Sigma,\mathbb{S}}$*. Then,*

(a) $w_C \in C^{\mathcal{K}}$.

(b) $w_C \in B^{\mathcal{K}}$ *if and only if* $C \sqsubseteq B \in \mathcal{T}^*$.

The following definition is similar to Definition 1, but is based on the canonical interpretation of the ABox $\mathcal{A}^*$.

Definition 2. *Let* $\mathcal{K}$ *be the canonical interpretation, and* u *a role name that occurs in* Σ*.* u *is said to be minimal with respect to* (a,b) *if*

1) $(a,b) \in u^{\mathcal{K}}$ *and*

2) there is no role name, v *that occurs in* $\mathcal{R}$ *such that* $v \neq u$, $(a,b) \in v^{\mathcal{K}}$ *and* $v \sqsubseteq u \in \mathcal{R}^*$.

Lemma 5. *Let* $a \in \mathcal{O}_{\Sigma}$ *and* $B \in \mathcal{C}_{\Sigma,\mathbb{S}}$*. If* $a \in B^{\mathcal{K}}$*, then* $B(a) \in \mathcal{A}^*$.

Proof. By induction on the structure of B.

- When $B \in N_C$, the claim follows directly from the definition of $\mathcal{K}$.
- When $B = \top$, the claim follows from the definition of $AX_{\mathcal{A}}$.
- $B = C \sqcap D$. Then, $a \in (C \sqcap D)^{\mathcal{K}} \Rightarrow a \in C^{\mathcal{K}}$ and $a \in D^{\mathcal{K}} \Rightarrow C(a), D(a) \in \mathcal{A}^*$, by inductive hypothesis. Since $C \sqcap D$ occurs in $\mathcal{C}_{\Sigma,\mathbb{S}}$, by the $\mathrm{A}_{\sqcap}^{+}$-rule, we have $C \sqcap D(a) = B(a) \in \mathcal{A}^*$.
- $B = \exists r.C$. Then, $a \in (\exists r.C)^{\mathcal{K}}$ implies that there is an element $b \in \Delta$ such that $(a,b) \in r^{\mathcal{K}}$ and $b \in C^{\mathcal{K}}$. There are two cases.
  - $b \in \mathcal{O}_{\Sigma}$. Since r occurs in Σ and C occurs in $\mathcal{C}_{\Sigma,\mathbb{S}}$, by Lemma 3, we have $r(a,b) \in \mathcal{A}^*$ and by the inductive hypothesis, $C(b) \in \mathcal{A}^*$. Since $\exists r.C$ occurs in $\mathcal{C}_{\Sigma,\mathbb{S}}$, by the $\mathrm{A}_{\exists}^{+}$-rule, we have $\exists r.C(a) = B(a) \in \mathcal{A}^*$.

- $b = w_D \in \mathcal{W}$ for some $D \in \mathcal{C}_{\Sigma,\mathbb{S}}$. Then, we have $(a, w_D) \in r^{\mathcal{K}}$ and $w_D \in C^{\mathcal{K}}$. By part (b) of Lemma 1, $D \sqsubseteq C \in \mathcal{T}^*$. Now, we have two subcases depending on a manner in which (a, w_D) entered $r^{\mathcal{K}}$.
  - If r is minimal with respect to (a, w_D), then, by the definition of $\mathcal{K}$ and Definition 2, $\exists r.D(a) \in \mathcal{A}^*$. Since $r \sqsubseteq r \in \mathcal{R}^*$, by the $\mathrm{A}^+_{\exists H}$-rule, we have $\exists r.C(a) \in \mathcal{A}^*$, i.e., $B(a) \in \mathcal{A}^*$.
  - If r is not minimal, then $(a, w_D) \in u^{\mathcal{K}}$, $u \neq r$ and $u \sqsubseteq r \in \mathcal{R}^*$ for some u that occurs in $\mathcal{R}$. If u is minimal with respect to (a, w_D), then by previous case $\exists u.D(a) \in \mathcal{A}^*$. By $\mathrm{A}^+_{\exists H}$-rule, we have $\exists r.C(a) \in \mathcal{A}^*$. Hence, $B(a) \in \mathcal{A}^*$. If u is not minimal with respect to (a, w_D), since RBox $\mathcal{R}$ is acyclic, there exists a chain $v \sqsubseteq v_1 \sqsubseteq v_2 ...... \sqsubseteq v_k \sqsubseteq u$ in $\mathcal{R}$ such that v is minimal with respect to (a, w_E). Since $\mathcal{R}^*$ is the transitive closure of $\mathcal{R}$, $v \sqsubseteq r \in \mathcal{R}^*$. Again by the previous case, $\exists v.D(a) \in \mathcal{A}^*$. By $\mathrm{A}^+_{\exists H}$-rule, we have $\exists r.C(a) \in \mathcal{A}^*$, i.e., $B(a) \in \mathcal{A}^*$.

□

The next lemma is, roughly, the inverse of Lemma 5 and its proof is omitted.

Lemma 6. *If $B(a) \in \mathcal{A}^*$, then $a \in B^{\mathcal{K}}$.*

In the following we prove that $\mathcal{K}$ satisfies $\mathcal{A}^*$, $\mathcal{T}^*$ and $\mathcal{R}^*$.

Lemma 7. $\mathcal{K} \models \mathcal{A}^* \cup \mathcal{T}^* \cup \mathcal{R}^*$.

Proof. It follows immediately from the definition of $\mathcal{K}$ that $\mathcal{K} \models \mathcal{R}^*$. Next, we show that $\mathcal{K}$ satisfies $\mathcal{A}^*$. $C(a) \in \mathcal{A}^*$; then, by Lemma 6, $a \in C^{\mathcal{K}}$, i.e., $\mathcal{K} \models C(a)$. For $r(a,b) \in \mathcal{A}^*$, $\mathcal{K} \models r(a,b)$, by the definition of $\mathcal{K}$. Hence $\mathcal{K} \models \mathcal{A}^*$.

Now, we show that $\mathcal{K}$ satisfies $\mathcal{T}^*$. Let $F \sqsubseteq G \in \mathcal{T}^*$ and $a \in F^{\mathcal{K}}$. We have two cases.

- $a \in \mathcal{O}_\Sigma$. Then, by Lemma 5, $F(a) \in \mathcal{A}^*$. Since $\mathcal{A}^*$ is completed, by the $\mathrm{A}_{\sqsubseteq}$-rule, we get $G(a) \in \mathcal{A}^*$. By Lemma 6, $a \in G^{\mathcal{K}}$. Hence, $\mathcal{K} \models F \sqsubseteq G$.
- $a = w_C \in \mathcal{W}$ for some $C \in \mathcal{C}_{\Sigma,\mathbb{S}}$. This implies, by the definition of $\mathcal{K}$, that $C \sqsubseteq F \in \mathcal{T}^*$. Since $\mathcal{T}^*$ is completed, we have $C \sqsubseteq G \in \mathcal{T}^*$. Again by the definition of $\mathcal{K}$, $a \in G^{\mathcal{K}}$ which implies $\mathcal{K} \models F \sqsubseteq G$.

□

We are ready to prove the completeness of $\Lambda_{\mathcal{A}}$.

Theorem 2. *Let $\Sigma^* = \langle \mathcal{A}, \mathcal{T}^*, \mathcal{R}^* \rangle$ be a $\mathcal{ELH}$ KB as defined in section 3.2 and $\mathcal{A}^*$ the completed ABox. Suppose that $B \in \mathcal{C}_{\Sigma,\mathbb{S}}$ and r occurs in Σ. Then, for any $a, b \in \mathcal{O}_\Sigma$,*

- $\Sigma^* \models B(a) \Rightarrow B(a) \in \mathcal{A}^*$.
- $\Sigma^* \models r(a,b) \Rightarrow r(a,b) \in \mathcal{A}^*$.

Proof. Since $\mathcal{A} \subseteq \mathcal{A}^*$, by Lemma 7, we have $\mathcal{K} \models \Sigma^*$. We show that $\mathcal{K} \not\models B(a)$ and $\mathcal{K} \not\models r(a,b)$. Assume that $B(a) \notin \mathcal{A}^*$. Then, $a \notin B^{\mathcal{K}}$ by Lemma 5 and hence $\mathcal{K} \not\models B(a)$. Now, assume that $r(a,b) \notin \mathcal{A}^*$. Then, $(a,b) \notin r^{\mathcal{K}}$ by Lemma 3 and hence $\mathcal{K} \not\models r(a,b)$. □

4 SECRECY-PRESERVING REASONING

Let $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ be an $\mathcal{ELH}$ KB. Also let $\mathbb{S}_{\mathcal{A}} \subseteq \mathcal{A}^* \setminus AX_{\mathcal{A}}$ and $\mathbb{S}_{\mathcal{T}} \subseteq \mathcal{T}^* \setminus AX_{\mathcal{T}}$ be the "secrecy sets". Given Σ, $\mathbb{S}_{\mathcal{A}}$ and $\mathbb{S}_{\mathcal{T}}$, the objective is to answer assertion or GCI queries while preserving secrecy. Our approach is to compute two sets $\mathbb{E}_{\mathcal{A}}$ and $\mathbb{E}_{\mathcal{T}}$, where $\mathbb{S}_{\mathcal{A}} \subseteq \mathbb{E}_{\mathcal{A}} \subseteq \mathcal{A}^* \setminus AX_{\mathcal{A}}$ and $\mathbb{S}_{\mathcal{T}} \subseteq \mathbb{E}_{\mathcal{T}} \subseteq \mathcal{T}^* \setminus AX_{\mathcal{T}}$, called the *secrecy envelopes* for $\mathbb{S}_{\mathcal{A}}$ and $\mathbb{S}_{\mathcal{T}}$ respectively, so that protecting $\mathbb{E}_{\mathcal{A}}$ and $\mathbb{E}_{\mathcal{T}}$, the querying agent cannot logically infer any assertion in $\mathbb{S}_{\mathcal{A}}$ and any GCI in $\mathbb{S}_{\mathcal{T}}$, see (Tao et al., 2010) where the DL language is just $\mathcal{EL}$ and secrecy is restricted to membership assertions. Similarly, (Tao et al., 2014) presents a general framework for secrecy preserving reasoning. The role of OWA in answering the queries is the following: When answering a query with "Unknown", the querying agent should not be able to distinguish between the case that the answer to the query is truly unknown to the KB reasoner and the case that the answer is being protected for reasons of secrecy. We envision a situation in which once the ABox $\mathcal{A}^*$ and TBox $\mathcal{T}^*$ are computed, a reasoner $\mathfrak{R}$ is associated with it. $\mathfrak{R}$ is designed to answer queries as follows: If a query cannot be inferred from Σ, the answer is "Unknown". If it can be inferred and it is not in $\mathbb{E}_{\mathcal{A}} \cup \mathbb{E}_{\mathcal{T}}$, the answer is "Yes"; otherwise, the answer is "Unknown". Note that since the syntax of $\mathcal{ELH}$ does not include negation, an $\mathcal{ELH}$ KB cannot entail a negative query.

We make the following assumptions about the capabilities of the querying agent:

(a) does not have direct access to the KB Σ, but is aware of the underlying vocabulary,
(b) can ask queries in the form of assertions or GCIs, and
(c) cannot ask queries in the form of role inclusions.

We formally define the notion of an envelope in the following.

Definition 3. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ be a $\mathcal{ELH}$ KB, and let $\mathbb{S}_{\mathcal{A}}$ and $\mathbb{S}_{\mathcal{T}}$ be two finite secrecy sets. The secrecy envelopes $\mathbb{E}_{\mathcal{A}}$ and $\mathbb{E}_{\mathcal{T}}$ of $\mathbb{S}_{\mathcal{A}}$ and $\mathbb{S}_{\mathcal{T}}$ respectively, have the following properties:*

- $\mathbb{S}_{\mathcal{A}} \subseteq \mathbb{E}_{\mathcal{A}} \subseteq \mathcal{A}^* \setminus AX_{\mathcal{A}}$,
- $\mathbb{S}_{\mathcal{T}} \subseteq \mathbb{E}_{\mathcal{T}} \subseteq \mathcal{T}^* \setminus AX_{\mathcal{T}}$,

- *for every $\alpha \in \mathbb{E}_{\mathcal{T}}$, $\mathcal{T}^* \setminus \mathbb{E}_{\mathcal{T}} \not\models \alpha$, and*
- *for every $\alpha \in \mathbb{E}_{\mathcal{A}}$, $\mathcal{A}^* \setminus \mathbb{E}_{\mathcal{A}} \not\models \alpha$.*

The intuition for the above definition is that no information in $\mathbb{E}_{\mathcal{A}}$ and $\mathbb{E}_{\mathcal{T}}$ can be inferred from the corresponding sets $\mathcal{A}^* \setminus \mathbb{E}_{\mathcal{A}}$ and $\mathcal{T}^* \setminus \mathbb{E}_{\mathcal{T}}$. To compute envelopes, we use the idea of inverting the rules of Figures 1 and 2 (see (Tao et al., 2010), where this approach was first utilized for membership assertions). Induced by the TBox and ABox expansion rules in Figures 1 and 2, we define the corresponding "inverted" ABox and TBox expansion rules in Figures 3 and 4, respectively. These inverted expansion rules are denoted by prefixing Inv- to the name of the corresponding expansion rules.

$$\begin{array}{rl}
\text{Inv-A}_{\sqcap}^{-}-\text{rule}: & \text{if } \{C(a),D(a)\}\cap\mathbb{E}_{\mathcal{A}}\neq\emptyset \\
& \text{and } C\sqcap D(a)\in\mathcal{A}^*\setminus\mathbb{E}_{\mathcal{A}}, \\
& \text{then } \mathbb{E}_{\mathcal{A}} := \mathbb{E}_{\mathcal{A}}\cup\{C\sqcap D(a)\}; \\
\text{Inv-A}_{\sqcap}^{+}-\text{rule}: & \text{if } C\sqcap D(a)\in\mathbb{E}_{\mathcal{A}},\ C\sqcap D\in\mathcal{C}_{\Sigma,\mathbb{S}} \\
& \text{and } \{C(a),D(a)\}\subseteq\mathcal{A}^*\setminus\mathbb{E}_{\mathcal{A}}, \\
& \text{then } \mathbb{E}_{\mathcal{A}} := \mathbb{E}_{\mathcal{A}}\cup\{C(a)\} \\
& \text{or } \mathbb{E}_{\mathcal{A}} := \mathbb{E}_{\mathcal{A}}\cup\{D(a)\}; \\
\text{Inv-A}_{\exists}^{+}-\text{rule}: & \text{if } \exists r.C(a)\in\mathbb{E}_{\mathcal{A}}, \\
& \{r(a,b),C(b)\}\subseteq\mathcal{A}^*\setminus\mathbb{E}_{\mathcal{A}} \\
& \text{and } \exists r.C\in\mathcal{C}_{\Sigma,\mathbb{S}}, \\
& \text{then } \mathbb{E}_{\mathcal{A}} := \mathbb{E}_{\mathcal{A}}\cup\{r(a,b)\} \\
& \text{or } \mathbb{E}_{\mathcal{A}} := \mathbb{E}_{\mathcal{A}}\cup\{C(b)\}; \\
\text{Inv-A}_{\sqsubseteq}-\text{rule}: & \text{if } D(a)\in\mathbb{E}_{\mathcal{A}},\ C\sqsubseteq D\in\mathcal{T}^*, \\
& \text{and } C(a)\ \in\mathcal{A}^*\setminus\mathbb{E}_{\mathcal{A}}, \\
& \text{then } \mathbb{E}_{\mathcal{A}} := \mathbb{E}_{\mathcal{A}}\cup\{C(a)\}; \\
\text{Inv-A}_{\exists H}^{+}-\text{rule}: & \text{if } \exists s.D(a)\in\mathbb{E}_{\mathcal{A}},\ C\sqsubseteq D\in\mathcal{T}^*, \\
& r\sqsubseteq s\in\mathcal{R}^*,\ \exists s.D\in\mathcal{C}_{\Sigma,\mathbb{S}} \text{ and} \\
& \exists r.C(a)\ \in\mathcal{A}^*\setminus\mathbb{E}_{\mathcal{A}}, \\
& \text{then } \mathbb{E}_{\mathcal{A}} := \mathbb{E}_{\mathcal{A}}\cup\{\exists r.C(a)\}; \\
\text{Inv-A}_{H}-\text{rule}: & \text{if } s(a,b)\in\mathbb{E}_{\mathcal{A}},\ r\sqsubseteq s\in\mathcal{R}^*, \\
& \text{and } r(a,b)\ \in\mathcal{A}^*\setminus\mathbb{E}_{\mathcal{A}}, \\
& \text{then } \mathbb{E}_{\mathcal{A}} := \mathbb{E}_{\mathcal{A}}\cup\{r(a,b)\}.
\end{array}$$

Figure 3: Inverted ABox Tableau expansion rules.

From now on, we assume that $\mathcal{A}^*$, $\mathcal{T}^*$ and $\mathcal{R}^*$ have been computed and readily available for computing the envelopes. The computation of envelopes proceeds in two steps. In the first step, we compute $\mathbb{E}_{\mathcal{A}}$ by initializing it to $\mathbb{S}_{\mathcal{A}}$ and then expanding it using the inverted expansion rules listed in Figure 3 until no further applications are possible. We denote by $\Lambda^{S}_{\mathcal{A}}$ the algorithm which computes the set $\mathbb{E}_{\mathcal{A}}$. Due to non-determinism in applying the rules Inv-$\text{A}_{\sqcap}^{+}$ and Inv-$\text{A}_{\exists}^{+}$, different executions of $\Lambda^{S}_{\mathcal{A}}$ may result in different outputs. Since $\mathcal{A}^*$ is finite, the computation of $\Lambda^{S}_{\mathcal{A}}$ terminates. Let $\mathbb{E}_{\mathcal{A}}$ be an output of $\Lambda^{S}_{\mathcal{A}}$. Since the size of $\mathcal{A}^*$ is polynomial in $|\Sigma| + |\mathcal{C}_{\Sigma,\mathbb{S}}|$, and each application of inverted expansion rule moves some assertions from $\mathcal{A}^*$ into $\mathbb{E}_{\mathcal{A}}$, the size of $\mathbb{E}_{\mathcal{A}}$ is at most the size of $\mathcal{A}^*$. Therefore $\Lambda^{S}_{\mathcal{A}}$ takes polynomial time in $|\,\Sigma\,| + |\,\mathcal{C}_{\Sigma,\mathbb{S}}\,|$ to compute the envelope $\mathbb{E}_{\mathcal{A}}$.

In step two, we compute $\mathbb{E}_{\mathcal{T}}$ independent of $\mathbb{E}_{\mathcal{A}}$ by initializing it to $\mathbb{S}_{\mathcal{T}}$ and then expanding it using the inverted TBox expansion rules listed in Figure 4 until no further applications of rules are possible. We denote by $\Lambda^{S}_{\mathcal{T}}$ the algorithm which computes the set $\mathbb{E}_{\mathcal{T}}$. Similarly to $\Lambda^{S}_{\mathcal{A}}$, due to non-determinism in applying Inv-$\text{T}_{\sqcap}^{+}$ and Inv-T_{H}^{+}-rules, different executions of $\Lambda^{S}_{\mathcal{T}}$ may result in different outputs. Since $\mathcal{T}^*$ is finite, the computation of $\Lambda^{S}_{\mathcal{T}}$ terminates. Let $\mathbb{E}_{\mathcal{T}}$ be an output of $\Lambda^{S}_{\mathcal{T}}$. Since the size of $\mathcal{T}^*$ is polynomial in the size of Σ and $\mathcal{C}_{\Sigma,\mathbb{S}}$, and each application of inverted TBox expansion rule moves some GCIs from $\mathcal{T}^*$ into $\mathbb{E}_{\mathcal{T}}$, the size of $\mathbb{E}_{\mathcal{T}}$ is at most the size of $\mathcal{T}^*$. Therefore $\Lambda^{S}_{\mathcal{T}}$ takes polynomial time in $|\,\Sigma\,| + |\,\mathcal{C}_{\Sigma,\mathbb{S}}\,|$ to compute the envelope $\mathbb{E}_{\mathcal{T}}$.

$$\begin{array}{rl}
\text{Inv-T}_{\sqsubseteq}-\text{rule}: & \text{if } C\sqsubseteq E\in\mathbb{E}_{\mathcal{T}},\ D\sqsubseteq E\in\mathcal{T} \\
& \text{and } C\sqsubseteq D\ \in\mathcal{T}^*\setminus\mathbb{E}_{\mathcal{T}}, \\
& \text{then } \mathbb{E}_{\mathcal{T}} := \mathbb{E}_{\mathcal{T}}\cup\{C\sqsubseteq D\}; \\
\text{Inv-T}_{\sqcap}^{-}-\text{rule}: & \text{if } \{C\sqsubseteq D, C\sqsubseteq E\}\cap\mathbb{E}_{\mathcal{T}}\neq\emptyset \\
& \text{and } C\sqsubseteq D\sqcap E\in\mathcal{T}^*\setminus\mathbb{E}_{\mathcal{T}}, \\
& \text{then } \mathbb{E}_{\mathcal{T}} := \mathbb{E}_{\mathcal{T}}\cup\{C\sqsubseteq D\sqcap E\}; \\
\text{Inv-T}_{\sqcap}^{+}-\text{rule}: & \text{if } C\sqsubseteq D\sqcap E\in\mathbb{E}_{\mathcal{T}},\ D\sqcap E\in\mathcal{C}_{\Sigma,\mathbb{S}} \\
& \text{and } \{C\sqsubseteq D, C\sqsubseteq E\}\subseteq\mathcal{T}^*\setminus\mathbb{E}_{\mathcal{T}}, \\
& \text{then } \mathbb{E}_{\mathcal{T}} := \mathbb{E}_{\mathcal{T}}\cup\{C\sqsubseteq D\} \\
& \text{or } \mathbb{E}_{\mathcal{T}} := \mathbb{E}_{\mathcal{T}}\cup\{C\sqsubseteq E\}; \\
\text{Inv-T}_{H}^{+}-\text{rule}: & \text{if } C\sqsubseteq\exists s.E\in\mathbb{E}_{\mathcal{T}},\ r\sqsubseteq s\in\mathcal{R}^*, \\
& \exists s.E\in\mathcal{C}_{\Sigma,\mathbb{S}} \text{ and} \\
& \{C\sqsubseteq\exists r.D, D\sqsubseteq E\}\subseteq\mathcal{T}^*\setminus\mathbb{E}_{\mathcal{T}}, \\
& \text{then } \mathbb{E}_{\mathcal{T}} := \mathbb{E}_{\mathcal{T}}\cup\{C\sqsubseteq\exists r.D\} \\
& \text{or } \mathbb{E}_{\mathcal{T}} := \mathbb{E}_{\mathcal{T}}\cup\{D\sqsubseteq E\}.
\end{array}$$

Figure 4: Inverted TBox Tableau expansion rules.

Example 3. *(Example 2 cont.) Recall that $\mathcal{A}^*$ and $\mathcal{T}^*$ are the completed ABox and TBox respectively. Let $\mathbb{S}_{\mathcal{A}} = \{D\sqcap E(a)\}$ and $\mathbb{S}_{\mathcal{T}} = \{C\sqsubseteq D\sqcap E\}$ be the secrecy sets. Then, by using rules in Figure 3, we get the envelope for $\mathbb{S}_{\mathcal{A}}$,*

$$\mathbb{E}_{\mathcal{A}} = \mathbb{S}_{\mathcal{A}}\cup\{D(a)\}.$$

Similarly, using the rules in Figure 4, we get the envelope for $\mathbb{S}_{\mathcal{T}}$,

$$\mathbb{E}_{\mathcal{T}} = \mathbb{S}_{\mathcal{T}}\cup\{C\sqsubseteq D\}.\quad\square$$

Before proving the main results on envelopes, we prove the following auxiliary lemmas. First, we show that no assertions in $\mathbb{E}_\mathcal{A}$ is “logically reachable” from any assertion in $\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$.

Lemma 8. *Let $\mathcal{A}^*$ be a completed ABox obtained from $\mathcal{A}$ by applying the tableau expansion rules in Figure 2. Also, let $\mathbb{E}_\mathcal{A}$ be a set of assertions which is completed by applying the tableau expansion rules in Figure 3 starting with the secrecy set $\mathbb{S}_\mathcal{A}$. Then, the ABox $\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$ is completed.*

Proof. We have to show that no rule in Figure 2 is applicable to $\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$. The proof is by contradiction according to cases: assuming that a rule in Figure 2 is applicable and showing that a some inverse rule is applicable.

- If A$_\sqcap^-$-rule is applicable, then there is an assertion $C \sqcap D(a) \in \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$ such that $C(a) \notin \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$ or $D(a) \notin \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$. Since $\mathcal{A}^*$ is completed, $\{C(a), D(a)\} \subseteq \mathcal{A}^*$. Hence, $\{C(a), D(a)\} \cap \mathbb{E}_\mathcal{A} \neq \emptyset$. This makes the Inv-A$_\sqcap^-$-rule applicable.
- If A$_\sqcap^+$-rule is applicable, then there are assertions $C(a), D(a) \in \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$ such that $C \sqcap D \in \mathcal{C}_{\Sigma,\mathbb{S}}$ and $C \sqcap D(a) \notin \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$. Since $\mathcal{A}^*$ is completed, $C \sqcap D(a) \in \mathcal{A}^*$. Hence, $C \sqcap D(a) \in \mathbb{E}_\mathcal{A}$. This makes the Inv-A$_\sqcap^+$-rule applicable.
- If A$_\exists^+$-rule is applicable, then there are assertions $r(a,b), C(b) \in \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$ such that $\exists r.C \in \mathcal{C}_{\Sigma,\mathbb{S}}$ and $\exists r.C(a) \notin \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$. Since $\mathcal{A}^*$ is completed, $\exists r.C(a) \in \mathcal{A}^*$. Hence, $\exists r.C(a) \in \mathbb{E}_\mathcal{A}$. This makes the Inv-A$_\exists^+$-rule applicable.
- If A$_\sqsubseteq$-rule is applicable, then there is an assertion $C(a) \in \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$ and a GCI $C \sqsubseteq D \in \mathcal{T}^*$ such that $D(a) \notin \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$. Since $\mathcal{A}^*$ is completed, $D(a) \in \mathcal{A}^*$. Hence, $D(a) \in \mathbb{E}_\mathcal{A}$. This makes the Inv-A$_\sqsubseteq$-rule applicable.
- If A$_{\exists H}^+$-rule is applicable, then there is an assertion $\exists r.C(a) \in \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$, a GCI $C \sqsubseteq D \in \mathcal{T}^*$, a role inclusion $r \sqsubseteq s \in \mathcal{R}^*$ such that $\exists s.D \in \mathcal{C}_{\Sigma,\mathbb{S}}$ and $\exists s.D(a) \notin \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$. Since $\mathcal{A}^*$ is completed, $\exists s.D(a) \in \mathcal{A}^*$. Hence, $\exists s.D(a) \in \mathbb{E}_\mathcal{A}$. This makes the Inv-A$_{\exists H}^+$-rule applicable.
- If A$_H$-rule is applicable, then there is an assertion $r(a,b) \in \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$ and a role inclusion $r \sqsubseteq s \in \mathcal{R}^*$ such that $s(a,b) \notin \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$. Since $\mathcal{A}^*$ is completed, $s(a,b) \in \mathcal{A}^*$. Hence, $s(a,b) \in \mathbb{E}_\mathcal{A}$. This makes the Inv-A$_H$-rule applicable.

□

The next lemma is an analog of Lemma 8 for $\mathcal{T}^*$. Its proof is similar.

Lemma 9. *Let $\mathcal{T}^*$ be a completed TBox obtained from Σ and $\mathcal{C}_{\Sigma,\mathbb{S}}$ by applying the tableau expansion rules in Figure 1. Also, let $\mathbb{E}_\mathcal{T}$ be a set of GCIs which is completed by using tableau expansion rules in Figure 4 starting with the secrecy set $\mathbb{S}_\mathcal{T}$. Then, the TBox $\mathcal{T}^* \setminus \mathbb{E}_\mathcal{T}$ is completed.*

We now show that the completed sets $\mathbb{E}_\mathcal{A}$ and $\mathbb{E}_\mathcal{T}$ are in fact envelopes.

Theorem 3. *$\mathbb{E}_\mathcal{A}$ and $\mathbb{E}_\mathcal{T}$ are envelopes for $\mathbb{S}_\mathcal{A}$ and $\mathbb{S}_\mathcal{T}$ respectively .*

Proof. We must show that the sets $\mathbb{E}_\mathcal{A}$ and $\mathbb{E}_\mathcal{T}$ satisfy the four properties of Definition 3. Properties 1 and 2 are obvious. To prove property 3, suppose $\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A} \models \alpha$, for some $\alpha \in \mathbb{E}_\mathcal{A}$. This means, by Theorem 2, that $\alpha \in (\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A})^*$ and since, by Lemma 8, $\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$ is completed, $(\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A})^* = \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$, whence $\alpha \in \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$. This is a contradiction. Proof of property 4 is similar, using Theorem 1 and Lemma 9 instead of Theorem 2 and Lemma 8, respectively. □

To answer queries as informatively as possible without revealing the secret information, we should aim to make the size of the envelope $\mathbb{E}$ as small as possible. From now on, we focus on computing an envelope $\mathbb{E}$ with the property that removing any member in $\mathbb{E}$ could reveal some of the secrets. We call such an envelope *tight*.

Definition 4. *An envelope $\mathbb{E}$ is said to be tight if for every $\alpha \in \mathbb{E}$, $\mathbb{E} \setminus \{\alpha\}$ is not an envelope.*

We now show by an example, that the envelopes computed by using the rules in Figures 3 and 4 are not necessarily tight.

Example 4. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ be a $\mathcal{ELH}$ KB, where $\mathcal{A} = \{C(a), r(b,a)\}$, $\mathcal{T} = \{A \sqsubseteq B, C \sqsubseteq D \sqcap E, C \sqsubseteq D \sqcap F\}$ and $\mathcal{R} = \emptyset$. Also let $\mathbb{S}_\mathcal{A} = \{D \sqcap E(a), D \sqcap F(a)\}$ and $\mathbb{S}_\mathcal{T} = \{C \sqsubseteq D \sqcap E, C \sqsubseteq D \sqcap F\}$ be the secrecy sets.*

Since $\Lambda^S_\mathcal{A}$ is non-deterministic, we may get different envelopes as an output. Some of the envelopes are

1 $\mathbb{E}_\mathcal{A} = \mathbb{S}_\mathcal{A} \cup \{D(a), F(a)\}$ *– not tight,*
2 $\mathbb{E}_\mathcal{A} = \mathbb{S}_\mathcal{A} \cup \{E(a), F(a)\}$ *– tight and*
3 $\mathbb{E}_\mathcal{A} = \mathbb{S}_\mathcal{A} \cup \{D(a)\}$ *– minimum and tight.*

Since $\Lambda^S_\mathcal{T}$ is non-deterministic, we may get different envelopes as an output depending on the choice made in the application of Inv-T$^+_\sqcap$-rule when computing the envelopes. The envelopes are

1 $\mathbb{E}_\mathcal{T} = \mathbb{S}_\mathcal{T} \cup \{C \sqsubseteq D, C \sqsubseteq F\}$ *– not tight,*
2 $\mathbb{E}_\mathcal{T} = \mathbb{S}_\mathcal{T} \cup \{C \sqsubseteq E, C \sqsubseteq F\}$ *– tight and*
3 $\mathbb{E}_\mathcal{T} = \mathbb{S}_\mathcal{T} \cup \{C \sqsubseteq D\}$ *– minimum and tight.* □

We briefly describe a naive procedure to compute a tight envelope. Given a precomputed $\mathcal{A}^*$ and a secrecy set $\mathbb{S}_\mathcal{A}$, we can compute an envelope $\mathbb{E}_\mathcal{A}$

of $\mathbb{S}_\mathcal{A}$ in polynomial time as explained in the beginning of this section. An assertion $\alpha \in \mathbb{E}_\mathcal{A} \setminus \mathbb{S}_\mathcal{A}$ is said to be *redundant* if $\mathbb{E}_\mathcal{A} \setminus \{\alpha\}$ is an envelope, i.e., $((\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}) \cup \{\alpha\})^* \cap (\mathbb{E}_\mathcal{A} \setminus \{\alpha\}) = \emptyset$. To compute a tight envelope, for each $\beta \in \mathbb{E}_\mathcal{A} \setminus \mathbb{S}_\mathcal{A}$ check whether β is redundant in which case it is moved from $\mathbb{E}_\mathcal{A}$ to $\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$. Otherwise, β remains in $\mathbb{E}_\mathcal{A}$. It is easy to see that checking whether an element in the set $\mathbb{E}_\mathcal{A} \setminus \mathbb{S}_\mathcal{A}$ is redundant or not, can be done in polynomial time. This redundancy check should be done for each $\beta \in \mathbb{E}_\mathcal{A} \setminus \mathbb{S}_\mathcal{A}$. Hence given $\mathcal{A}^*$, $\mathbb{S}_\mathcal{A}$ and $\mathbb{E}_\mathcal{A}$, a tight envelope can be computed in polynomial time. The same procedure may be used to compute a tight envelope for the secrecy set $\mathbb{S}_\mathcal{T}$.

5 QUERY ANSWERING

The recursive procedures given in Figures 5 and 6 take an input q (as a query) and output “Yes” or “Unknown”.

```
EvalA($q$)
 1: case $q \in \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$
 2:     return “Yes”
 3: case $q = C \sqcap D(a)$
 4:     if EvalA($C(a)$) = “Yes” and
        EvalA($D(a)$) = “Yes” then
 5:         return “Yes”
 6:     else
 7:         return “Unknown”
 8: case $q = \exists r.C(a)$
 9:     if for some $d \in \mathcal{O}_\Sigma$ [ $r(a,d) \in \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$
        and EvalA($C(d)$) =“Yes”]  then
10:         return “Yes”
11:     else
12:         if for some $E \in \mathcal{C}_{\Sigma,\mathbb{S}}$ [$E \sqsubseteq C \in \mathcal{T}^*$
            and EvalA($\exists r.E(a)$) = “Yes” ] then
13:             return “Yes”
14:         else
15:             if  for some $s \in R_\mathcal{R}$
                [$s \sqsubseteq r \in \mathcal{R}^*$ and
                EvalA($\exists s.C(a)$) = “Yes” ] then
16:                 return “Yes”
17:             else
18:                 return “Unknown”
```

Figure 5: Query answering algorithm for assertional queries.

In Section 4, we have described briefly how the reasoner $\mathfrak{R}$ responds to queries. In this section we provide a few more details. Here we assume that $\mathcal{A}^*$, $\mathbb{E}_\mathcal{A}$, $\mathcal{T}^*$, $\mathbb{E}_\mathcal{T}$ and $\mathcal{R}^*$ have all been precomputed and are considered to be globally accessible. Define the set $R_\mathcal{R} = \{r \mid r \text{ is a role name that occurs in } \mathcal{R}\}$. The recursive procedures for answering the assertional queries and the GCI queries are given in Figure 5 and Figure 6 respectively. In Lines 1 and 2 of Figure 5, we check the membership of q in $\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$ and answer “Yes” if $q \in \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$. From line 3 onwards we consider several cases in which we break the query q into subqueries based on the constructors defined in the language $\mathcal{ELH}$ and apply the procedure recursively. The following theorem proves the correctness of the algorithm.

Theorem 4. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ be an $\mathcal{ELH}$ KB. Let $\mathcal{A}^*$ be an completed ABox, $\mathbb{E}_\mathcal{A}$ an envelope of the secrecy set $\mathbb{S}_\mathcal{A}$ and q an assertional query. Then,*

- *Soundness: EvalA(q) outputs “Yes” $\Rightarrow$ $\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A} \models q$*
- *Completeness: EvalA(q) outputs “Unknown” $\Rightarrow$ $\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A} \not\models q$*

Proof. We omit the proof of soundness.

We prove the completeness part using a contrapositive argument. Assume that $\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A} \models q$. We have to show that EvalA(q) = “Yes”. Let $\mathcal{K}$ be the canonical interpretation as defined in section 3.2. By Lemma 7, $\mathcal{K}$ satisfies $\mathcal{A}^*$, $\mathcal{T}^*$ and $\mathcal{R}^*$ and hence $\mathcal{K}$ satisfies $\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$ and q. We argue that: if $\mathcal{K} \models q$ then EvalA(q) = “Yes”, by induction on the structure of q. There are two cases. If $q \in \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$, then the claim follows immediately. Next, consider the case $q \notin \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$. There are several cases:

- $q = C \sqcap D(a)$. To answer this query the algorithm computes EvalA($C(a)$) and EvalA($D(a)$). Now, the assumption $\mathcal{K} \models C \sqcap D(a)$ implies $\mathcal{K} \models C(a)$ and $\mathcal{K} \models D(a)$ which, by inductive hypothesis, implies that EvalA($C(a)$) = EvalA($D(a)$) = “Yes”. Hence, by Lines 4 and 5 in Figure 5, EvalA($C \sqcap D(a)$)=“Yes”.
- $q = \exists r.C(a)$. By the assumption, $\mathcal{K} \models \exists r.C(a)$. This implies, for some $b \in \Delta$ [$(a,b) \in r^\mathcal{K}$ and $b \in C^\mathcal{K}$]. There are two subcases:
  - r is minimal with respect to (a,b). Again there are two subcases:
    - $b \in \mathcal{O}_\Sigma$. Then, $\mathcal{K} \models r(a,b)$ and $\mathcal{K} \models C(b)$. By the first case $r(a,b) \in \mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$ and by inductive hypothesis EvalA($C(b)$) = “Yes”. Hence, by Lines 9 and 10 in Figure 5, EvalA ($\exists r.C(a)$)=“Yes”.
    - $b = w_D \in \mathcal{W}$ for some $D \in \mathcal{C}_{\Sigma,\mathbb{S}}$. Then, $\mathcal{K} \models \exists r.D(a)$ and by part (b) of Lemma 4, $D \sqsubseteq C \in \mathcal{T}^*$. By inductive hypothesis EvalA($\exists r.D(a)$) = “Yes”. Hence, by Lines 12 and 13 in Figure 5, EvalA($\exists r.C(a)$) =“Yes”.
  - r is not minimal with respect to (a,b). Since RBox $\mathcal{R}$ is acyclic, there exists a chain $s \sqsubseteq v_1 \sqsubseteq$

$v_2 ...... \sqsubseteq v_k \sqsubseteq u$ in $\mathcal{R}$ such that s is minimal with respect to (a,b). Since $\mathcal{R}^*$ is the transitive closure of $\mathcal{R}$, $s \sqsubseteq r \in \mathcal{R}^*$. Again there are two cases:

- $b \in \mathcal{O}_\Sigma$. Then, by Definition 2 and the definition of $\mathcal{K}$, $\mathcal{K} \models s(a,b)$. Also, $\mathcal{K} \models s \sqsubseteq r$ and $\mathcal{K} \models C(b)$. By the first subcase of the previous case EvalA($\exists s.C(a)$) = "Yes". Hence, by Lines 15 and 16 in Figure 5, EvalA($\exists r.C(a)$)="Yes".
- $b = w_D \in \mathcal{W}$ for some $D \in \mathcal{C}_{\Sigma,\mathbb{S}}$. Then, by Definition 2 and the definition of $\mathcal{K}$, $\mathcal{K} \models \exists s.D(a)$. Also, $\mathcal{K} \models s \sqsubseteq r$ and by part (b) of Lemma 4, $D \sqsubseteq C \in \mathcal{T}^*$. By the second subcase of the previous case EvalA($\exists s.C(a)$) = "Yes". Hence, by Lines 15 and 16 in Figure 5, EvalA($\exists r.C(a)$) ="Yes".

□

Since the algorithm given in Figure 5 runs in polynomial time in the size of $\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$ and q, the assertional query answering can be done in polynomial time as a function of $| \mathcal{A}^* | + | q |$.

```
EvalT($q$)
 1: case $q \in \mathcal{T}^* \setminus \mathbb{E}_\mathcal{T}$
 2:     return "Yes"
 3: case $q = C \sqsubseteq D \sqcap E$
 4:     if EvalT($C \sqsubseteq D$) ="Yes" and EvalT($C \sqsubseteq E$) ="Yes" then
 5:         return "Yes"
 6:     else
 7:         return "Unknown"
 8: case $q = C \sqsubseteq \exists r.D$
 9:     if for some $E \in \mathcal{C}_{\Sigma,\mathbb{S}}$ [$E \sqsubseteq D \in \mathcal{T}^*$ and EvalT($C \sqsubseteq \exists r.E$) ="Yes"] then
10:         return "Yes"
11:     else
12:         if for some $s \in R_\mathcal{R}$ [$s \sqsubseteq r \in \mathcal{R}^*$ and EvalT($C \sqsubseteq \exists s.D$) ="Yes"] then
13:             return "Yes"
14:         else
15:             return "Unknown"
```

Figure 6: Query answering algorithm for GCI queries.

Next, suppose that the querying agent poses a GCI query q. In response, the reasoner $\mathfrak{R}$ invokes the query answering algorithm EvalT(q) given in Figure 6 and returns the answer as output. We prove in the following the correctness of the recursive algorithm given in Figure 6.

Example 5. *(Example 3 cont.) Recall that $\mathcal{A}^*$ and $\mathcal{T}^*$ are the completed ABox and TBox respectively. Also, recall that $\mathbb{E}_\mathcal{A} = \mathbb{S}_\mathcal{A} \cup \{D(a)\}$ and $\mathbb{E}_\mathcal{T} = \mathbb{S}_\mathcal{T} \cup \{C \sqsubseteq D\}$ are the the envelopes for $\mathbb{S}_\mathcal{A}$ and $\mathbb{S}_\mathcal{T}$ respectively.*

Suppose that the querying agent asks the assertional queries $C \sqcap E(a)$, $\exists r.C(b)$, $\exists r.E(b)$ and $D(a)$. Using the algorithm in Figure 5, we get the following answers:

q	*EvalA(q)*	*Remarks*
$C \sqcap E(a)$	*Yes*	*by Lines 4, 5*
$\exists r.E(b)$	*Yes*	*by Lines 12, 13*
$D(a)$	*Unknown*	*by Line 18*

Next, suppose that the querying agent asks the GCI queries $C \sqsubseteq C \sqcap E$, $\exists r.C \sqsubseteq \exists r.E$ and $C \sqsubseteq D$. Using the algorithm in Figure 6, we get the following answers:

q	*EvalT(q)*	*Remarks*
$C \sqsubseteq C \sqcap E$	*Yes*	*by Lines 4, 5*
$\exists r.C \sqsubseteq \exists r.E$	*Yes*	*by Lines 9, 10*
$C \sqsubseteq D$	*Unknown*	*by Line 15* □

Theorem 5. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ be an $\mathcal{ELH}$ KB. Let $\mathcal{T}^*$ be an completed TBox, $\mathbb{E}_\mathcal{T}$ an envelope of the secrecy set $\mathbb{S}_\mathcal{T}$ and q a GCI query. Then,*

- *Soundness: EvalT(q) outputs "Yes" $\Rightarrow$ $\mathcal{T}^* \setminus \mathbb{E}_\mathcal{T} \models q$*
- *Completeness: EvalT(q) outputs "Unknown" $\Rightarrow$ $\mathcal{T}^* \setminus \mathbb{E}_\mathcal{T} \not\models q$*

Proof. We prove the completeness part using a contrapositive argument. Assume that $\mathcal{T}^* \setminus \mathbb{E}_\mathcal{T} \models q$. We have to show that EvalT(q) ="Yes". Let $\mathcal{J}$ be the canonical interpretation as defined in section 3.1. By Lemma 2, $\mathcal{J}$ satisfies $\mathcal{T}^*$ and $\mathcal{R}^*$. Hence $\mathcal{J}$ satisfies $\mathcal{T}^* \setminus \mathbb{E}_\mathcal{T}$ and q. We argue by induction on the structure of q that, if $\mathcal{J} \models q$ then EvalT(q) = "Yes". The basic case is. $q \in \mathcal{T}^* \setminus \mathbb{E}_\mathcal{T}$. Then, by Lines 1 and 2 in Figure 6, the claim is obvious. Next, consider the case $q \notin \mathcal{T}^* \setminus \mathbb{E}_\mathcal{T}$. There are several cases:

- $q = C \sqsubseteq D \sqcap E$. The algorithm in Figure 6 computes EvalT($C \sqsubseteq D$) and EvalT($C \sqsubseteq E$). Now, the assumption $\mathcal{J} \models C \sqsubseteq D \sqcap E$ implies $\mathcal{J} \models C \sqsubseteq D$ and $\mathcal{J} \models C \sqsubseteq E$ which, by inductive hypothesis, implies that EvalT($C \sqsubseteq D$) = EvalT($C \sqsubseteq E$) = "Yes". Hence, by Lines 4 and 5 in Figure 6, EvalT($C \sqsubseteq D \sqcap E$) = "Yes".
- $q = C \sqsubseteq \exists r.D$. By the assumption, $\mathcal{J} \models C \sqsubseteq \exists r.D$. This implies, $C, D \in \mathcal{C}_{\Sigma,\mathbb{S}}$ and $\exists r.D \notin \mathcal{C}_{\Sigma,\mathbb{S}}$.
  - $\mathcal{J} \models C \sqsubseteq \exists r.E_1, E_1 \sqsubseteq E_2, ...E_{k-1} \sqsubseteq E_k, E_k \sqsubseteq D$ where $\exists r.E_1$, $E_2, ... \ E_k \in \mathcal{C}_{\Sigma,\mathbb{S}}$ and $C \sqsubseteq \exists r.E_1, E_1 \sqsubseteq E_2, ...E_{k-1} \sqsubseteq E_k, E_k \sqsubseteq D \in \mathcal{T}^* \setminus \mathbb{E}_\mathcal{T}$. Since, by Lemma 9, $\mathcal{T}^* \setminus \mathbb{E}_\mathcal{T}$ is completed, $E_1 \sqsubseteq D \in \mathcal{T}^* \setminus \mathbb{E}_\mathcal{T}$. Also, by the basic step, EvalT($C \sqsubseteq \exists r.E_1$) = "Yes". Hence, by Lines 9 and 10, EvalT($C \sqsubseteq \exists r.D$) = "Yes".
  - $\mathcal{J} \models C \sqsubseteq \exists s.D$, $s \sqsubseteq v_1, v_1 \sqsubseteq v_2, ...v_k \sqsubseteq r$ where $\exists s.D \in \mathcal{C}_{\Sigma,\mathbb{S}}$, $s, v_1, v_2, ...v_k \in R_\mathcal{R}$, $C \sqsubseteq \exists s.D \in \mathcal{T}^*$ and $s \sqsubseteq v_1, v_1 \sqsubseteq v_2, ...v_k \sqsubseteq r \in \mathcal{R}^*$. Then, $s \sqsubseteq r \in$

$\mathcal{R}^*$ and by the basic step, EvalT($C \sqsubseteq \exists s.D$) = "Yes". Hence, by Lines 15 and 16, EvalT($C \sqsubseteq \exists r.D$) = "Yes".

□

Since the algorithm runs in polynomial time in the size of $\mathcal{T}^* \setminus \mathbb{E}_\mathcal{T}$ and q, the GCI query answering can be done in polynomial time as a function of $|\,\mathcal{T}^*\,| + |\,q\,|$.

6 SUMMARY

The main contribution of this paper is that we allow secrets as well as queries to be of two types: (a) local type, assertions about specific individuals (e.g., $C(a)$ or $r(a,b)$), as well as (b) global type, GCIs (e.g., $C \sqsubseteq D$) which specify hierarchical inclusion relationships between concepts. Another contribution is in the way that we compute the consequences and preserve secrecy while answering queries. We break the process into two parts, first one precomputes all the consequences for concepts and individuals that occur in the given KB. For this we use four separate (but related) tableau procedures. As for the actual query answering, we parse the query all the way to constituents that occur in the previously precomputed set of consequences. Then, the queries are answered based on the membership of the constituents of the query in $\mathcal{A}^* \setminus \mathbb{E}_\mathcal{A}$ and $\mathcal{T}^* \setminus \mathbb{E}_\mathcal{T}$. All the algorithms are efficient and can be implemented in polynomial time. As for future work, we would like to study secrecy-preserving reasoning framework in modalized $\mathcal{ELH}$ description logic and possibly in probabilistic description logic Prob-$\mathcal{ELH}^{>0,=1}$.

REFERENCES

Bao, J., Slutzki, G., and Honavar, V. (2007). Privacy-preserving reasoning on the semanticweb. In *Web Intelligence, IEEE/WIC/ACM Conference,791–797*.

Bienvenu, M., Ortiz, M., Šimkus, M., and Xiao, G. (2013). Tractable queries for lightweight description logics. In *Proceedings of the Twenty-Third UJCAI,768–774*.

Biskup, J. and Tadros, C. (2012). Revising belief without revealing secrets. In *Foundations of Information and Knowledge Systems*, pages 51–70. Springer.

Biskup, J. and Weibert, T. (2008). Keeping secrets in incomplete databases. *International Journal of Information Security*, 7(3):199–217.

Delaitre, V. and Kazakov, Y. (2009). Classifying $\mathcal{ELH}$ ontologies in sql databases. In *OWLED*.

Kagal, L., Finin, T., and Joshi, A. (2003). A policy based approach to security for the semantic web. In *International Semantic Web Conference*, volume 2870, pages 402–418. Springer.

Kazakov, Y., Krötzsch, M., and Simančík, F. (2014). The incredible elk. *JAR*, 53(1):1–61.

Lutz, C., Toman, D., and Wolter, F. (2008). Conjunctive query answering in $\mathcal{EL}$ using a database system.

Tao, J., Slutzki, G., and Honavar, V. (2010). Secrecy-preserving query answering for instance checking in $\mathcal{EL}$. In *Web Reasoning and Rule Systems, 195–203*.

Tao, J., Slutzki, G., and Honavar, V. (2014). A conceptual framework for secrecy-preserving reasoning in knowledge bases. *TOCL*, 16(1):3.