Secrecy-preserving Reasoning in Acyclic *DL-Lite*$_{\mathcal{R}}$ Knowledge Bases in the Presence of BCQs

Gopalakrishnan Krishnasamy-Sivaprakasam[1] and Giora Slutzki[2]
[1]*Math & Computer Science Department, Central State University, Wilberforce, Ohio, U.S.A.*
[2]*Department of Computer Science, Iowa State University, Ames, Iowa, U.S.A.*

Keywords: Description Logic, Theory of Database Privacy and Security, Knowledge Representation and Reasoning.

Abstract: In this paper we study Secrecy-Preserving Query Answering under Open World Assumption (OWA) for *DL-Lite*$_{\mathcal{R}}$ Knowledge Bases (KBs) with acyclic TBox. Using a tableau algorithm, we construct $\mathcal{A}^*$, an inferential closure of the given ABox $\mathcal{A}$, which includes both positive as well as negative assertions. We use a notational variant of Kleene 3-valued semantics, which we call OW-semantics as it fits nicely with OWA. This allows us to answer queries, including Boolean Conjunctive Queries (BCQs) with "Yes", "No" or "Unknown", as opposed to the just answering "Yes" or "No" as in Ontology Based Data Access (OBDA) framework, thus improving the informativeness of the query-answering procedure. Being able to answer "Unknown" plays a key role in protecting secrecy under OWA. One of the main contributions of this paper is a study of answering BCQs without compromising secrecy. Using the idea of secrecy envelopes, previously introduced by one of the authors, we give a precise characterization of when BCQs should be answered "Yes", "No" or "Unknown". We prove the correctness of the secrecy-preserving query-answering algorithm.

1 INTRODUCTION

Preserving secrecy in a database setting is a problem of paramount importance and it has been studied for a long time, see (Biskup and Weibert, 2008; Biskup et al., 2010; Denning and Denning, 1979; Sicherman et al., 1983). With the advent of the semantic web and its increasingly pervasive usage, there is a lot of interest in studying this problem in knowledge base (KB) setting, see (Bao et al., 2007; Cuenca Grau et al., 2013; Tao et al., 2010; Tao et al., 2015; Stouppa and Studer, 2009; Sivaprakasam, 2016). The concern here is that in view of the fundamental assumption that KBs possess incomplete knowledge, despite our best efforts, a situation could arise in which logical reasoning (used to produce implicit knowledge from explicit one stored in the KB) may possibly lead to disclosure of secret information, see (Cuenca Grau et al., 2013). Some approaches dealing with "information protection" are based on access control mechanisms (Bell and LaPadula, 1973), defining appropriate policy languages to represent obligation, provision and delegation policies (Kagal et al., 2003), and logic based methods applied to protect secrets of one agent's knowledge from the other agents in a multiagent system (Halpern and O'Neill, 2008). One approach to secrecy in incomplete database was presented by Biskup et al. in (Biskup and Weibert, 2008; Biskup et al., 2010; Biskup and Tadros, 2012) in the form of controlled query evaluation (CQE). The idea behind CQE is that rather than providing strict access control to data, the CQE approach enforces secrecy by checking (at run time) whether from a truthful answer to a query a user can deduce secret information. In this case the answer is distorted by either simply refusing to answer or by outright lying. For a study of confidentiality in a setting that is an adaptation of CQE framework to ontologies over OWL 2 RL profile of OWL 2, see (Cuenca Grau et al., 2013).

In response to concerns raised in (Weitzner et al., 2008), we have developed a secrecy framework that attempts to satisfy the following, competing, properties: (a) it protects secret information, and (b) queries are answered as informatively as possible (subject to satisfying property (a)), see (Bao et al., 2007; Tao et al., 2010). This approach is based on Open World Assumption (OWA) and (so far) it has been restricted to instance-checking queries. More specifically, in (Bao et al., 2007) the main idea (which was restricted to hierarchical KBs) was to utilize the secret information within the reasoning process, but then answering "Unknown" whenever the answer is truly unknown or

Krishnasamy-Sivaprakasam, G. and Slutzki, G.
Secrecy-preserving Reasoning in Acyclic DL-LiteR Knowledge Bases in the Presence of BCQs.
DOI: 10.5220/0010335403600372
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, pages 360-372
ISBN: 978-989-758-484-8

in case the true answer could compromise confidentiality. The authors defined and used the notion of an *envelope* to hide secret information against logical inference. Further, in (Tao et al., 2015), the authors introduced a more elaborate conceptual framework for secrecy-preserving query answering (SPQA) problem under OWA with multiple querying agents. This framework was restricted to instance-checking queries and illustrated on very simple description logic languages.

The World Wide Web Consortium (W3C) has proposed OWL 2 profiles which have limited modeling features, but provide substantial improvements in scalability as well as a significant reduction in the complexity of various reasoning tasks. Based on this proposal, there has been a lot of work done on developing languages tailored to specific applications, in particular those that involve massive amount of data, i.e., large ABoxes. In addition, a lot of work has dealt with answering conjunctive queries over these data sets, see (Ortiz and Simkus, 2012). The goal is to provide just enough expressive power to deal with those applications, while keeping low complexity of reasoning, see (Calvanese et al., 2007; Krotzsch, 2012). *DL-Lite* family is one such family of languages designed with an eye towards precisely these kinds of applications, see (Artale et al., 2009; Calvanese et al., 2007; Ortiz and Simkus, 2012).

One of the contributions in this paper is answering Boolean Conjunctive Assertions/Queries (BCQs) without revealing secrets, where the secrecy set contains both assertions and BCQs. As explained below, having BCQs in the secrecy set can be avoided at the expense of considerable "manual labor" of augumenting the secrecy set with all the instances of the given BCQs. For instance, when we want to protect the existence of individuals satisfying certain properties e.g. $A(x)$ and $P(x,y)$, is suffices to add the BCQ $\exists x,y\ [A(x) \wedge P(x,y)]$ into the secrecy set. Otherwise we would have to "manually" add all the assertions of the form $A(a)$ and $P(a,b)$, for all individuals a, b occurring in the KB; see section 5.1. We note that the situation is different with respect to query answering. Here, allowing BCQs indeed adds extra power and cannot be replaced with any number of membership queries. Observe that in this work we pursue (secrecy-preserving) query answering with the answers being "Yes", "No" or "Unknown". For this reason we are interested in BCQs rather than more general CQs. Moreover, to the best of our knowledge, this work presents the first study of secrecy-preserving reasoning which allows BCQ queries.

In this paper we continue the work begun in (Tao et al., 2010; Krishnasamy Sivaprakasam and Slutzki, 2016). The framework introduced in (Tao et al., 2015), which we use here as well, was illustrated on very simple examples: the Propositional Horn Logic and the Description Logic $\mathcal{AL}$. As DL languages become more involved (expressive), the corresponding SPQA problems become more challenging. Here we consider SPQA problem under OWA for *DL-Lite*$_{\mathcal{R}}$ *acyclic* KBs[1]. Given a *DL-Lite*$_{\mathcal{R}}$ KB (consisting of an ABox $\mathcal{A}$ and an acyclic TBox $\mathcal{T}$) and a secrecy set $\mathbb{S}$, the querying agent is allowed to ask queries of both kinds. Moreover, we allow the ABox of the KB to contain both positive and negative assertions, see (Artale et al., 2009) for a survey of *DL-Lite* family of logics. By OWA, the answer to a query against a KB can be "Yes", "No" or "Unknown". As the first step in constructing our SPQA system, we use a tableau algorithm to compute a finite set $\mathcal{A}^*$ which consists of the consequences of the KB (with respect to the TBox), both positive and negative. To prove the completeness of this algorithm, we use the 3-valued OW-semantics as introduced in (Tao et al., 2015), see also Section 2.2. Next, starting from the secrecy set $\mathbb{S}$ we compute a finite set of assertions, viz., the *envelope* $\mathbb{E} \subseteq \mathcal{A}^*$ of the secrecy set $\mathbb{S}$, whose goal is to provide a "logical shield" against reasoning launched from $\mathcal{A}^* \setminus \mathbb{E}$ (outside the envelope) and whose aim is to "infiltrate" the secrecy set $\mathbb{S}$ (i.e., to compromise some assertions in $\mathbb{S}$). Computation of the envelope is based on the ideas given in (Tao et al., 2010; Tao et al., 2015), viz., inversion of the tableau expansion rules used in computing $\mathcal{A}^*$. Moreover, we add two special expansion rules to deal with BCQs. The details are presented in Section 5.1.

The answer to the instance-checking queries posed to the KB is based on membership of those queries in the set $\mathcal{A}^* \setminus \mathbb{E}$. To answer BCQs, we use graph terminology: we express both the ABox $\mathcal{A}^* \setminus \mathbb{E}$ and the BCQ q as node-edge labeled graphs, see also (Ortiz and Simkus, 2012). The answer is based on the existence or non-existence of specific mappings between these two graphs. In more detail, if there is a (labeled) homomorphism from the query graph $G[q]$ (for q) to the ABox graph $G[\mathcal{A}^* \setminus \mathbb{E}]$ (for $\mathcal{A}^* \setminus \mathbb{E}$), then an answer to the query is "Yes"; if there are no such homomorphisms and there is a 'non-clashy' mapping[2] from $G[q]$ to $G[\mathcal{A}^* \setminus \mathbb{E}]$ then the answer to the query is "Unknown"; otherwise the answer is "No", see Section 5.2 for details. Based on the OW-semantics, we are able to provide an exact characterization of all answers. The rest of the paper is orga-

[1] A *DL-Lite*$_{\mathcal{R}}$ KB is said to be acyclic if neither $\exists P \sqsubseteq \exists P^-$ nor $\exists P^- \sqsubseteq \exists P$ follows from the KB.

[2] The term non-clashy mapping refers to a mapping which is not clashy, see Definition 4.6.

nized as follows: Section 2 explains the syntax of the language *DL-Lite*$_{\mathcal{R}}$ and its OW-semantics. In Section 3, we prove the soundness and completeness of the tableau algorithm that computes $\mathcal{A}^*$. Section 4 deals with syntax and semantics of BCQs. Section 5.1 introduces the secrecy preserving framework and explains the details of envelope construction. In Section 5.2, we explain the procedure to answer queries, and in Section 6, briefly we provide a summary and some directions for future research.

2 PRELIMINARIES: SYNTAX AND SEMANTICS OF *DL-LITE*$_{\mathcal{R}}$

2.1 Syntax

A vocabulary of *DL-Lite*$_{\mathcal{R}}$ is a triple $< N_O, N_C, N_R >$ of countably infinite, pairwise disjoint sets. The elements of N_O are called *objects* or *individual names*, the elements of N_C are called *concept names* (unary relation symbols) and the elements N_R are called *role names* (binary relation symbols). The set of *basic concepts* and the set of *basic roles*, respectively denoted by $\mathcal{BC}$ (generated by $\hat{B}$) and $\mathcal{BR}$ (generated by $\hat{R}$), are defined below by the grammar (a) where $A \in N_C$, $P \in N_{\mathcal{R}}$ and P^- stands for the inverse of the role name P. The set of *concept expressions* and *role expressions* in *DL-Lite*$_{\mathcal{R}}$, denoted by $\mathcal{C}$ (generated by $\hat{C}$) and $\mathcal{R}$ (generated by $\hat{E}$), is defined by the grammar (b).

$$(a)\quad \hat{B} ::= A \mid \exists \hat{R} \qquad (b)\quad \hat{C} ::= \hat{B} \mid \neg \hat{B}$$
$$\hat{R} ::= P \mid P^- \qquad \hat{E} ::= \hat{R} \mid \neg \hat{R}$$

Note that $\mathcal{BC} \subseteq \mathcal{C}$, and $\mathcal{BR} \subseteq \mathcal{R}$. For $C \in \mathcal{C}$ and $D \in \mathcal{BC}$, we write $\neg C$ to stand for D if $C = \neg D$ and for $\neg D$ if $C = D$. Similarly, for $E \in \mathcal{R}$ and $R \in \mathcal{BR}$, $\neg E$ denotes R if $E = \neg R$ and $\neg R$ if $E = R$. *Assertions* in *DL-Lite*$_{\mathcal{R}}$ are expressions of the form $C(a)$ and $E(a,b)$ where $a, b \in N_O$, $C \in \mathcal{C}$ and $E \in \mathcal{R}$; these are called *basic assertions* if $C \in \mathcal{BC}$ and $E \in \mathcal{BR}$.

There are two types of subsumptions in *DL-Lite*$_{\mathcal{R}}$,

a) *concept subsumptions* of the form $B \sqsubseteq C$ with $B \in \mathcal{BC}$ and $C \in \mathcal{C}$, and

b) *role subsumptions* of the form $R \sqsubseteq E$ with $R \in \mathcal{BR}$ and $E \in \mathcal{R}$.

Note the asymmetry between the left-hand side and the right-hand side of subsumptions in *DL-Lite*$_{\mathcal{R}}$.

2.2 Semantics

In this section we reformulate Kleene's 3-valued logic so as to provide semantics for *DL-Lite*$_{\mathcal{R}}$ which we feel is particularly well-suited in the context of OWA, see also (Tao et al., 2015). It allows us to give an "epistemic separation" between "known that Yes", "known that No" and "Unknown". We use the idea of *weak 3-partition* [3], defined as follows. Let X be a non-empty set, and A_1, A_2, A_3 (possibly empty) subsets of X. The ordered triple (A_1, A_2, A_3) is a *weak 3-partition* of X if

1. $A_1 \cup A_2 \cup A_3 = X$ and
2. $\forall i, j$ with $i \neq j$, $A_i \cap A_j = \emptyset$.

An *OW-interpretation* of the language *DL-Lite*$_{\mathcal{R}}$ is a tuple $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ where Δ is a non-empty domain and $\cdot^{\mathcal{I}}$ is an *interpretation function* such that

- $\forall a \in N_O$, $a^{\mathcal{I}} \in \Delta$,
- $\forall A \in N_C$, $A^{\mathcal{I}} = (A_N^{\mathcal{I}}, A_U^{\mathcal{I}}, A_Y^{\mathcal{I}})$ is a weak 3-partition of Δ, and
- $\forall P \in N_R$, $P^{\mathcal{I}} = (P_N^{\mathcal{I}}, P_U^{\mathcal{I}}, P_Y^{\mathcal{I}})$ is a weak 3-partition of $\Delta \times \Delta$.

We extend the interpretation function $\cdot^{\mathcal{I}}$ inductively to all concept and role expressions as follows. Let $C \in \mathcal{BC}$, $P \in N_R$, $R \in \mathcal{BR}$ and suppose that $C^{\mathcal{I}} = (C_N^{\mathcal{I}}, C_U^{\mathcal{I}}, C_Y^{\mathcal{I}})$, $P^{\mathcal{I}} = (P_N^{\mathcal{I}}, P_U^{\mathcal{I}}, P_Y^{\mathcal{I}})$ and $R^{\mathcal{I}} = (R_N^{\mathcal{I}}, R_U^{\mathcal{I}}, R_Y^{\mathcal{I}})$. Then,

- $(\neg C)^{\mathcal{I}} = (C_Y^{\mathcal{I}}, C_U^{\mathcal{I}}, C_N^{\mathcal{I}})$ and $(\neg R)^{\mathcal{I}} = (R_Y^{\mathcal{I}}, P_U^{\mathcal{I}}, R_N^{\mathcal{I}})$,
- $(P^-)^{\mathcal{I}} = ((P^-)_N^{\mathcal{I}}, (P^-)_U^{\mathcal{I}}, (P^-)_Y^{\mathcal{I}})$, where $(P^-)_X^{\mathcal{I}} = \{(a,b) | (b,a) \in P_X^{\mathcal{I}}\}$, $X \in \{N, U, Y\}$,
- $(\exists R)^{\mathcal{I}} = ((\exists R)_N^{\mathcal{I}}, (\exists R)_U^{\mathcal{I}}, (\exists R)_Y^{\mathcal{I}})$, where $(\exists R)_Y^{\mathcal{I}} = \{a | \exists b \in \Delta[(a,b) \in R_Y^{\mathcal{I}}]\}$, $(\exists R)_N^{\mathcal{I}} = \{a | \forall b \in \Delta[(a,b) \in R_N^{\mathcal{I}}]\}$ and $(\exists R)_U^{\mathcal{I}} = \Delta \setminus ((\exists R)_Y^{\mathcal{I}} \cup (\exists R)_N^{\mathcal{I}})$.

Remark. The subscripts "N", "U" and "Y" stand for "No", "Unknown" and "Yes", which represent the possible dispositions of a domain element with respect to a given OW-interpretation of a concept. Similarly, for roles. In addition, all the weak 3-partitions in this paper are ordered: First the N-component, second the U-component and third the Y-component.

Let $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ be an OW-interpretation, $B \in \mathcal{BC}$, $C \in \mathcal{C}$, $R \in \mathcal{BR}$, $E \in \mathcal{R}$ and $a, b \in N_{\mathcal{O}}$. We say that

- $\mathcal{I}$ *satisfies* $C(a)$, notation $\mathcal{I} \models C(a)$, if $a^{\mathcal{I}} \in C_Y^{\mathcal{I}}$;
- $\mathcal{I}$ *satisfies* $E(a,b)$, notation $\mathcal{I} \models E(a,b)$, if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in E_Y^{\mathcal{I}}$;
- $\mathcal{I}$ *satisfies* $B \sqsubseteq C$, notation $\mathcal{I} \models B \sqsubseteq C$, if $B_Y^{\mathcal{I}} \subseteq C_Y^{\mathcal{I}}$ and $C_N^{\mathcal{I}} \subseteq B_N^{\mathcal{I}}$, and

[3] It is weak in that we do not require that the sets A_i are non-empty.

- $\mathcal{I}$ *satisfies* $R \sqsubseteq E$, notation $\mathcal{I} \models R \sqsubseteq E$, if $R_Y^{\mathcal{I}} \subseteq E_Y^{\mathcal{I}}$ and $E_N^{\mathcal{I}} \subseteq R_N^{\mathcal{I}}$.

DL-Lite$_{\mathcal{R}}$ KB is a pair $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$, where $\mathcal{A}$, called the ABox [4], is a finite, non-empty set of assertions of the form $A(a)$, $\neg A(a)$, $P(a,b)$ and $\neg P(a,b)$ with $A \in N_C$, $P \in N_R$ and $a,b \in N_O$, and $\mathcal{T}$ is a finite set of concept and role subsumptions, called *TBox*. An OW-interpretation $\mathcal{I} = \left\langle \Delta, \cdot^{\mathcal{I}} \right\rangle$ is an *OW-model* of Σ, notation $\mathcal{I} \models \Sigma$, if for all $\alpha \in \mathcal{A} \cup \mathcal{T}$, $\mathcal{I} \models \alpha$. Let α be an assertion or a concept/role subsumption. We say that Σ *entails* α, notation $\Sigma \models \alpha$, if all OW-models of Σ satisfy α.

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

In (Calvanese et al., 2007), the authors had presented an algorithm based on query rewriting approach to answer CQs over *DL-Lite*$_{\mathcal{R}}$ KBs. The strategy used in this procedure is to convert the given CQ into a *union of conjunctive queries* (UCQ) by embedding the given TBox into the CQ. Note that the number of unions in the resulting UCQ could be exponential (depending on the TBox). Then, an answer to the given CQ is obtained by evaluating the UCQ over the given ABox. Lutz et.al., in (Lutz et al., 2008; Lutz et al., 2009) adopted the query rewriting approach to answer CQs in $\mathcal{EL}$ and $\mathcal{ELH}$ KBs respectively. Finding the set of all assertions entailed by an $\mathcal{EL}^+$ KB with acyclic TBox has been considered by Mei et al., see (Mei et al., 2009). Also in (Mei et al., 2009), the authors observed that even though the assumption of acyclicity restricts the expressive power of the language, in practice the idea is really useful, as it is expressive enough for the commonly used biomedical ontologies, e.g., Gene Ontology, SNOMED CT. In that paper, the authors had used a mixed approach which combines the computation of all assertions entailed by the given KB and the query rewriting method to answer the CQs. In our paper, we follow an approach which is different from the query rewriting approach to answer CQs. Below, we use a tableau-style procedure to construct a set of consequences of the given KB $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$, denoted by $\mathcal{A}^*$. Then, BCQs are answered based on the information available in the set $\mathcal{A}^*$, for more details see Section 6. Since our main interest in this paper is studying secrecy-preserving query answering, we shall henceforth assume that all TBoxes are acyclic; this will guarantee that $\mathcal{A}^*$ is finite.

[4] Note that we do not allow assertions of the form $\exists R(a)$ in the ABox $\mathcal{A}$.

Given $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$, before we start computing $\mathcal{A}^*$, we first arrange the individual names occurring in Σ, assertions in $\mathcal{A}$ and subsumptions in $\mathcal{T}$ in lexicographic order. We also program the algorithm which computes $\mathcal{A}^*$ in a way that selects these individual names, assertions and subsumptions in that order. This ordering will enable us to get a unique $\mathcal{A}^*$, see (Calvanese et al., 2007). The computation of $\mathcal{A}^*$ proceeds in several stages. In the first stage, $\mathcal{A}^*$ is initialized as $\mathcal{A}$ and expanded by exhaustively applying expansion rules listed in Figure 1. The resulting ABox is denoted by $\mathcal{A}_1^*$. The sets of all the individual names appearing in $\mathcal{A}$ and $\mathcal{A}_1^*$ are denoted by $\mathcal{O}_\Sigma$ and $\mathcal{O}^*$, respectively. $\mathcal{O}^*$ is initialized as $\mathcal{O}_\Sigma$ and expanded with applications of the $\sqsubseteq_{N\exists}$- and $\sqsubseteq_{\exists\exists}$-rules. An individual a is said to be *fresh* if $a \in \mathcal{O}^* \setminus \mathcal{O}_\Sigma$. It is important to note that all the fresh individuals are added in the first stage (Figure 1) and no new individuals are added in the following stages. This can be easily seen by inspecting the rules in Figures 1, 2 and 3. The rules are designed based on subsumptions present in the TBox $\mathcal{T}$. To name the rules in Figure 1, we adopt the following conventions. The first subscript of $\sqsubseteq$ represents the type of the symbol on the left hand side of the subsumption, and the second represents the type of the symbol on the right hand side. For example, the $\sqsubseteq_{N\exists}$ - rule has a concept name on the left hand side of the subsumption and existential restriction on the right hand side.

In order to write the rules more succinctly, we define two functions $\underline{inv}$ (standing for $\underline{inv}erse$) and $\underline{neg}$ (standing for $\underline{neg}ation$) as follows:

- for $P \in N_R$, $\underline{inv}(R,a,b) = \begin{cases} P(a,b) & \text{if } R = P, \\ P(b,a) & \text{if } R = P^- \end{cases}$
- $R \in \mathcal{BR}$, $\underline{neg}(E,a,b) = \begin{cases} \underline{inv}(R,a,b) & \text{if } E = R, \\ \neg\underline{inv}(R,a,b) & \text{if } E = \neg R \end{cases}$

For instance, $\underline{neg}(\neg P^-,a,b) = \neg\underline{inv}(P^-,a,b) = \neg P(b,a)$. In addition, we use L to denote either a concept name or a negation of concept name. We write $\neg L$ with the intended meaning that $\neg L = \neg A$ if $L = A$, and $\neg L = A$ if $L = \neg A$. To illustrate, we explain application of the rule $\sqsubseteq_{RE}$-rule. Let $P(a,b) \in \mathcal{A}^*$, $P^- \sqsubseteq \neg Q \in \mathcal{T}$ and $\neg Q(b,a) \notin \mathcal{A}^*$. Since, $\underline{neg}(P^-,b,a) = P(a,b)$ and $\underline{neg}(\neg Q,b,a) = \neg\underline{inv}(Q,b,a) = \neg Q(b,a)$, $\sqsubseteq_{RE}$-rule is applicable. Therefore, we add $\neg Q(b,a)$ to $\mathcal{A}^*$.

In the second stage, $\mathcal{A}_1^*$ is expanded by applying expansion rules listed in Figure 2. These rules deal with subsumptions in which the right hand side is a negation of existential restriction. For example, let $A(a) \in \mathcal{A}^*$, $A \sqsubseteq \exists P^- \in \mathcal{T}$ and for some $e \in \mathcal{O}^*$ such that $P(e,a) \notin \mathcal{A}^*$. Then, $\sqsubseteq_{N\sharp}$-rule is applicable and there-

fore we should add $P(e,a)$ to $\mathcal{A}^*$. The resulting ABox is denoted as $\mathcal{A}^*_{12}$. Observe that every application of a rule in Figure 2 adds at most $|\mathcal{O}^*|$ new assertions to $\mathcal{A}^*_1$. To name the rules in Figure 2, we adopt the same naming conventions as for the rules in Figure 1 except that the second symbol in the subscript represents the right hand side of $\sqsubseteq$: $\not\exists$ stands for a negated unqualified existential restriction.

$\sqsubseteq_{NL}$ – rule : if $A(a) \in \mathcal{A}^*$, $A \sqsubseteq L \in \mathcal{T}$ and $L(a) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{L(a)\}$;

$\sqsubseteq_{N\exists}$ – rule : if $A(a) \in \mathcal{A}^*$, $A \sqsubseteq \exists R \in \mathcal{T}$, and $\forall d \in \mathcal{O}^*$, $\underline{inv}(R,a,d) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{\underline{inv}(R,a,b)\}$ where b is fresh, and $\mathcal{O}^* := \mathcal{O}^* \cup \{b\}$;

$\sqsubseteq_{\exists L}$ – rule : if $\underline{inv}(R,a,b) \in \mathcal{A}^*$, $\exists R \sqsubseteq L \in \mathcal{T}$, and $L(a) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{L(a)\}$;

$\sqsubseteq_{\exists\exists}$ – rule : if $\underline{inv}(R,a,b) \in \mathcal{A}^*$, $\exists R \sqsubseteq \exists S \in \mathcal{T}$, and $\forall d \in \mathcal{O}^*$, $\underline{inv}(S,a,d) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{\underline{inv}(S,a,c)\}$ where c is fresh, and $\mathcal{O}^* := \mathcal{O}^* \cup \{c\}$;

$\sqsubseteq_{RE}$ – rule : if $\underline{inv}(R,a,b) \in \mathcal{A}^*$, $R \sqsubseteq E \in \mathcal{T}$ and $\underline{neg}(E,a,b) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{\underline{neg}(E,a,b)\}$.

Figure 1: We use the following conventions not stated explicitly within the individual rules: $A \in N_C$, $L \in \{A, \neg A \mid A \in N_C\}$, $R,S \in \mathcal{BR}$ and $E \in \mathcal{R}$.

$\sqsubseteq_{N\not\exists}$ – rule : Let $A(a) \in \mathcal{A}^*$ and $A \sqsubseteq \neg\exists R \in \mathcal{T}$. $\forall c \in \mathcal{O}^*$: if $\neg\underline{inv}(R,a,c) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{\neg\underline{inv}(R,a,c)\}$;

$\sqsubseteq_{\exists\not\exists}$ – rule : Let $\underline{inv}(R,a,b) \in \mathcal{A}^*$ and $\exists R \sqsubseteq \neg\exists S \in \mathcal{T}$. $\forall c \in \mathcal{O}^*$: if $\neg\underline{inv}(S,a,c) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{\neg\underline{inv}(S,a,c)\}$.

Figure 2: Computing $\mathcal{A}^*_{12}$: An application of each rule adds negation of role assertions for all $c \in \mathcal{O}^*$.

In the third stage, $\mathcal{A}^*_{12}$ is expanded by applying rules listed in Figure 3. The resulting final ABox is denoted as $\mathcal{A}^*$. To name the rules in Figure 3, we follow the previously adopted conventions. Additionally, negation in the subscript (see Figure 3) should be thought of as follows: For each rule in Figures 1 and 2, e.g. $\sqsubseteq_{NL}$-rule with $A \sqsubseteq L$, we have a corresponding $\sqsubseteq_{NL\neg}$-rule, which captures the effect of the subsumption $\neg L \sqsubseteq \neg A$ (which is not allowed in our syntax). It is easy to see that during the execution of rules in Figure 3 none of the rules in Figures 1 and 2 becomes applicable.

$\sqsubseteq_{NL\neg}$ – rule : if $\neg L(a) \in \mathcal{A}^*$, $A \sqsubseteq L \in \mathcal{T}$ and $\neg A(a) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{\neg A(a)\}$;

$\sqsubseteq_{N\exists\neg}$ – rule : if $\forall b \in \mathcal{O}^*$, $\neg\underline{inv}(R,a,b) \in \mathcal{A}^*$, $A \sqsubseteq \exists R \in \mathcal{T}$, and $\neg A(a) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{\neg A(a)\}$;

$\sqsubseteq_{\exists L\neg}$ – rule : Let $\neg L(a) \in \mathcal{A}^*$ and $\exists R \sqsubseteq L \in \mathcal{T}$. $\forall c \in \mathcal{O}^*$: if $\neg\underline{inv}(R,a,c) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{\neg\underline{inv}(R,a,c)\}$;

$\sqsubseteq_{\exists\exists\neg}$ – rule : Let $\forall b \in \mathcal{O}^*$, $\neg\underline{inv}(S,a,b) \in \mathcal{A}^*$ and $\exists R \sqsubseteq \exists S \in \mathcal{T}$. $\forall c \in \mathcal{O}^*$: if $\neg\underline{inv}(R,a,c) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{\neg\underline{inv}(R,a,c)\}$;

$\sqsubseteq_{RE\neg}$ – rule : if $\neg\underline{neg}(E,a,b) \in \mathcal{A}^*$, $R \sqsubseteq E \in \mathcal{T}$ and $\neg\underline{inv}(R,a,b) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{\neg\underline{inv}(R,a,b)\}$;

$\sqsubseteq_{N\not\exists\neg}$ – rule : if $\underline{inv}(R,a,b) \in \mathcal{A}^*$, $A \sqsubseteq \neg\exists R \in \mathcal{T}$ and $\neg A(a) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{\neg A(a)\}$.

$\sqsubseteq_{\exists\not\exists\neg}$ – rule : Let $\underline{inv}(S,a,b) \in \mathcal{A}^*$ and $\exists R \sqsubseteq \neg\exists S \in \mathcal{T}$. $\forall c \in \mathcal{O}^*$: if $\neg\underline{inv}(R,a,c) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{\neg\underline{inv}(R,a,c)\}$.

Figure 3: Computing $\mathcal{A}^*$: We use the same conventions as in Figure 1.

We say that $\mathcal{A}^*$ is *completed*, or that it is an *assertional closure of* $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$, if no assertion expansion rule is applicable. We denote by Λ the *tableau algorithm* which (lexicographically) applies assertion expansion rules, first those in Figure 1 then those in Figure 2 and finally those in Figure 3, until no further applications are possible. Since, as explained previously, Λ works in a lexicographic fashion, for a given KB $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$, it outputs a unique $\mathcal{A}^*$.

Since some of the expansion rules can in some cases be applied exponentially many times in the size of the KB, the size of $\mathcal{A}^*$ can be exponential in the size of the KB. As an example consider a $DL-Lite_R$ KB $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$, where $\mathcal{A} = \{A(a)\}$ and $\mathcal{T} = \{A \sqsubseteq \exists P_1, A \sqsubseteq \exists Q_1, \exists P_i^- \sqsubseteq \exists P_{i+1}, \exists P_i^- \sqsubseteq \exists Q_{i+1}, Q_i \sqsubseteq P_{i+1}, 1 \leq i \leq n\}$. Clearly, in this example the TBox $\mathcal{T}$ is acyclic and the size of the KB is linear in n. To compute $\mathcal{A}^*$ for

this KB, the $\sqsubseteq_{\exists\exists}$-rule has to be applied exponentially many times. It follows that $\mathcal{A}^*$ can be exponential in the size of Σ, implying that the computation of $\mathcal{A}^*$ could require exponential time as well.

Example 1. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$ be a DL-Lite$_\mathcal{R}$ KB, where $\mathcal{A}$ is defined by 1 and 2, and $\mathcal{T}$ is defined by 3, 4, 5 and 6,*

1	$A(a)$	*3*	$A \sqsubseteq B$	*5*	$\exists P^- \sqsubseteq \neg \exists R$
2	$D(b)$	*4*	$A \sqsubseteq \exists P$	*6*	$C \sqsubseteq \neg D$

Applying the assertion expansion rules in Figure 1, we can derive the following conclusions.

7 $B(a)$ $\sqsubseteq_{NL}$ *on 1,3*

8 $P(a,c)$, *c is fresh* $\sqsubseteq_{N\exists}$ *on 1,4*

Therefore $\mathcal{A}_1^* = \mathcal{A} \cup \{B(a), P(a,c)\}$. Now applying the assertion expansion rules in Figure 2 on $\mathcal{A}_1^*$, we calculate $\mathcal{A}_{12}^*$.

9 $\neg R(c,a), \neg R(c,b), \neg R(c,c)$ $\sqsubseteq_{\exists\exists}$ *on 8,5*

Thus $\mathcal{A}_{12}^* = \mathcal{A}_1^* \cup \{\neg R(c,a), \neg R(c,b), \neg R(c,c)\}$. Finally, using the assertion expansion rules in Figure 3 on $\mathcal{A}_{12}^*$, we get $\mathcal{A}^*$.

10 $\neg C(b)$ $\sqsubseteq_{NL\neg}$ *on 2,6*

Hence, $\mathcal{A}^* = \mathcal{A}_{12}^* \cup \{\neg C(b)\}$.

Observe that if we restrict the application of expansions rules in Figure 1, 2 and 3 to those ABox assertions involving only non-fresh individual names then we get $\{A(a), B(a), D(b), \neg C(b)\}$.

In general, if the computation is restricted to ABox assertions involving non-fresh individual names, then it is easy to see that the size of $\mathcal{A}^*$ is polynomial in the size of Σ and that it can be computed in polynomial time.

3.1 Soundness

The proof of the soundness of the tableau procedure Λ is split into two parts, dealing separately with rules in Figures 1 and 2 and Figure 3. The proofs of the next two Lemmas 1 and 2 are standard and are omitted.

Lemma 1 (**Soundness of Λ, Part A).** *Let $\mathcal{A}_{12}^*$ be a completed ABox obtained from Σ by first applying the rules listed in Figure 1 and then the rules of Figure 2. Then for every OW-model $\mathcal{I}$ of Σ, there is a OW-model $\mathcal{I}_{12}^*$ of Σ such that $\mathcal{I}_{12}^* \models \mathcal{A}_{12}^*$, where the domain of $\mathcal{I}_{12}^*$ is same as the domain of $\mathcal{I}$ and $\mathcal{I}_{12}^*$ remains same as $\mathcal{I}$ except for the interpretation of fresh individuals.*

Let $\mathcal{O}^*$ be the set of individual names that occur in the completed ABox $\mathcal{A}_{12}^*$. We define a new OW-interpretation $\mathcal{I}^* = \left\langle \Delta^*, \cdot^{\mathcal{I}^*} \right\rangle$, where $\Delta^* = \mathcal{I}_{12}^*(\mathcal{O}^*)$, i.e., Δ^* is precisely the set of those elements of Δ that are interpretations of individuals in $\mathcal{O}^*$. The interpretation function $\cdot^{\mathcal{I}^*}$ is defined as a restriction of $\mathcal{I}_{12}^*$ to Δ^*:

(i) $\forall a \in \mathcal{O}^*[a^{\mathcal{I}^*} = a^{\mathcal{I}_{12}^*}]$;

(ii) $\forall A \in N_C[(A_N^{\mathcal{I}^*} = A_N^{\mathcal{I}_{12}^*} \cap \Delta^*, A_U^{\mathcal{I}^*} = A_U^{\mathcal{I}_{12}^*} \cap \Delta^*, A_Y^{\mathcal{I}^*} = A_Y^{\mathcal{I}_{12}^*} \cap \Delta^*)]$;

(iii) $\forall P \in N_R[P_N^{\mathcal{I}^*} = P_N^{\mathcal{I}_{12}^*} \cap (\Delta^* \times \Delta^*), P_U^{\mathcal{I}^*} = P_U^{\mathcal{I}_{12}^*} \cap (\Delta^* \times \Delta^*), P_Y^{\mathcal{I}^*} = P_Y^{\mathcal{I}_{12}^*} \cap (\Delta^* \times \Delta^*)]$ and

(iv) $\mathcal{I}^*$ is extended to compound concepts and roles as in Section 2.2.

Since every weak 3-partition of Δ induces a weak 3-partition of Δ^*, we have the following consequence of Lemma 1,

Corollary 1. *$\mathcal{I}^*$ is an OW-model of $\left\langle \mathcal{A}_{12}^*, \mathcal{T} \right\rangle$.*

Lemma 2 (**Soundness of Λ, Part B).** *Let $\mathcal{A}^*$ be the completed ABox obtained from $\mathcal{A}_{12}^*$ by applying the rules listed in Figure 3. For any OW-model $\mathcal{I}$ of Σ, let $\mathcal{I}^* = \left\langle \Delta^*, \cdot^{\mathcal{I}^*} \right\rangle$ be an OW-interpretation as defined above. Then, $\mathcal{I}^*$ is an OW-model of Σ and $\mathcal{I}^* \models \mathcal{A}^*$.*

In summary, given an OW-model $\mathcal{I}$ of Σ, using the proof of Lemma 1, we transform $\mathcal{I}$ to another OW-model $\mathcal{I}_{12}^*$ of Σ such that $\mathcal{I}_{12}^* \models \mathcal{A}_{12}^*$, where the domain of $\mathcal{I}_{12}^*$ is same as the domain of $\mathcal{I}$. In fact, $\mathcal{I}_{12}^*$ remains the same as $\mathcal{I}$ except for the interpretation of fresh individuals. Moreover, $\mathcal{I}_{12}^*$ is constructed in a canonical fashion, i.e., it is uniquely determined from $\mathcal{I}$. Having obtained $\mathcal{I}_{12}^*$, using Lemma 2, we modify $\mathcal{I}_{12}^*$ to obtain yet another OW-model $\mathcal{I}^*$ of Σ such that $\mathcal{I}^* \models \mathcal{A}^*$, where the domain of $\mathcal{I}^*$ was defined to be $\mathcal{I}_{12}^*(\mathcal{O}^*)$. We use the notation $\Sigma \models^* \alpha$, where α is a concept (or role) name assertion or negation of a concept (or role) name assertion, to represent the following statement: For every OW-model $\mathcal{I}$ of Σ, $\mathcal{I}^*$ is an OW-model of Σ and $\mathcal{I}^* \models \alpha$. We can combine Lemma 1 and Lemma 2 into a single theorem.

Theorem 1. *(***Soundness of Λ***): Let $\mathcal{A}^*$ be a completed ABox obtained from Σ by first applying the rules listed in Figure 1, then rules listed in Figure 2, and finally the rules listed in Figure 3. Then $\Sigma \models^* \mathcal{A}^*$, i.e., for every $\alpha \in \mathcal{A}^*$, $\Sigma \models^* \alpha$.*

3.2 Completeness

To prove the completeness of Λ, we first define a *canonical OW-interpretation* $\mathcal{J} = \left\langle \Delta, \cdot^{\mathcal{J}} \right\rangle$ for a completed ABox $\mathcal{A}^*$ as follows:

- $\Delta = \mathcal{O}^* = \{a \in N_O |\ a \text{ occurs in } \mathcal{A}^*\}$;
- $a^{\mathcal{J}} = a$, for each individual name $a \in \mathcal{O}^*$;
- for $A \in N_C$, $A^{\mathcal{J}} = (A_N^{\mathcal{J}}, A_U^{\mathcal{J}}, A_Y^{\mathcal{J}})$, where

  $A_Y^{\mathcal{J}} = \{a |\ A(a) \in \mathcal{A}^*\}$,

  $A_N^{\mathcal{J}} = \{a |\ \neg A(a) \in \mathcal{A}^*\}$ and

  $A_U^{\mathcal{J}} = (\Delta \setminus A_Y^{\mathcal{J}}) \setminus A_N^{\mathcal{J}}$;

- for $P \in N_R$, $P^{\mathcal{J}} = (P_N^{\mathcal{J}}, P_U^{\mathcal{J}}, P_Y^{\mathcal{J}})$, where
  $P_Y^{\mathcal{J}} = \{(a,b) | P(a,b) \in \mathcal{A}^*\}$,
  $P_N^{\mathcal{J}} = \{(a,b) | \neg P(a,b) \in \mathcal{A}^*\}$ and
  $P_U^{\mathcal{J}} = ((\Delta \times \Delta) \setminus P_Y^{\mathcal{J}}) \setminus P_N^{\mathcal{J}}$;
- $\mathcal{J}$ is extended to compound concepts and roles as in Section 2.2.

The proof that $\mathcal{J}$ is a OW-model of Σ is standard and is omitted.

Lemma 3. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$ be a DL-Lite$_\mathcal{R}$ KB. Then $\forall \alpha \in \mathcal{A} \cup \mathcal{T}$, $\mathcal{J} \models \alpha$.*

Theorem 2 (Completeness of Λ). *Let $\mathcal{A}^*$ be a completed ABox obtained from Σ by applying Λ. Let α be a concept (or role) name assertion or negation of a concept (or role) name assertion*[5]*. Then $\Sigma \models^* \alpha \Rightarrow \alpha \in \mathcal{A}^*$.*

Proof. Let $\mathcal{J}$ be the canonical model of Σ as defined above, and let α be an assertion as in the statement of the theorem. Suppose $\Sigma \models^* \alpha$. By Lemma 3, $\mathcal{J} \models \Sigma$ and hence $\mathcal{J}^* \models \alpha$. Since $\mathcal{A}^*$ is completed, $\mathcal{J}^* = \mathcal{J}$, and so $\mathcal{J} \models \alpha$. In the following, we argue by cases for different α.

- $\alpha = A(a)$, $A \in N_C$. Then, $\mathcal{J} \models A(a) \Rightarrow a \in A_Y^{\mathcal{J}} \Rightarrow A(a) \in \mathcal{A}^*$.
- $\alpha = \neg A(a)$, $A \in N_C$. Then, $\mathcal{J} \models \neg A(a) \Rightarrow a \in A_N^{\mathcal{J}} \Rightarrow \neg A(a) \in \mathcal{A}^*$.
- $\alpha = P(a,b)$, $P \in N_R$. Then, $\mathcal{J} \models P(a,b) \Rightarrow (a,b) \in P_Y^{\mathcal{J}} \Rightarrow P(a,b) \in \mathcal{A}^*$.
- $\alpha = \neg P(a,b)$, $P \in N_R$. Then, $\mathcal{J} \models \neg P(a,b) \Rightarrow (a,b) \in P_N^{\mathcal{J}} \Rightarrow \neg P(a,b) \in \mathcal{A}^*$.

□

4 GRAPH REPRESENTATION OF ABoxes AND BCQs OVER *DL-Lite*$_\mathcal{R}$ KBs

In this section, we will use node-edge labeled directed graph to represent the completed ABox $\mathcal{A}^*$ as well as Boolean conjunctive queries (BCQs), see (Ortiz and Simkus, 2012) for similar representations. This helps "visualize" reasoning about such queries as well as being useful in formulating precise conditions for answering BCQs with 'Yes', 'No' and 'Unknown'.

The ABox graph for $\mathcal{A}^*$ is node-edge labeled digraph $G[\mathcal{A}^*] = (V[\mathcal{A}^*], E[\mathcal{A}^*], L[\mathcal{A}^*])$ with nodes $V[\mathcal{A}^*] = \mathcal{O}^*$ and edges $E[\mathcal{A}^*] = \{(a,b) \mid R(a,b) \in \mathcal{A}^*$, for some $R \in \mathcal{R}\}$, where each node $a \in V[\mathcal{A}^*]$ is labeled with the set of literals $L[\mathcal{A}^*](a) = \{ L \mid L(a) \in \mathcal{A}^*\}$ and each directed edge $(a,b) \in E[\mathcal{A}^*]$ is labeled with a set of roles $L[\mathcal{A}^*](a,b) = \{ R \mid R(a,b) \in \mathcal{A}^*\}$.

Example 2. *Let $\mathcal{A}^* = \{A(a), \neg D(a), B(b), F(b), H(d), P(a,b), Q(a,b), P(b,c), Q(b,c), R(a,d), \neg S(a,d), \neg Q(c,c)\}$. Then ABox graph for $\mathcal{A}^*$ is:*

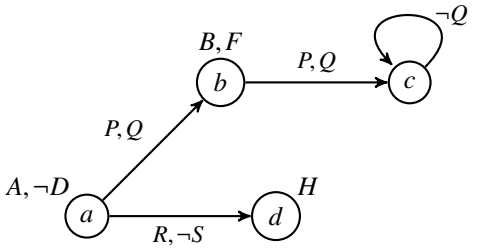

Figure 4: The ABox graph $G[\mathcal{A}^*]$ for the given ABox, $\mathcal{A}^*$.

We next define the syntax and semantics of Boolean conjunctive queries. Let N_V denote a countably infinite set of variables.

Definition 1. *A Boolean conjunctive query over DL-Lite$_\mathcal{R}$ is a finite expression of the form $\exists y_1, y_2, ..., y_n[\bigwedge_{i=1}^{k} A_i(\zeta_i) \wedge \bigwedge_{j=1}^{m} P_j(\eta_j, \mu_j)]$, where*

- *$A_i \in N_C$ for $1 \leq i \leq k$, $P_j \in N_R$ for $1 \leq j \leq m$ and $y_l \in N_V$, $1 \leq l \leq n$,*
- *$\zeta_i, \eta_j, \mu_j \in \{y_1, y_2, ..., y_n\} \cup N_O$ for $1 \leq i \leq k$ and $1 \leq j \leq m$.*

Query atoms of a BCQ q are of two sorts: *concept atoms* $A(v)$, and *role atoms* $P(u,v)$, where $u, v \in N_V \cup N_O$, $A \in N_C$ and $P \in N_R$. By $Atoms(q)$ we denote the set of concept and role atoms occurring in q. For instance the concept atoms in the BCQ $q = \exists y, z[A(a) \wedge B(y) \wedge B(z) \wedge P(a,y) \wedge Q(a,z) \wedge P(z,y)]$ are: $A(a)$, $B(y)$ and $B(z)$ and the role atoms are: $P(a,y)$, $Q(a,z)$ and $P(z,y)$.

As was the case with the ABox, we can represent the BCQ as a node-edge labeled directed graph capturing the syntactic structure of the query. The *query graph* of a BCQ q is the node-edge labeled directed graph $G[q] = (V[q], E[q], L[q])$ with nodes $V[q] = \{v \in N_V \cup N_O \mid v$ occurs in $q\}$ and edges $E[q] = \{(u,v) \mid$ for some role name P, $P(u,v) \in Atoms(q)\}$; each node $v \in V[q]$ is labeled with the set of concept names $L[q](v) = \{A | A(v) \in Atoms(q)\}$ and each edge $(u,v) \in E[q]$ is labeled with the set of role names $L[q](u,v) = \{P \mid P(u,v) \in Atoms(q)\}$.

[5]Recall that assertions of the form $\exists R(a)$ do not belong to $\mathcal{A}^*$.

Example 3. *The query graph of the BCQ q mentioned above is:*

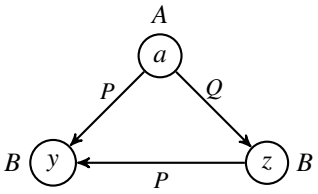

Figure 5: The query graph of $q = \exists y,z[A(a) \wedge B(y) \wedge B(z) \wedge P(a,y) \wedge Q(a,z) \wedge P(z,y)]$.

An interpretation of a BCQ q is provided by an OW-interpretation $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ together with a *valuation* which is a function $\pi : V[q] \to \Delta$ such that $\pi(a) = a^{\mathcal{I}}$ for each individual $a \in V[q] \cap N_O$. We say that $(\mathcal{I},\pi)$ *satisfies* $A(v)$, notation $(\mathcal{I},\pi) \models A(v)$, if $\pi(v) \in A_Y^{\mathcal{I}}$. $(\mathcal{I},\pi)$ *falsifies* $A(v)$, notation $(\mathcal{I},\pi) \models \neg A(v)$, if $\pi(v) \in A_N^{\mathcal{I}}$. Similarly, $(\mathcal{I},\pi)$ *satisfies* $P(u,v)$, notation $(\mathcal{I},\pi) \models P(u,v)$, if $(\pi(u),\pi(v)) \in P_Y^{\mathcal{I}}$ and $(\mathcal{I},\pi)$ *falsifies* $P(u,v)$, notation $(\mathcal{I},\pi) \models \neg P(u,v)$, if $(\pi(u),\pi(v)) \in P_N^{\mathcal{I}}$. We say that $(\mathcal{I},\pi)$ *satisfies* q, notation $(\mathcal{I},\pi) \models q$, if $(\mathcal{I},\pi) \models \alpha$ for every $\alpha \in Atoms(q)$. $(\mathcal{I},\pi)$ *falsifies* q, notation $(\mathcal{I},\pi) \not\models q$, if $(\mathcal{I},\pi)$ falsifies some atom $\alpha \in Atoms(q)$. $\mathcal{I}$ *satisfies* q, notation $\mathcal{I} \models q$, if there exists a valuation $\pi : V[q] \to \Delta$ such that $(\mathcal{I},\pi) \models q$. In this case, we say that $\mathcal{I}$ is an OW-model of q. $\mathcal{I}$ *falsifies* q, notation $\mathcal{I} \not\models q$, if for all valuations $\pi : V[q] \to \Delta$, $(\mathcal{I},\pi) \not\models q$.

Recall (subsection 3.1) that given any OW-model $\mathcal{I}$ of Σ we have defined a unique more compact OW-model $\mathcal{I}^*$ and we introduced the notation $\Sigma \models^* \alpha$ to mean that for any OW-model $\mathcal{I}$ of Σ, $\mathcal{I}^*$ is an OW-model of Σ and $\mathcal{I}^* \models \alpha$. Finally, a BCQ q is *entailed* from Σ, notation $\Sigma \models^* q$, if for every OW-model $\mathcal{I}$ of Σ, $\mathcal{I}^* \models q$. A BCQ q is *disentailed* from Σ, notation $\Sigma \not\models^* q$, if for every OW-model $\mathcal{I}$ of Σ, $\mathcal{I}^* \not\models q$, i.e., $\mathcal{I}^*$ falsifies q. Thus the property $\Sigma \not\models^* q$ precisely captures the requirement for answering the query q with "No".

Notation. We write $h : V[q] \underset{c}{\to} V[\mathcal{A}^*]$ to denote the fact that h is a mapping $h : V[q] \to V[\mathcal{A}^*]$ which "respects constants", i.e. $h(a) = a$, for every individual $a \in V[q] \cap N_O$.

Definition 2. *Mapping $h : V[q] \underset{c}{\to} V[\mathcal{A}^*]$ is a labeled graph homomorphism, if*

- *for every node v in $V[q]$, $L[q](v) \subseteq L[\mathcal{A}^*](h(v))$, and*
- *for every edge (u,v) in $E[q]$, $L[q](u,v) \subseteq L[\mathcal{A}^*](h(u),h(v))$.*

In the next two theorems we provide a complete characterization of entailment and disentailment of BCQs in terms of properties of mappings $h : V[q] \underset{c}{\to} V[\mathcal{A}^*]$.

Theorem 3. *Let q be a BCQ and Σ a DL-Lite$_{\mathcal{R}}$ KB. Then, $\Sigma \models^* q$ iff there exists a labeled graph homomorphism $h : V[q] \underset{c}{\to} V[\mathcal{A}^*]$.*

Proof. ($\Rightarrow$) Suppose $\Sigma \models^* q$ and let $\mathcal{J} = \langle \Delta, \cdot^{\mathcal{J}} \rangle$ be the canonical OW-model of Σ, see Section 3.2. Then, $\mathcal{J}^* = \mathcal{J}$, and by hypothesis, $\mathcal{J} \models q$. Hence, for some valuation $\pi : V[q] \to \mathcal{O}^* = V[\mathcal{A}^*]$, $(\mathcal{J},\pi) \models \alpha$, for every $\alpha \in Atoms(q)$. Note that $\pi : V[q] \underset{c}{\to} V[\mathcal{A}^*]$. Now, let $v \in V[q]$ and $A(v) \in Atoms(q)$. Then, $(\mathcal{J},\pi) \models A(v) \Rightarrow \pi(v) \in A_Y^{\mathcal{J}} \Rightarrow A(\pi(v)) \in \mathcal{A}^* \Rightarrow A \in L[\mathcal{A}^*](\pi(v))$. Similarly, for $u,v \in V[q]$ with $P(u,v) \in Atoms(q)$: $(\mathcal{J},\pi) \models P(u,v) \Rightarrow (\pi(u),\pi(v)) \in P_Y^{\mathcal{J}} \Rightarrow P(\pi(u),\pi(v)) \in \mathcal{A}^* \Rightarrow P \in L[\mathcal{A}^*](\pi(u),\pi(v))$. It follows that π is a labeled graph homomorphism.

($\Leftarrow$) Assume that $h : V[q] \underset{c}{\to} V[\mathcal{A}^*]$ is a labeled graph homomorphism and let $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ be an arbitrary OW-model of Σ. By Lemma 2, $\mathcal{I}^* = \langle \Delta^*, \cdot^{\mathcal{I}^*} \rangle$, with $\Delta^* = \mathcal{I}^*(\mathcal{O}^*)$, is an OW-model of Σ and $\mathcal{I}^* \models \mathcal{A}^*$. Since $(\mathcal{I}^* \circ h) : V[q] \to \Delta^*$, we have $(\mathcal{I}^* \circ h)(a) = \mathcal{I}^*(h(a)) = a^{\mathcal{I}^*}$ for all $a \in V[q] \cap N_O$. I.e., $\mathcal{I}^* \circ h$ is a valuation. It remains to show that $\mathcal{I}^*$ is an OW-model of q. Let $v \in V[q]$ and $A \in L[q](v)$. Then, by the definition of labeled homomorphism, $A \in L[\mathcal{A}^*](h(v)) \Rightarrow A(h(v)) \in \mathcal{A}^* \Rightarrow h(v)^{\mathcal{I}^*} \in A_Y^{\mathcal{I}^*} \Rightarrow (\mathcal{I}^* \circ h)(v) \in A_Y^{\mathcal{I}^*} \Rightarrow (\mathcal{I}^*, (\mathcal{I}^* \circ h)) \models A(v)$. Similarly, for $u,v \in V[q]$ with $P \in L[q](u,v)$: $P \in L[\mathcal{A}^*](h(u),h(v)) \Rightarrow P(h(u),h(v)) \in \mathcal{A}^* \Rightarrow (h(u)^{\mathcal{I}^*}, h(v)^{\mathcal{I}^*}) \in P_Y^{\mathcal{I}^*} \Rightarrow ((\mathcal{I}^* \circ h)(u), (\mathcal{I}^* \circ h)(v)) \in P_Y^{\mathcal{I}^*} \Rightarrow (\mathcal{I}^*, (\mathcal{I}^* \circ h)) \models P(u,v)$. Thus, $\Sigma \models^* q$. □

Next we define mappings that cannot be extended to labeled homomorphisms and prove a tight connection between such mappings and disentailment.

Definition 3. *A mapping $f : V[q] \underset{c}{\to} V[\mathcal{A}^*]$ is said to be clashy, if*

- *there exist $v \in V[q]$ and $A \in L[q](v)$ such that $\neg A \in L[\mathcal{A}^*](f(v))$, or*
- *there exist $u,v \in V[q]$ and $P \in L[q]((u,v))$ such that $\neg P \in L[\mathcal{A}^*]((f(u),f(v)))$.*

Theorem 4. *Let q be a BCQ and Σ a DL-Lite$_{\mathcal{R}}$ KB. Then, $\Sigma \not\models^* q$ iff every mapping $f : V[q] \underset{c}{\to} V[\mathcal{A}^*]$ is clashy.*

Proof. ($\Rightarrow$) Assume $\Sigma \not\models^* q$ and let $\mathcal{J} = \langle \Delta, \cdot^{\mathcal{J}} \rangle$ be the canonical OW-model of Σ. Then, $\mathcal{J}^* = \mathcal{J}$ and so for every valuation $\tau : V[q] \to \Delta^*$, there is an $\alpha \in Atoms(q)$ such that $(\mathcal{J},\tau) \models \neg\alpha$. Since $\Delta^* = \mathcal{J}(\mathcal{O}^*) = \mathcal{O}^* = V(\mathcal{A}^*)$ and $\tau(a) = a^{\mathcal{J}} = a$ for all $a \in V[q] \cap N_O$,

$\tau : V[q] \underset{c}{\rightarrow} V[\mathcal{A}^*]$ and it follows that τ is clashy. Moreover, since τ is arbitary the conclusion follows.

($\Leftarrow$) Suppose now that every mapping $f : V[q] \underset{c}{\rightarrow} V[\mathcal{A}^*]$ is clashy. Let $\mathcal{I} = \left\langle \Delta, \cdot^{\mathcal{I}} \right\rangle$ be an arbitrary OW-model of Σ. By Lemma 2, $\mathcal{I}^* = \left\langle \Delta^*, \cdot^{\mathcal{I}^*} \right\rangle$ with $\Delta^* = \mathcal{I}^*(\mathcal{O}^*)$ is an OW-model of Σ such that $\mathcal{I}^* \models \mathcal{A}^*$. Let $\pi : V[q] \rightarrow \Delta^*$ be an arbitrary valuation and define the mapping $g_\pi : V[q] \rightarrow V[\mathcal{A}^*]$ by

$$g_\pi(v) = \begin{cases} a & \text{if } v = a \in N_O \cap V[q] \\ c & \text{if } v \in N_V \cap V[q], \end{cases}$$

where $\pi(v) = c^{\mathcal{I}^*}$ and c be the first constant that satisfies in some arbitrary (but fixed) total ordering of $\mathcal{O}^*$, see the end of Section 3. It is easy to check that $\pi = \mathcal{I}^* \circ g_\pi$ (in other words, π factors via $V[\mathcal{A}^*]$). Since, by assumption, g_π is clashy, for some $A(v) \in Atoms(q)$, $\neg A \in L[\mathcal{A}^*](g_\pi(v))$ or for some $P(u,v) \in Atoms(q)$, $\neg P \in L[\mathcal{A}^*]((g_\pi(u), g_\pi(v)))$. In the first case, $\neg A(g_\pi(v)) \in \mathcal{A}^* \Rightarrow g_\pi(v)^{\mathcal{I}^*} \in A_N^{\mathcal{I}^*} \Rightarrow \pi(v) \in A_N^{\mathcal{I}^*}$ implying, $(\mathcal{I}^*, \pi) \models \neg A(v)$. In the second case, $\neg P((g_\pi(u), g_\pi(v))) \in \mathcal{A}^* \Rightarrow (g_\pi(u)^{\mathcal{I}^*}, g_\pi(v)^{\mathcal{I}^*}) \in P_N^{\mathcal{I}^*} \Rightarrow (\pi(u), \pi(v)) \in P_N^{\mathcal{I}^*}$ implying, $(\mathcal{I}^*, \pi) \models \neg P(u,v)$. It follows that, $\Sigma \mid\!\models^* q$. □

5 SECRECY-PRESERVING REASONING: ENVELOPES AND QUERY ANSWERING

5.1 Computing Envelopes

As mentioned before the main goal of the paper is to study secrecy-preserving reasoning. The tool we use are the construction of envelopes (Tao et al., 2010; Tao et al., 2015; Krishnasamy Sivaprakasam and Slutzki, 2016). This is discussed in detail in Section 5.1. Once envelope is available, query answering becomes easy. Given a knowledge base Σ and a finite secrecy set $\mathbb{S}$ consisting of assertions in $\mathcal{A}^*$ and BCQs, the goal is to answer queries while preserving secrecy. Here we assume that $\mathcal{A}^*$ has been computed previously. Our approach is to compute a subset $\mathbb{E} \subseteq \mathcal{A}^*$, called the *secrecy envelope* for $\mathbb{S}$, so that by protecting $\mathbb{E}$, the querying agent cannot logically infer any assertions in $\mathbb{S}$, see (Tao et al., 2010; Tao et al., 2015). It is interesting to note that, though the BCQs in $\mathbb{S}$ are not in $\mathbb{E}$, we can store the information pertinent to answering BCQs in $\mathbb{E}$. The OWA plays a vital role in protecting secret information when query answering is the main objective. When answering a query with "Unknown", the querying agent cannot differentiate between the following cases: (1) the case that the answer to the query is actually unknown to the KB reasoner and (2) the case that the answer is being protected in order to maintain secrecy.

Formally, the secrecy set is made of two parts, $\mathbb{S} = S_\Sigma \cup S_{CQ}$, where $S_\Sigma \subseteq \mathcal{A}_0^* \subseteq \mathcal{A}^*$ with $\mathcal{A}_0^*$ the subset of assertions which do not involve fresh individuals, and S_{CQ} is a finite set of BCQs. Clearly, the size of $\mathcal{A}_0^*$ is polynomial to the size of the input KB.

Definition 4. *Given a knowledge base $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$ and a finite secrecy set $\mathbb{S} = S_\Sigma \cup S_{CQ}$, where $S_\Sigma \subseteq \mathcal{A}^*$ and S_{CQ} is a finite set of BCQs, a secrecy envelope for $\mathbb{S}$, denoted by $\mathbb{E}$, is a set of assertions having the following properties:*

1 $S_\Sigma \subseteq \mathbb{E} \subseteq \mathcal{A}^*$,
2 *for every $\alpha \in \mathbb{E}$, $\mathcal{A}^* \setminus \mathbb{E} \not\models^* \alpha$, and*
3 *for every $q \in S_{CQ}$, $\mathcal{A}^* \setminus \mathbb{E} \not\models^* q$ and $\mathcal{A}^* \setminus \mathbb{E} \mid\!\not\models^* q$.*

Property 2 says that no information in $\mathbb{E}$ can be entailed from $\mathcal{A}^* \setminus \mathbb{E}$. Property 3 makes sure that BCQs in S_{CQ} can neither be entailed nor disentailed from $\mathcal{A}^* \setminus \mathbb{E}$. To compute an envelope, we use the idea of inverting assertion expansion rules (see (Tao et al., 2010), where this approach was first utilized). Induced by the tableau expansion rules in Figure 1 (except for the rules $\sqsubseteq_{N\exists}$ and $\sqsubseteq_{\exists\exists}$) and in Figure 2, we have the corresponding "inverted" secrecy closure rules in Figure 6. The reason for the omission of secrecy closure rules corresponding to the rules $\sqsubseteq_{N\exists}$ and $\sqsubseteq_{\exists\exists}$ is that an application of these rules results in adding assertions with fresh individual names. By the *hidden name assumptions* (HNA), the querying agent is barred from asking any queries that involve fresh individual names, see also (Tao et al., 2010).

As an illustration of a secrecy closure rules in Figure 6, consider the $\sqsubseteq^{\leftarrow}_{N\sharp}$-rule. Let $\neg P(a,b) \in \mathbb{E}$, $A \sqsubseteq \neg\exists P \in \mathcal{T}$ and $A(a) \in \mathcal{A}^* \setminus \mathbb{E}$. If the querying agent asks the query $q = \neg P(a,b)$, then the reasoner $\Re$ could answer "Yes". This is because of the $\sqsubseteq_{N\sharp}$-rule and the fact that $A(a) \notin \mathbb{E}$. So, to protect $\neg P(a,b)$, we have to put $A(a)$ in $\mathbb{E}$. Similarly, in Figure 7 the secrecy closure rules are given corresponding to the rules in Figure 3. For instance, consider the $\sqsubseteq^{\leftarrow}_{\exists L\neg}$-rule. Let $\neg P(a,b) \in \mathbb{E}$, $\exists P \sqsubseteq B \in \mathcal{T}$ and $\neg B(a) \in \mathcal{A}^* \setminus \mathbb{E}$. If the querying agent asks the query $q = \neg P(a,b)$, then the reasoner $\Re$ could answer "Yes". This is because of the $\sqsubseteq_{\exists L\neg}$-rule and the fact that $\neg B(a) \notin \mathbb{E}$. So, to protect $\neg P(a,b)$, we have to put $\neg B(a)$ in $\mathbb{E}$. In both cases, these secrecy closure rules are named by adding the superscript $\leftarrow$ in the name of the corresponding assertion expansion rules.

Rules that specifically deal with BCQs are given in Figure 8 and have been designed to protect BCQ's in S_{CQ}. Few words of explanation may be helpful in

$\sqsubseteq^{\leftarrow}_{NL}$ – rule : if $L(a) \in \mathbb{E}$, $A \sqsubseteq L \in \mathcal{T}$ and
$A(a) \in \mathcal{A}^* \setminus \mathbb{E}$, then $\mathbb{E} := \mathbb{E} \cup \{A(a)\}$;

$\sqsubseteq^{\leftarrow}_{\exists L}$ – rule : if $L(a) \in \mathbb{E}$, $\exists R \sqsubseteq L \in \mathcal{T}$ and
$\underline{inv}(R,a,c) \in \mathcal{A}^* \setminus \mathbb{E}$, for some $c \in \mathcal{O}^*$
then $\mathbb{E} := \mathbb{E} \cup \{\underline{inv}(R,a,c)\}$;

$\sqsubseteq^{\leftarrow}_{RE}$ – rule : if $\underline{neg}(E,a,b) \in \mathbb{E}$, $R \sqsubseteq E \in \mathcal{T}$ and
$\underline{inv}(R,a,b) \in \mathcal{A}^* \setminus \mathbb{E}$,
then $\mathbb{E} := \mathbb{E} \cup \{\underline{inv}(R,a,b)\}$;

$\sqsubseteq^{\leftarrow}_{N\not\exists}$ – rule : if $\neg\underline{inv}(R,a,b) \in \mathbb{E}$, $A \sqsubseteq \neg\exists R \in \mathcal{T}$
and $A(a) \in \mathcal{A}^* \setminus \mathbb{E}$,
then $\mathbb{E} := \mathbb{E} \cup \{A(a)\}$;

$\sqsubseteq^{\leftarrow}_{\exists\not\exists}$ – rule : if $\neg\underline{inv}(S,a,b) \in \mathbb{E}$, $\exists R \sqsubseteq \neg\exists S \in \mathcal{T}$
and $\underline{inv}(R,a,c) \in \mathcal{A}^* \setminus \mathbb{E}$,
for some $c \in \mathcal{O}^*$,
then $\mathbb{E} := \mathbb{E} \cup \{\underline{inv}(R,a,c)\}$.

Figure 6: Secrecy closure rules obtained by inverting rules in Figures 1 and 2.

understanding BCQ-rules. Let $q \in S_{CQ}$ be a BCQ. To protect q, we use BCQ$_h$-rule which "disrupts" each homomorphism $h : G[q] \to G[\mathcal{A}^* \setminus \mathbb{E}]$ and adds to $\mathbb{E}$ one of the atoms of q (whose variables are evaluated under h). Similarly, in the BCQ$_c$-rule, we pick an arbitrary clashy mapping $g : G[q] \to G[\mathcal{A}^* \setminus \mathbb{E}]$ and make it into a non-clashy mapping: This can be done by considering all the clashing atoms of q under g ($A \in L[q](v)$ and $\neg A \in L[\mathcal{A}^* \setminus \mathbb{E}](g(v))$, or $P \in L[q]((u,v))$ and $\neg P \in L[\mathcal{A}^* \setminus \mathbb{E}](g(u),g(v))$) and adding them to $\mathbb{E}$.

The computation of $\mathbb{E}$ proceeds in two stages. In the first stage, $\mathbb{E}$ is initialized as S_Σ and expanded by using secrecy closure rules listed in Figures 6 and 7. In the second stage, $\mathbb{E}$ is expanded by using BCQ$_h$ and BCQ$_c$-rules. We denote by Λ_S the tableau algorithm which computes the envelope $\mathbb{E}$ by using secrecy closure rules listed in Figures 6, 7 and 8 until no more rules are applicable. Due to non-determinism in applying the BCQ-rules, different executions of Λ_S may result different envelopes. Since $\mathcal{A}^*$ is finite, the computation of Λ_S terminates. Let $\mathbb{E}$ be the output of Λ_S. By the assumption that $S_\Sigma \subseteq \mathcal{A}^*$, and by the BCQ$_h$- and BCQ$_c$-rules, it is easy to see that $\mathbb{E} \subseteq \mathcal{A}^*$.

Example 4. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$ be a DL-Lite$_\mathcal{R}$ KB, where $\mathcal{A} = \{A(a), B(a), E(a), \neg F(a)\}$ and $\mathcal{T} = \{A \sqsubseteq D,\ A \sqsubseteq \neg C,\ A \sqsubseteq \exists P,\ B \sqsubseteq \exists P,\ \exists P^- \sqsubseteq \neg C,\ \exists P^- \sqsubseteq \neg F,\ P \sqsubseteq Q\}$. Also let $\mathbb{S} = \{D(a), \exists y_1, y_2[A(y_1) \wedge P(y_1,y_2)], \exists y_1, y_2[P(y_1,y_2) \wedge C(y_2)]\}$ be the secrecy set. Using the assertion expansion rules in Figures 1, 2 and 3, we get $\mathcal{A}^* = \{A(a), B(a), \neg C(a), \neg C(b), \neg C(c), D(a), E(a), \neg F(a),$*

$\sqsubseteq^{\leftarrow}_{NL\neg}$ – rule : if $\neg A(a) \in \mathbb{E}$, $A \sqsubseteq L \in \mathcal{T}$ and
$\neg L(a) \in \mathcal{A}^* \setminus \mathbb{E}$,
then $\mathbb{E} := \mathbb{E} \cup \{\neg L(a)\}$;

$\sqsubseteq^{\leftarrow}_{N\exists\neg}$ – rule : if $\neg A(a) \in \mathbb{E}$, $A \sqsubseteq \exists R \in \mathcal{T}$ and
$\forall b \in \mathcal{O}^*$, $\neg\underline{inv}(R,a,b) \in \mathcal{A}^* \setminus \mathbb{E}$,
then pick a $c \in \mathcal{O}^*$ such that
$\mathbb{E} := \mathbb{E} \cup \{\neg\underline{inv}(R,a,c)\}$;

$\sqsubseteq^{\leftarrow}_{\exists L\neg}$ – rule : if $\neg\underline{inv}(R,a,b) \in \mathbb{E}$, $\exists R \sqsubseteq L \in \mathcal{T}$
and $\neg L(a) \in \mathcal{A}^* \setminus \mathbb{E}$,
then $\mathbb{E} := \mathbb{E} \cup \{\neg L(a)\}$;

$\sqsubseteq^{\leftarrow}_{\exists\exists\neg}$ – rule : if $\neg\underline{inv}(R,a,b) \in \mathbb{E}$, $\exists R \sqsubseteq \exists S \in \mathcal{T}$
and $\forall c \in \mathcal{O}^*, \neg\underline{inv}(S,a,c)$
$\in \mathcal{A}^* \setminus \mathbb{E}$, then pick a $d \in \mathcal{O}^*$
such that $\mathbb{E} := \mathbb{E} \cup \{\neg\underline{inv}(S,a,d)\}$;

$\sqsubseteq^{\leftarrow}_{RE\neg}$ – rule : if $\neg\underline{inv}(R,a,b) \in \mathbb{E}$, $R \sqsubseteq E \in \mathcal{T}$
and $\neg\underline{neg}(E,a,b) \in \mathcal{A}^* \setminus \mathbb{E}$,
then $\mathbb{E} := \mathbb{E} \cup \{\neg\underline{neg}(E,a,b)\}$;

$\sqsubseteq^{\leftarrow}_{N\not\exists\neg}$ – rule : if $\neg A(a) \in \mathbb{E}$, $A \sqsubseteq \neg\exists R \in \mathcal{T}$ and
$\underline{inv}(R,a,c) \in \mathcal{A}^* \setminus \mathbb{E}$, for some $c \in \mathcal{O}^*$,
then $\mathbb{E} := \mathbb{E} \cup \{\underline{inv}(R,a,b)\}$;

$\sqsubseteq^{\leftarrow}_{\exists\not\exists\neg}$ – rule : if $\neg\underline{inv}(R,a,b) \in \mathbb{E}$, $\exists R \sqsubseteq \neg\exists S \in \mathcal{T}$
and $\underline{inv}(S,a,c) \in \mathcal{A}^* \setminus \mathbb{E}$,
for some $c \in \mathcal{O}^*$,
then $\mathbb{E} := \mathbb{E} \cup \{\underline{inv}(S,a,c)\}$

Figure 7: Secrecy closure rules obtained by inverting rules in Figure 3.

$\neg F(b), \neg F(c), P(a,b), P(a,c), Q(a,b), Q(a,c)\}$. Using the secrecy closure rules in Figures 6, 7 and 8, we get $\mathbb{E} = \{A(a), D(a), \neg C(b)\}$. Then graphs for $\mathcal{A}^*$ and $\mathcal{A}^* \setminus \mathbb{E}$ are listed in Figure 9.

The following results show that no assertion in the envelope $\mathbb{E}$ is "logically reachable" from outside the envelope. The proof of the next Lemma is standard and is omitted.

Lemma 4. *Let $\mathcal{A}^*$ be a completed ABox obtained from Σ by first applying the rules in Figure 1, then in Figure 2 and then rules in Figure 3 as specified in Section 3. Also, let $\mathbb{E}$ be a set of assertions which, starting from S_Σ, is completed by first using rules in Figures 6 and 7, and then rules in Figure 8. Then, the ABox $\mathcal{A}^* \setminus \mathbb{E}$ is completed.*

The following corollary states, roughly, that the secret BCQs are not logically reachable from $\mathcal{A}^* \setminus \mathbb{E}$.

Corollary 2. *Let $\mathbb{E}'$ be any subset of $\mathcal{A}^*$ which is completed with respect to secrecy closure rules listed in*

BCQ_h – rule : if $q \in S_{CQ}$, and there is a labeled homomorphism $h : V[q] \xrightarrow[c]{} V[\mathcal{A}^* \setminus \mathbb{E}]$ such that $\{A_1(h(\zeta_1)),.., A_k(h(\zeta_k)), P_1(h(\eta_1),h(\mu_1)),.., P_m(h(\eta_m),h(\mu_m))\} \cap \mathbb{E} = \varnothing$ then $\mathbb{E} := \mathbb{E} \cup \{A_p(h(\zeta_p))\}$ for some $1 \le p \le k$ or $\mathbb{E} := \mathbb{E} \cup \{P_r(h(\eta_r),h(\mu_r))\}$ for some $1 \le r \le m$;

BCQ_c – rule : if $q \in S_{CQ}$, and every $f : V[q] \xrightarrow[c]{} V[\mathcal{A}^* \setminus \mathbb{E}]$ is clashy, then pick one such clashy mapping g. Then,

- $\forall p, 1 \le p \le k$, if $\neg A_p(g(\zeta_p)) \in \mathcal{A}^* \setminus \mathbb{E}$ then $\mathbb{E} := \mathbb{E} \cup \{\neg A_p(g(\zeta_p))\}$, and
- $\forall r, 1 \le r \le m$, if $\neg P_r(g(\eta_r), g(\mu_r)) \in \mathcal{A}^* \setminus \mathbb{E}$ then $\mathbb{E} := \mathbb{E} \cup \{\neg P_r(g(\eta_r), g(\mu_r))\}$.

Figure 8: Secrecy closure rules for $q \in S_{CQ}$: $q = \exists y_1,.,y_n[A_1(\zeta_1) \wedge ... \wedge A_k(\zeta_k) \wedge P_1(\eta_1,\mu_1) \wedge .... \wedge P_m(\eta_m,\mu_m)]$.

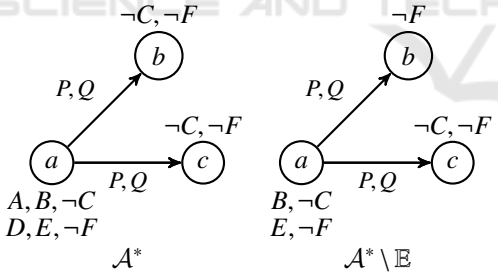

Figure 9: The graphs of $\mathcal{A}^*$ and $\mathcal{A}^* \setminus \mathbb{E}$.

Figure 8. Then, for every $q \in S_{CQ}$,

- *there is no labeled graph homomorphism $h : V[q] \xrightarrow[c]{} V[\mathcal{A}^* \setminus \mathbb{E}']$, and*
- *there exists at least one mapping $f : V[q] \xrightarrow[c]{} V[\mathcal{A}^* \setminus \mathbb{E}']$ which is not clashy.*

Proof. Let $\mathbb{E}'$ be completed with respect to secrecy closure rules listed in Figure 8. This implies that for every $q \in S_{CQ}$, no BCQ_h-rule is applicable to q. Hence, by the conditions of BCQ_h-rule, there is no labeled graph homomorphism $h : V[q] \xrightarrow[c]{} V[\mathcal{A}^* \setminus \mathbb{E}']$, for any $q \in S_{CQ}$. Similarly, no BCQ_c-rule is applicable to q. It follows that for each $q \in S_{CQ}$, there exist at least one mapping $f : V[q] \xrightarrow[c]{} V[\mathcal{A}^* \setminus \mathbb{E}']$ which is not clashy. □

Finally, we show that the completed set $\mathbb{E}$ (an output of Λ_S), is in fact an envelope.

Theorem 5. *$\mathbb{E}$ is an envelope for $\mathbb{S}$.*

Proof. We must show that the set $\mathbb{E}$ satisfies the properties of Definition 4. Clearly, $S_\Sigma \subseteq \mathbb{E}$. First we show that, for every $\alpha \in \mathbb{E}$, $\mathcal{A}^* \setminus \mathbb{E} \not\models^* \alpha$. Suppose $\mathcal{A}^* \setminus \mathbb{E} \models^* \alpha$, for some $\alpha \in \mathbb{E}$. By Theorem 2, we have $\alpha \in (\mathcal{A}^* \setminus \mathbb{E})^*$ and by Lemma 4, $\alpha \in \mathcal{A}^* \setminus \mathbb{E}$, a contradiction.

Next we show that for each $q \in S_{CQ}$, $\mathcal{A}^* \setminus \mathbb{E} \not\models^* q$ and $\mathcal{A}^* \setminus \mathbb{E} \mid\not\models^* q$.

- Assume $\mathcal{A}^* \setminus \mathbb{E} \models^* q$. Then, for every OW-model $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ of $(\mathcal{A}^* \setminus \mathbb{E}, \mathcal{T})$, $\mathcal{I}^* \models q$ where $\mathcal{I}^* = (\Delta^* = \mathcal{I}^*(\mathcal{O}^*), \cdot^{\mathcal{I}^*})$, see Section 3.1. Let $\mathcal{J}$ be the canonical model of $\mathcal{A}^* \setminus \mathbb{E}$. Then, $\mathcal{J}^* = \mathcal{J}$, and $\pi_{\mathcal{J}} : V[q] \xrightarrow[c]{} \Delta^* = \mathcal{J}(\mathcal{O}^*) = \mathcal{O}^*$ and $(\mathcal{J}, \pi_{\mathcal{J}}) \models \beta$, for every $\beta \in Atoms(q)$.
  Now, let $v \in V[q]$ and $A(v) \in Atoms(q)$. Then, $(\mathcal{J}, \pi_{\mathcal{J}}) \models A(v) \Rightarrow \pi_{\mathcal{J}}(v) \in A_Y^{\mathcal{J}} \Rightarrow A(\pi_{\mathcal{J}}(v)) \in \mathcal{A}^* \setminus \mathbb{E} \Rightarrow A \in L[\mathcal{A}^* \setminus \mathbb{E}](\pi_{\mathcal{J}}(v))$. Similarly, let $u, v \in V[q]$ and $P(u,v) \in Atoms(q)$. Then, $(\mathcal{J}, \pi_{\mathcal{J}}) \models P(u,v) \Rightarrow (\pi_{\mathcal{J}}(u), \pi_{\mathcal{J}}(v)) \in P_Y^{\mathcal{J}} \Rightarrow P((\pi_{\mathcal{J}}(u), \pi_{\mathcal{J}}(v))) \in \mathcal{A}^* \setminus \mathbb{E} \Rightarrow P \in L[\mathcal{A}^* \setminus \mathbb{E}]((\pi_{\mathcal{J}}(u), \pi_{\mathcal{J}}(v)))$. It follows that, $\pi_{\mathcal{J}} : V[q] \xrightarrow[c]{} V[\mathcal{A}^* \setminus \mathbb{E}]$ is a labeled graph homomorphism contradicting Corollary 2.
- Assume $\mathcal{A}^* \setminus \mathbb{E} \mid\models^* q$. Then, for every OW-model $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ of $(\mathcal{A}^* \setminus \mathbb{E}, \mathcal{T})$, $\mathcal{I}^* \mid\models q$ where $\mathcal{I}^* = (\Delta^* = \mathcal{I}^*(\mathcal{O}^*), \cdot^{\mathcal{I}^*})$. Let $\mathcal{J}$ be the canonical model of $\mathcal{A}^* \setminus \mathbb{E}$. Then, $\mathcal{J}^* = \mathcal{J}$ and for each valuation $\pi : V[q] \xrightarrow[c]{} \Delta^* = \mathcal{J}(\mathcal{O}^*) = \mathcal{O}^*$, $(\mathcal{J}, \pi) \models \neg\beta$, for some $\beta \in Atoms(q)$. Let k be any such valuation. Then, $(\mathcal{J}, k) \models \neg A(v)$ for some $A(v) \in Atoms(q)$ or $(\mathcal{J}, k) \models \neg P(u,v)$ for some $P(u,v) \in Atoms(q)$. In the first case, $k(v) \in A_N^{\mathcal{J}} \Rightarrow \neg A(k(v)) \in \mathcal{A}^* \setminus \mathbb{E} \Rightarrow \neg A \in L[\mathcal{A}^* \setminus \mathbb{E}](k(v))$ and in the second case, $(k(u), k(v)) \in P_N^{\mathcal{J}} \Rightarrow \neg P((k(u), k(v))) \in \mathcal{A}^* \setminus \mathbb{E} \Rightarrow \neg P \in L[\mathcal{A}^* \setminus \mathbb{E}]((k(u), k(v)))$. Hence, $k : V[q] \xrightarrow[c]{} V[\mathcal{A}^* \setminus \mathbb{E}]$ is clashy. Since k was arbitrary, it follows that all valuations are clashy. However, $\mathbb{E}$ is completed, so by Corollary 2 there exist at least one mapping $k : V[q] \xrightarrow[c]{} V[\mathcal{A}^* \setminus \mathbb{E}]$ which is not clashy. This is a contradiction. Hence, $\mathcal{A}^* \setminus \mathbb{E} \mid\not\models^* q$.

□

Ideally, we would like to compute a minimum envelope $\mathbb{E}$ which makes query answering as informative as possible without compromising secrecy. However, computing minimum envelope appears to be hard, see (Tao et al., 2015) where the authors proved that computing minimum size envelopes is *NP-hard* even for propositional Horn KBs. So, our focus now is to compute a minimal envelope with the property that removing any one of the assertions in $\mathbb{E}$ would reveal some of the secrets. We call such an envelope a *tight envelope*. Formally,

Definition 5. *An envelope $\mathbb{E}$ is said to be tight if for every $\alpha \in \mathbb{E}$, $\mathbb{E} \setminus \{\alpha\}$ is not an envelope.*

Next, we observe that an envelope computed using the rules in Figures 6, 7 and 8 need not be tight.

Example 5. *Consider a DL-Lite$_\mathcal{R}$ KB, where $\mathcal{A} = \{W(a,b), W(a,c)\}$ and $\mathcal{T} = \{\exists W \sqsubseteq A, \exists W^- \sqsubseteq B\}$. Let $\mathbb{S} = \{\exists y,z[A(y) \wedge W(y,z) \wedge B(z)]\}$ be the secrecy set. Using the rules in Figure 1, we compute $\mathcal{A}^* = \{A(a), B(b), B(c), W(a,b), W(a,c)\}$. Since Λ_S is a non-deterministic algorithm, Λ_S may output different envelopes. For illustration purposes, we considered two envelopes namely $\mathbb{E}_1 = \{A(a), W(a,b), W(a,c)\}$ and $\mathbb{E}_2 = \{W(a,b),$
$W(a,c)\}$. It is easy to see that $\mathbb{E}_2$ is tight, whereas $\mathbb{E}_1$ is not.*

A simple naive approach to compute a tight envelope could work as follows. Given a precomputed $\mathcal{A}^*$ and a secrecy set $\mathbb{S} = S_\Sigma \cup S_{CQ}$, we can compute an envelope $\mathbb{E}$ of $\mathbb{S}$ as explained in the beginning of this section. An assertion $\alpha \in \mathbb{E} \setminus \mathbb{S}$ is said to be *redundant* if $\mathbb{E} \setminus \{\alpha\}$ is an envelope, i.e., $((\mathcal{A}^* \setminus \mathbb{E}) \cup \{\alpha\})^* \cap (\mathbb{E} \setminus \{\alpha\}) = \emptyset$. To compute a tight envelope, for each $\beta \in \mathbb{E} \setminus \mathbb{S}$ we check whether β is redundant in which case it is moved from $\mathbb{E}$ to $\mathcal{A}^* \setminus \mathbb{E}$. Otherwise, β remains in $\mathbb{E}$.

5.2 Query Answering

At this point all the necessary computations have been done just once, and we are ready to answer queries while maintaining secrecy. Thus, we assume that $\mathcal{A}^*$ and $\mathbb{E}$ have been precomputed. From an algorithmic point of view, answering queries may be based on checking membership in the set $\mathcal{A}^* \setminus \mathbb{E}$ or searching for specific graph substructures in the graph $G[\mathcal{A}^* \setminus \mathbb{E}]$. Suppose that the agent poses query q of the form $C(a)$ or $E(a,b)$. Then, the reasoner checks for the membership of q and $\neg q$ in the set $\mathcal{A}^* \setminus \mathbb{E}$. If $q \in \mathcal{A}^* \setminus \mathbb{E}$, then the reasoner should answer "Yes". If $\neg q \in \mathcal{A}^* \setminus \mathbb{E}$, then the reasoner should answer "No". If neither q nor $\neg q$ is in $\mathcal{A}^* \setminus \mathbb{E}$, then the reasoner should answer "Unknown". Since assertional queries do not involve fresh individuals, in this case, the answer can be computed in polynomial time (in the size of the original KB Σ) .

Now suppose that the agent poses a BCQ q. Then, the reasoner considers the mappings $V[q] \underset{c}{\rightarrow} V[\mathcal{A}^* \setminus \mathbb{E}]$. If there exists a labeled homomorphism $h : V[q] \underset{c}{\rightarrow} V[\mathcal{A}^* \setminus \mathbb{E}]$, then the reasoner should answer "Yes" by Theorem 3. It follows that the problem of deciding whether answer to a BCQ is "Yes", is *NP-Complete*. If every such mapping is clashy, then the reasoner should answer "No", see Theorem 4. Therefore, the problem of deciding whether answer to a BCQ is "No", is *coNP*. Finally, we should answer "Unknown" precisely when (a) there is no homomorphism $V[q] \underset{c}{\rightarrow} V[\mathcal{A}^* \setminus \mathbb{E}]$ and (b) not every such mapping is clashy. It follows that the problem of deciding whether answer to a BCQ is "Unknown" lies in $DP = \{L \mid L = L_1 \cap L_2$ with $L_1 \in NP$ and $L_2 \in coNP\}$, see (Papadimitriou, 2003).

Example 6. *We use the KB, the secrecy set $\mathbb{S}$ and the envelope $\mathbb{E}$ considered in Example 4. Answers for the BCQs q_1, q_2 and q_3 whose query graphs are given below, are computed in the following based on $\mathcal{A}^* \setminus \mathbb{E}$.*

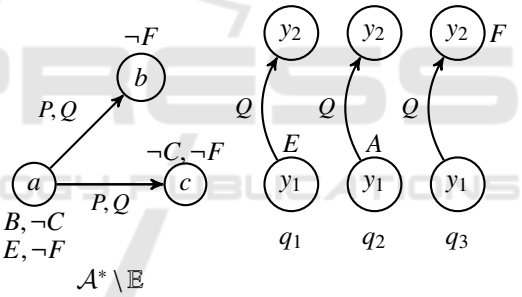

Figure 10: The graphs of $\mathcal{A}^* \setminus \mathbb{E}$ and queries.

First let us consider the BCQ $q_1 = \exists y_1, y_2[E(y_1) \wedge Q(y_1, y_2)]$. Since there exists a homomorphism from $G[q_1]$ to $G[\mathcal{A}^* \setminus \mathbb{E}]$, namely, $y_1 \mapsto a$, $y_2 \mapsto b$ and since $L[q_1](y_1) \subseteq L[\mathcal{A}^* \setminus \mathbb{E}](a)$, $L[q_1](y_1, y_2) \subseteq L[\mathcal{A}^* \setminus \mathbb{E}](a,b)$, $L[q_1](y_2) \subseteq L[\mathcal{A}^* \setminus \mathbb{E}](b)$, the answer to q_1 is "Yes". Actually, there are two labeled homomorphisms from $G[q_1]$ to $G[\mathcal{A}^* \setminus \mathbb{E}]$, the other one being, $y_1 \mapsto a$, $y_2 \mapsto c$.

Next, $q_2 = \exists y_1, y_2[A(y_1) \wedge Q(y_1, y_2)]$. Since there is no labeled homomorphism and there exist non-clashy mappings from $G[q_2]$ to $G[\mathcal{A}^* \setminus \mathbb{E}]$, e.g., $y_1 \mapsto a$, $y_2 \mapsto b$, answer to q_2 is "Unknown".

Finally, consider the BCQ $q_3 = \exists y_1, y_2[Q(y_1, y_2) \wedge F(y_2)]$. It is easy to see that all the mappings from $G[q_3]$ to $G[\mathcal{A}^* \setminus \mathbb{E}]$ are clashy. Hence, answer for the BCQ q_3 is "No".

6 CONCLUSIONS

In this paper we have studied the problem of secrecy-preserving query answering over acyclic *DL-Lite*$_{\mathcal{R}}$ KBs. We have extended the conceptual logic-based framework for secrecy-preserving reasoning which was introduced by Tao et al., see (Tao et al., 2015), so as to allow BCQs. As the OWA underlies the foundational aspects of KBs, to show that the reasoner is sound and complete we used the semantics based on Kleene's 3-valued logic, see (Avron, 1991; Tao et al., 2015). We provide syntactic characterizations for entailment and disentailment of BCQs in terms of properties of mappings (Section 4).

ACKNOWLEDGMENTS

This research work was done by the first author while he was a graduate student of Department of Computer Science, Iowa State University. This work was supported by the NSF grant CNS1116050. Any opinion, finding, and conclusions contained in this article are those of authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Artale, A., Calvanese, D., Kontchakov, R., and Zakharyaschev, M. (2009). The dl-lite family and relations. *J. of Artificial Intelligence Research*, 36:1–69.

Avron, A. (1991). Natural 3-valued logics–characterization and proof theory. *The Journal of Symbolic Logic*, 56(01):276–294.

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

Bell, D. and LaPadula, L. (1973). Secure computer systems: Mathematical foundations. Technical report, DTIC Document.

Biskup, J. and Tadros, C. (2012). Revising belief witwith revealing secrets. In *Lukasiewiez, T, and Sali, A. (eds) FoLKS 2012*, volume 7153 of *LNCS*, pages 51–70. Springer.

Biskup, J., Tadros, C., and Wiese, L. (2010). Towards controlled query evaluation for incomplete first-order databases. In *Link, S. (ed) FoLKS 2010*, volume 5956 of *LNCS*, pages 230–247. Springer.

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

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., and Rosati, R. (2007). Tractable reasoning and efficient query answering in description logics: The dl-lite family. *J. of Automated Reasoning*, 39(3):385–429.

Cuenca Grau, B., Kharlamov, E., Kostylev, E., and Zheleznyakov, D. (2013). Controlled query evaluation over owl 2 rl ontologies. In *The Semantic Web–ISWC 2013*, pages 49–65. Springer.

Denning, D. and Denning, P. (1979). Data security. *ACM Computing Surveys (CSUR)*, 11(3):227–249.

Halpern, J. and O'Neill, K. (2008). Secrecy in multiagent systems. *ACM Transactions on Information and System Security (TISSEC)*, 12(1):5.

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

Krishnasamy Sivaprakasam, G. and Slutzki, G. (2016). Secrecy-preserving query answering in $\mathcal{ELH}$ knowledge bases. In *Proceedings of 8th International Conference on Agents and Artificial Intelligence, Rome, Italy*.

Krotzsch, M. (2012). Owl 2 profiles: An introduction to lightweight ontology languages. In *Eiter, T; and Krennwallner, T. (eds) Reasoning Web Summer School Proceedings of Proc. of 8th Int. Summer school*, volume 7487 of *LNCS*, pages 112–183. Springer.

Lutz, C., Toman, D., and Wolter, F. (2008). Conjunctive query answering in $\mathcal{EL}$ using a database system. In *Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2008)*.

Lutz, C., Toman, D., and Wolter, F. (2009). Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *IJCAI*, volume 9, pages 2070–2075.

Mei, J., Liu, S., Xie, G., Kalyanpur, A., Fokoue, A., Ni, Y., Li, H., and Pan, Y. (2009). A practical approach for scalable conjunctive query answering on acyclic$\mathcal{EL}^+$ knowledge base. In *The Semantic Web-ISWC 2009*, pages 408–423. Springer.

Ortiz, M. and Simkus, M. (2012). Reasoning and query answering in description logics. In *Eiter, T; and Krennwallner, T. (eds) Reasoning Web Summer School Proceedings of Proc. of 8th Int. Summer school*, volume 7487 of *LNCS*, pages 1–53. Springer.

Papadimitriou, C. H. (2003). *Computational complexity*. John Wiley and Sons Ltd.

Sicherman, G., De Jonge, W., and Van de Riet, R. (1983). Answering queries without revealing secrets. *ACM Transactions on Database Systems (TODS)*, 8(1):41–59.

Sivaprakasam, G. K. (2016). *Secrecy-preserving reasoning in simple description logic knowledge bases*. PhD thesis, Iowa State University.

Stouppa, P. and Studer, T. (2009). Data privacy for $\mathcal{ALC}$ knowledge bases. In *Logical Foundations of Computer Science*, pages 409–421. Springer.

Tao, J., Slutzki, G., and Honavar, V. (2010). Secrecy-preserving query answering for instance-checking in $\mathcal{EL}$. In *Hitzler, P. and Lukasiwiez, T. (eds) RR 2010*, volume 6333 of *LNCS*, pages 195–203. Springer.

Tao, J., Slutzki, G., and Honavar, V. (2015). A conceptual framework for secrecy-preserving reasoning in knowledge bases. *ACM Transactions on Computational Logic (TOCL)*, 16(1):3.

Weitzner, D., Abelson, H., Berners-Lee, T., Feigenbaum, J., Hendler, J., and Sussman, G. (2008). Information accountability. *Commun. ACM*, 51, 6.