

Query Answering in the Description Logic Horn-*SHIQ**

Thomas Eiter¹, Georg Gottlob², Magdalena Ortiz¹, and Mantas Šimkus¹

¹ Institute of Information Systems, TU Vienna, Austria
{eiter,ortiz,simkus}@kr.tuwien.ac.at

² Computing Laboratory, Oxford University, United Kingdom
georg.gottlob@comlab.ox.ac.uk

Abstract. We provide an EXPTIME algorithm for answering conjunctive queries (CQs) in Horn-*SHIQ*, a Horn fragment of the well-known Description Logic *SHIQ* underlying the OWL-Lite standard. The algorithm employs a domino system for model representation, which is constructed via a worst-case optimal tableau algorithm for Horn-*SHIQ*; the queries are answered by reasoning over the domino system. Our algorithm not only shows that CQ answering in Horn-*SHIQ* is not harder than satisfiability testing, but also that it is polynomial in data complexity, making Horn-*SHIQ* an attractive expressive Description Logic.

1 Introduction

Driven by the development of semantically enhanced systems, as in the context of the Semantic Web and of Enterprise Application Integration, query answering in Description Logics (DLs) has emerged as an important topic. A variety of algorithms have been proposed for this problem and, aiming at different applications, aspects like *combined* and *data complexity* have been guiding their development. The former characterizes the cost of query answering in the general case, while the latter in the case when the query and the knowledge base except the factual part are fixed. Data complexity is especially important for applications in which DLs are used to formalize rich data models for data repositories, as in such context the model is static as compared to the data contents, and typical user queries are known. For querying DLs, conjunctive queries (CQs) have been most widely considered and three major settings have been addressed:

- Very expressive DLs for which standard reasoning tasks, like satisfiability testing or instance checking, are intractable both in data and combined complexity. As query answering is at least as hard, the problem is trivially intractable in general. For example, CQs in *SHIQ* have 2EXPTIME-complete combined complexity [13] and CONP-complete data complexity [6].

* This work was partially supported by the Austrian Science Fund (FWF) grant P20840, Wolfgang Pauli Institute (WPI), and the Mexican National Council for Science and Technology (CONACYT) grant 187697.

- Tailored DLs, like DL-Lite [5], which aim at lower complexity at the price of limited expressiveness. In DL-Lite, CQ answering is CONP-complete in combined complexity, but polynomial if the query is fixed, and has very low data complexity (reducible to FOL, thus inside logarithmic space).
- Weak DLs, like \mathcal{EL} [1], for which standard reasoning and CQ answering under data complexity are P-complete, while CQ answering under combined complexity is intractable [17]. Several extensions of \mathcal{EL} sharing this property (e.g., \mathcal{ELH} , \mathcal{ELI}^f , \mathcal{EL}^+ , \mathcal{EL}^{++}) can be found in [17,12,10].

Since CQ answering is intractable under combined complexity already over very simple knowledge bases, fragments of expressive DLs with tractable data complexity are of particular interest. Ideally, such fragments should also allow for CQ answering with combined complexity not higher than of standard reasoning.

In this paper, we identify Horn-*SHIQ* as a DL with this property. It was introduced in [9] as a Horn fragment of *SHIQ*, in which the syntax is restricted in a way that disjunction is not expressible. While standard reasoning in Horn-*SHIQ* is EXPTIME-complete in general [11], it is polynomial if the taxonomy is fixed [9].

Our main contributions and results are briefly summarized as follows:

- We provide an EXPTIME algorithm for answering CQs in Horn-*SHIQ*. The algorithm is based on answering tree-shaped queries over particular trees (that capture a universal model of the KB) finitely represented by domino systems. The latter are relatives of saturated mosaic sets known in other branches of logic, and the recent knot sets [16] and domino sets [18] in DLs.
- For constructing domino systems, we exploit a dedicated tableaux-based algorithm for consistency checking in Horn-*SHIQ*, which is of independent interest. It adapts the standard *SHIQ* tableaux [8] (using, e.g., *anywhere blocking* [15] and a kind of *lazy unfolding* [7]) to terminate in deterministic single exponential time, yielding a representation of a universal model of \mathcal{K} such that each CQ over \mathcal{K} can be answered on it. This may also be exploited to precompile \mathcal{K} into a (query-independent) domino system for on-line query answering.
- Based on our algorithm, we show that CQs in Horn-*SHIQ* have EXPTIME-complete combined complexity and P-complete data complexity. We also present a fragment of Horn-*SHIQ* for which CQs are easier. In Horn-*SHQ*⁻, which forbids inverse roles and existential projection on the left hand side of containment axioms, the combined complexity of CQs is lowered to PSPACE-completeness.

As Horn-*SHIQ* is an expressive fragment of OWL-Lite, our results are relevant for the Semantic Web context. They extend in fact from CQs to the class of positive (existential) queries. Our result on the combined complexity of CQs in Horn-*SHIQ* is of particular interest. Firstly, it reveals another expressive DL for which CQ answering is not harder than standard reasoning (cf. [16,13,14]). Secondly, it suggests that the exponential jump in combined complexity of CQs by adding inverse roles to \mathcal{ALC} , which was found by Lutz [13], relies on their interaction with disjunction. In other words, if disjunction is eliminated, then inverse roles do not make CQ answering harder.

2 Preliminaries

The Description Logics \mathcal{SHIQ} and Horn- \mathcal{SHIQ} . We assume countably infinite sets \mathbf{C} , \mathbf{R} and \mathbf{I} of *concept names*, *role names*, and *individuals* respectively, where \mathbf{C} contains special concept names \top and \perp . *Roles* are expressions R and R^- , where $R \in \mathbf{R}$, and their inverses are $\text{Inv}(R) = R^-$ and $\text{Inv}(R^-) = R$, respectively. For any roles R and S , $R \sqsubseteq S$ is a *role inclusion axiom (RI)*, and $\text{Trans}(R)$ is a *transitivity axiom (TA)*. For any set \mathcal{R} of RIs and TAs, $\sqsubseteq_{\mathcal{R}}^*$ denotes the reflexive transitive closure of $\{(R, S) \mid R \sqsubseteq S \in \mathcal{R} \text{ or } \text{Inv}(R) \sqsubseteq \text{Inv}(S) \in \mathcal{R}\}$; we write $\text{Trans}_{\mathcal{R}}(R')$ if $R' \sqsubseteq_{\mathcal{R}}^* R$ and $R \sqsubseteq_{\mathcal{R}}^* R'$ for some R s.t. $\text{Trans}(R) \in \mathcal{R}$ or $\text{Trans}(R^-) \in \mathcal{R}$. A role R is *simple* w.r.t. \mathcal{R} , if there is no $S \sqsubseteq_{\mathcal{R}}^* R$ with $\text{Trans}_{\mathcal{R}}(S)$.

Concepts are inductively defined as follows: (a) each $A \in \mathbf{C}$ is a concept, and (b) if C, D are concepts, R is a role, and S is a simple role, then $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall R.C$, $\exists R.C$, $\geq n.S.C$ and $\leq n.S.C$, for $n \geq 1$, are concepts.

An expression $C \sqsubseteq D$, where C, D are concepts, is a *general concept inclusion axiom (GCI)*, and expressions $a:A$ and $\langle a, b \rangle : R$, where $A \in \mathbf{C}$, a and b are individuals, and R is a role, are *concept and role assertions*, respectively. A \mathcal{SHIQ} knowledge base (KB) is a tuple $\mathcal{K} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, where the *TBox* \mathcal{T} is a finite set of GCIs, the *RBox* \mathcal{R} is a finite set of RIs and TAs, and the *ABox* \mathcal{A} is a finite nonempty set of assertions. We denote by $\mathbf{C}(\mathcal{K})$, $\mathbf{R}(\mathcal{K})$ and $\mathbf{I}(\mathcal{K})$ the sets of concept names, role names, and individuals occurring in \mathcal{K} .

An *interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ for a KB \mathcal{K} consists of a nonempty *domain* $\Delta^{\mathcal{I}}$ and a *valuation function* $\cdot^{\mathcal{I}}$ that maps each individual $c \in \mathbf{I}(\mathcal{K})$ to an element $c^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each concept name $C \in \mathbf{C}(\mathcal{K})$ to a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, and each role name $R \in \mathbf{R}(\mathcal{K})$ to a subset $R^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, in such a way that $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ and $\perp^{\mathcal{I}} = \emptyset$. The function $\cdot^{\mathcal{I}}$ is extended to all concepts and roles in the standard way (see, e.g., [11]), and satisfaction of \mathcal{K} by \mathcal{I} ($\mathcal{I} \models \mathcal{K}$), i.e. modelhood, is also standard.

The DL Horn- \mathcal{SHIQ} was introduced [9] as a fragment of \mathcal{SHIQ} . The main idea is to restrict the syntax in a way that \sqcup is not expressible, establishing a correspondence to a Horn fragment of first-order logic with equality. Without loss of generality, we focus here on a normal form of Horn- \mathcal{SHIQ} in [11], to which each Horn- \mathcal{SHIQ} KB can be efficiently rewritten while preserving the answers to arbitrary CQs (as follows from [11]).

Definition 1. (Normal) Horn- \mathcal{SHIQ} KBs contain only GCIs of the forms

$$\begin{array}{lll} A \sqcap B \sqsubseteq C & A \sqsubseteq \forall R.B & A \sqsubseteq \geq m.S.B \\ \exists R.A \sqsubseteq B & A \sqsubseteq \exists R.B & A \sqsubseteq \leq 1.S.B \end{array}$$

where A, B, C are concept names, R is a role, S is a simple role, and $m \geq 1$.

Example 1. Assume two Horn- \mathcal{SHIQ} KBs $\mathcal{K}_1 = \langle \mathcal{T}, \emptyset, \mathcal{A} \rangle$ and $\mathcal{K}_2 = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, where $\mathcal{T} = \{A \sqsubseteq \exists R.A, B \sqsubseteq \exists P.C\}$, $\mathcal{R} = \{P \sqsubseteq R\}$, and $\mathcal{A} = \{a : A, a : B\}$. Note that both are consistent, however adding $A \sqsubseteq \forall R.\perp$ to \mathcal{T} makes them inconsistent.

Conjunctive Queries. Let \mathbf{V} be a countably infinite set of variables. A (Boolean) *conjunctive query (CQ, or query)* over a KB \mathcal{K} is a finite set q of atoms of the

form $A(x)$ or $R(x, y)$, where A is a concept name, R is a role and $x, y \in \mathbf{V}$.¹ By $\mathbf{V}(q)$ we denote the variables occurring in the atoms of q . The query graph of q is the directed graph G^q over nodes $\mathbf{V}(q)$ with an edge between nodes x and y iff $R(x, y) \in q$ for some role R . The query q is *tree-shaped* if G^q is a tree.

A *match* for q in an interpretation \mathcal{I} is a mapping $\theta : \mathbf{V}(q) \rightarrow \Delta^{\mathcal{I}}$ s.t. (i) $\theta(x) \in A^{\mathcal{I}}$ for each $A(x) \in q$, and (ii) $\langle \theta(x), \theta(y) \rangle \in R^{\mathcal{I}}$ for each $R(x, y) \in q$. We say that \mathcal{I} satisfies q ($\mathcal{I} \models q$), if q has a match in \mathcal{I} , and that \mathcal{K} entails q ($\mathcal{K} \models q$), if q has a match in each model \mathcal{I} of \mathcal{K} .

Example 2. Assume the queries $t_{q_1} = \{A(x), R(x, y), A(y), R(x, z), C(z)\}$ and $t_{q_2} = \{B(x), R(x, y), A(y), P(x, z), C(z)\}$. As easily seen, $\mathcal{K}_1 \not\models t_{q_1}$ and $\mathcal{K}_1 \models t_{q_2}$, while $\mathcal{K}_2 \models t_{q_1}$ and $\mathcal{K}_2 \models t_{q_2}$. Note that both queries are tree-shaped.

3 Tree Queries over Domino Trees

This section describes an algorithm for answering CQs over trees induced by *domino systems*, which is exploited in the next sections for deciding CQ entailment in Horn-*SHIQ* KBs. A domino system finitely represents a possibly infinite tree-shaped interpretation that be can built by connecting matching *dominoes*.

Definition 2. A domino is a tuple $\langle c, r, c' \rangle$ where c, c' are sets of concepts names and r is a set of roles (w.r.t. an underlying alphabet). A domino system is a tuple $\langle D, \triangleright, \mathcal{R} \rangle$, where D is a set of dominoes, $\triangleright \subseteq D \times D$ is a direct successor relation with $c'_1 = c_2$ whenever $\langle c_1, r_1, c'_1 \rangle \triangleright \langle c_2, r_2, c'_2 \rangle$, and \mathcal{R} is an *RBox*. We also require that for each $\langle c, r, c' \rangle \in D$, the set r is closed under role inclusions in \mathcal{R} , i.e., $R \in r$ and $R \sqsubseteq R' \in \mathcal{R}$ imply $R' \in r$. Furthermore, D contains one designated initial domino of the form $\langle \emptyset, \emptyset, c' \rangle$.

Following the terminology in [14], we define next the tree-shaped interpretation induced by a domino system. Its domain is represented by a prefix-closed set of words; for a word $w = e_1 \cdots e_n$, let $\langle w | e_{n+1} \rangle$ denote the word $e_1 \cdots e_n \cdot e_{n+1}$.

Definition 3. The tree base of a domino system $\mathcal{D} = \langle D, \triangleright, \mathcal{R} \rangle$ is the interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ (w.r.t. the alphabet underlying \mathcal{D}) defined as follows:

1. The domain $\Delta^{\mathcal{I}}$ is the smallest set of words over dominoes such that:
 - if $t \in D$ is the initial domino, then $t \in \Delta^{\mathcal{I}}$;
 - if $t_1 \cdots t_n \in \Delta^{\mathcal{I}}$ and $t_n \triangleright t_{n+1}$, then $t_1 \cdots t_n \cdot t_{n+1} \in \Delta^{\mathcal{I}}$.
2. The valuation function $\cdot^{\mathcal{I}}$ is defined as follows:
 - For each atomic concept A , $A^{\mathcal{I}} = \{ \langle s | t \rangle \in \Delta^{\mathcal{I}} \mid t = \langle c, r, c' \rangle \wedge A \in c' \}$.
 - For each role name R ,

$$R^{\mathcal{I}} = \{ \langle s, \langle s | t \rangle \rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid t = \langle c, r, c' \rangle \wedge R \in r \} \cup \{ \langle \langle s | t \rangle, s \rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid t = \langle c, r, c' \rangle \wedge \text{Inv}(R) \in r \}.$$

¹ W.l.o.g. no individuals occur in q ; we can replace each a in q by a new variable y , add $C_a(y)$ to q and $a : C_a$ to \mathcal{A} , where C_a is a new concept name. Non-Boolean queries (i.e., with answer variables) can be reduced to Boolean queries as usual.

function: checkRoleSucc

input: $\mathcal{D} = \langle D, \triangleright, \mathcal{R} \rangle$; dominoes $t_1 = \langle c_1, r_1, c'_1 \rangle$, $t_3 = \langle c_3, r_3, c'_3 \rangle$ from D ; role set $r \neq \emptyset$;

output: *true* iff t_3 is an r -successor of t_1

if t_1 has no direct successor in \mathcal{D} , **then return** *false*;

Initialize t with some direct successor $\langle c_2, r_2, c'_2 \rangle$ of t_1 ;

$s := r_2$; $i := 0$

repeat

if $t = t_3$ and $r \subseteq s$ **then return** *true*;

if $r \not\subseteq s$ or t has no direct successor in \mathcal{D} , **then return** *false*;

 Reassign t to some direct successor $\langle c_2, r_2, c'_2 \rangle$ of t ;

 Set s' to the smallest set closed under the following rule:

if $S \sqsubseteq_{\mathcal{R}}^* R$, $S \in s$, $S \in r_2$, and $\text{Trans}_{\mathcal{R}}(S)$, **then** $R \in s'$;

$s := s'$

until $i > |D|$; **return** *false*

Fig. 1. Verifying Successor Dominoes

The domino tree $\mathcal{T}_{\mathcal{D}} = \langle \Delta^{\mathcal{T}}, \cdot^{\mathcal{T}} \rangle$ of \mathcal{D} is the interpretation identical to \mathcal{I} except that, for each role R , we have $R^{\mathcal{T}} = R^{\mathcal{I}} \cup \bigcup_{S \sqsubseteq_{\mathcal{R}}^* R \wedge \text{Trans}_{\mathcal{R}}(S)} (S^{\mathcal{T}})^+$.

Query entailment in a domino system is naturally defined via the existence of matches in the represented domino tree. We first provide a procedure to verify the existence of special *ordered* matches for tree-shaped queries, and we then extend the result to all CQs via the standard method of query treeification.

Definition 4. A domino system \mathcal{D} entails a CQ q ($\mathcal{D} \models q$), if there is a match for q in $\mathcal{T}_{\mathcal{D}}$. A match π for a tree-shaped CQ tq in $\mathcal{T}_{\mathcal{D}}$ is *ordered* if, for each $x, y \in \mathbf{V}(tq)$, $\pi(x)$ is a proper prefix of $\pi(y)$ whenever $R(x, y) \in tq$ for some R . We write $\mathcal{D} \models_o tq$, if there is some ordered match for tq in $\mathcal{T}_{\mathcal{D}}$.

Let tq be a fixed tree-shaped query and $\mathcal{D} = \langle D, \triangleright, \mathcal{R} \rangle$ a domino system with tree $\mathcal{T}_{\mathcal{D}}$. To obtain a procedure for deciding $\mathcal{D} \models_o tq$, we provide some necessary and sufficient conditions for the existence of ordered matches that can be verified without building $\mathcal{T}_{\mathcal{D}}$ explicitly. Roughly, we search for an association of dominoes from \mathcal{D} with the variables of tq . As the association must witness an ordered match, the domino t_x associated with variable x must encode the concept names needed to satisfy each unary atom $A(x) \in tq$, while for each role atom $R(x, y) \in tq$, the domino t_x must ‘reach’ the domino t_y via an R -path.

Definition 5. For two dominoes $t_1 = \langle c_1, r_1, c'_1 \rangle$ and $t_3 = \langle c_3, r_3, c'_3 \rangle$ in D and a set of roles $r \neq \emptyset$, we say t_3 is a r -successor of t_1 if one of the following holds:

(a) $t_1 \triangleright t_3$ and $r \subseteq r_3$, or

(b) for some role set r' , D contains an r' -successor t_2 of t_1 such that $t_2 \triangleright t_3$ and for each $R \in r$ there exists $S \in r'$ with $\text{Trans}_{\mathcal{R}}(S)$, $S \sqsubseteq_{\mathcal{R}}^* R$ and $S \in r_3$.

We are ready to define domino associations, which characterize the \models_o relation.

Definition 6. A domino association for tq is a mapping μ that assigns to each $z \in \mathbf{V}(tq)$ a domino $\mu(z) \in D$ in a way such that, for each pair $x, y \in \mathbf{V}(tq)$:

- (a) if $A(x) \in tq$, then $A \in c'$, where $\mu(x) = \langle c, r, c' \rangle$; and
- (b) if $r = \{R \mid R(x, y) \in tq\}$ is not empty, then $\mu(y)$ is an r -successor of $\mu(x)$.

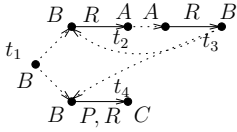


Fig. 2. Domino \mathcal{D}

Example 3. Consider the domino system $\mathcal{D} = \langle D, \triangleright, \mathcal{R} \rangle$ in Figure 2, where the black arrows show the dominoes of D and the dashed arrows the \triangleright relation; $\mathcal{R} = \{Trans(R)\}$ and the initial domino is t_1 . A possible domino association for t_{q_1} in Example 2 is the mapping μ_1 with $\mu_1(x) = t_2$, $\mu_1(y) = t_2$, $\mu_1(z) = t_4$. Note that t_2 and t_4 are $\{R\}$ -successors of t_2 as R is transitive, and that t_{q_1} has no domino association

for $\mathcal{R} = \emptyset$. The query t_{q_2} in Example 2 has a domino association even in this case, witnessed by μ_2 with $\mu_2(x) = t_3$, $\mu_2(y) = t_2$, and $\mu_2(z) = t_4$.

The following is immediate from the definition of \models_o and Definition 6.

Theorem 1. $\mathcal{D} \models_o tq$ iff there exists a domino association for tq .

By Theorem 1, we can decide $\mathcal{D} \models_o tq$ by deciding existence of a domino association. We exploit for the latter the procedure `checkRoleSucc` in Figure 1, which nondeterministically checks whether a domino t_2 is an r -successor of a domino t_1 .

Proposition 1. Let t_1, t_2 be dominoes of \mathcal{D} , and r a role set. Then t_2 is an r -successor of t_1 iff some run of `checkRoleSucc`(\mathcal{D}, t_1, t_2, r) returns true.

Now the following simple procedure `assocDominoes`(\mathcal{D}, tq) non-deterministically decides the existence of a domino association for tq w.r.t. \mathcal{D} : (1) guess a mapping μ from $\mathbf{V}(tq)$ to dominoes of \mathcal{D} , and (2) check satisfaction of the conditions (a) and (b) in Definition 6; to check (b), call `checkRoleSucc` for each arc in tq .

Theorem 2. $\mathcal{D} \models_o tq$ iff some run of `assocDominoes`(\mathcal{D}, tq) returns true.

Having a procedure to decide $\mathcal{D} \models_o tq$ for tree-shaped queries tq , we now settle deciding $\mathcal{D} \models q$ for arbitrary CQs q . Following [4,6,14], we consider query treeifications, i.e., tree-shaped queries whose matches induce matches for q .²

Definition 7. For every CQ q , let $q^{\mathcal{R}}$ be the smallest query such that: (a) $q \subseteq q^{\mathcal{R}}$, (b) $R(x, y) \in q^{\mathcal{R}}$ and $R \sqsubseteq P \in \mathcal{R}$ implies $P(x, y) \in q^{\mathcal{R}}$, (c) $R(x, y) \in q^{\mathcal{R}}$, $R(y, z) \in q^{\mathcal{R}}$ and $Trans_{\mathcal{R}}(R)$ imply $R(x, z) \in q^{\mathcal{R}}$, and (d) $R(x, y) \in q^{\mathcal{R}}$ implies $Inv(R)(y, x) \in q^{\mathcal{R}}$. A treeification of q is a tree-shaped query q' such that $|q'| \leq 2|q|$ and there exists a mapping θ from $\mathbf{V}(q)$ to $\mathbf{V}(q')$ fulfilling

- a) $A(x) \in q$ implies $A(\theta(x)) \in q'$, and
- b) $R(x, y) \in q$ implies $R(\theta(x), \theta(y)) \in (q')^{\mathcal{R}}$.

² Unlike [4,6,14], we do not use treeifications to reduce CQ entailment to concept satisfiability, as this would require the use of role conjunction and the decidability of this extension of Horn-*SHIQ* in EXPTIME is not apparent.

As easily shown, each match for a treeification q' of q in $\mathcal{T}_{\mathcal{D}}$ is also a match for q . On the other hand, from each match for q in $\mathcal{T}_{\mathcal{D}}$, we can obtain a treeification q' that is mappable into $\mathcal{T}_{\mathcal{D}}$ via an ordered match.³ Hence, we obtain:

Theorem 3. *For any CQ q , $\mathcal{D} \models q$ iff $\mathcal{D} \models_o q'$, for some treeification q' of q .*

It follows that we can decide $\mathcal{D} \models q$ by listing all treeifications of q and posing them separately over \mathcal{D} . Note that there are finitely many treeifications of q .

4 A Tableaux Algorithm for Horn-*SHIQ*

In order to exploit the results of Section 3 for query answering in Horn-*SHIQ*, we first provide a tableau algorithm for KB satisfiability. Like the standard *SHIQ* tableau [8], it uses a *completion forest* to represent a model; in the next section we extract from it a domino system that can be used for query answering.

In what follows, $\mathcal{K} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ is a Horn-*SHIQ* KB. We use A, B, C to denote concepts names; D, E to denote (arbitrary) concepts; R, R' to denote a role; S a simple role; and a, b to denote individuals.

Most of the following definitions are based on [8], while 9 and 14 follow [15] and are related to *anywhere blocking*. Definition 8 is simplified since only normalized KBs are considered, and the \approx relation from [8] is omitted in Definition 10.⁴

Definition 8. (*concept closure*) We define $\mathbf{CI}(\mathcal{K})$ as the smallest set of concepts closed under subconcepts such that (i) $D, E \in \mathbf{CI}(\mathcal{K})$ for every $D \sqsubseteq E \in \mathcal{T}$; and (ii) if $\forall R.A \in \mathbf{CI}(\mathcal{K})$, $\text{Trans}_{\mathcal{R}}(R')$ and $R' \sqsubseteq_{\mathcal{R}}^* R$ for some R' , then $\forall R'.A \in \mathbf{CI}(\mathcal{K})$.

Definition 9. (*(named/unnamed) nodes*) We assume a countably infinite set \mathbf{N} of nodes and a strict total order $<$ on \mathbf{N} . Each $a \in \mathbf{I}(\mathcal{K})$ is associated with one fixed node $n_a \in \mathbf{N}$; the nodes n_a are named, all other nodes are unnamed.

Definition 10. (*completion forest*) A completion forest for a KB \mathcal{K} is a tuple $\mathcal{F} = \langle \mathcal{N}, \mathcal{E}, \mathcal{L}, \not\sim \rangle$ where $\mathcal{N} \subseteq \mathbf{N}$ and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ define a directed graph; \mathcal{L} is a labeling function assigning each $n \in \mathcal{N}$ a subset of $\mathbf{CI}(\mathcal{K})$ and each pair $u, u' \in \mathcal{N} \times \mathcal{N}$ to a set of roles (over \mathcal{K}), in such a way that $\mathcal{L}(u, u') = \emptyset$ for all $(u, u') \notin \mathcal{E}$; and $\not\sim \subseteq \mathcal{N} \times \mathcal{N}$ is a binary relation, tacitly assumed to be symmetric.

Definition 11. (*successor, neighbor*) For a completion forest $\mathcal{F} = \langle \mathcal{N}, \mathcal{E}, \mathcal{L}, \not\sim \rangle$ and a pair $u, u' \in \mathcal{N}$, u' is a successor of u if $(u, u') \in \mathcal{E}$. The inverse of successor is called predecessor; the transitive closures of successor and predecessor are ancestor and descendant respectively. For all R , u' is an R -successor of u if $R' \in \mathcal{L}(u, u')$ for some R' with $R' \sqsubseteq_{\mathcal{R}}^* R$. We call u' an R -neighbor of u , if u' is an R -successor of u , or if u is an $\text{Inv}(R)$ -successor of u' .

³ As implicit in [6], such a q' with $|q'| \leq 2|q|$ exists: to obtain a treeification from a match, one replaces each atom $R(x, y) \in q$ with a pair of atoms in case x, y are mapped (i) to the same node, or (ii) to nodes in different branches of the domino tree.

⁴ It is irrelevant for query answering, but could be emulated e.g. using node labels C_a .

Definition 12. (*clash-free completion forest*) A completion forest \mathcal{F} contains a clash, if (i) for some $u \in \mathcal{N}$, $\perp \in \mathcal{L}(u)$; or (ii) for some $u \in \mathcal{N}$ with $\leq n$ $S.C \in \mathcal{L}(u)$, u has $n + 1$ S -neighbors w_0, \dots, w_n such that, for all $0 \leq i < j \leq n$, $C \in \mathcal{L}(w_i)$ and $w_i \not\approx w_j \in \mathcal{F}$. If \mathcal{F} contains no clash, then \mathcal{F} is clash-free.

Definition 13. (*initial completion forest*) The initial completion forest \mathcal{F}_A for \mathcal{K} has the named node n_a labeled with $\mathcal{L}(n_a) = \{A \in \mathbf{C}(\mathcal{K}) \mid a : A \in \mathcal{A}\}$ for each individual $a \in \mathbf{I}(\mathcal{K})$, and an edge $(n_a, n_b) \in \mathcal{E}$ labeled $\mathcal{L}(n_a, n_b) = \{R \mid \langle a, b \rangle : R \in \mathcal{A}\}$ for each pair $a, b \in \mathbf{I}(\mathcal{K})$; the relation $\not\approx$ is empty.

Definition 14. (*blocking*) For a completion forest $\mathcal{F} = \langle \mathcal{N}, \mathcal{E}, \mathcal{L}, \not\approx \rangle$, a node $u \in \mathcal{N}$ is blocked if u is unnamed and u is either directly or indirectly blocked; u is indirectly blocked if one of its ancestors is blocked; u is directly blocked if none of its ancestors is blocked and there is some $u' \leq u$ such that u, u' are unnamed nodes, $\mathcal{L}(u) = \mathcal{L}(u')$, $\mathcal{L}(v) = \mathcal{L}(v')$, and $\mathcal{L}(v, u) = \mathcal{L}(v', u')$, where v and v' are the predecessors of u and u' respectively.

The expansion rules are given in Figure 3, where a node $u \in \mathbf{N}$ is *new* in \mathcal{F} if $u' \leq u$ for every $u' \in \mathcal{N}$. The \leq -rule calls the operation $merge(u, N)$ described in Figure 4. The rules are similar to those in [8], except for the first three, which (lazily) ensure the satisfaction of the TBox axioms. Also, the restricted form of at-most number restrictions allows us to have just one \leq -rule and a deterministic $merge(u, N)$ that simultaneously merges all nodes in N into one.⁵

The initial \mathcal{F}_A is expanded by exhaustively applying the rules in Figure 3. The expansion stops, if a clash is reached; otherwise, it continues until the forest is *complete*, i.e., no rule is applicable. It can be shown similarly as in [8,15] that this algorithm is a decision procedure for KB satisfiability in Horn-*SHIQ*.

Theorem 4. Let \mathcal{K} be a Horn-*SHIQ* KB. Then \mathcal{K} is satisfiable iff a complete and clash-free completion forest for \mathcal{K} can be obtained.

Note that after applying any rule from Figure 3, the resulting forest is uniquely determined up to renaming of nodes. The only source of differences in the resulting forests lies in possibly different orderings of rule applications (this could be eliminated, e.g., using \leq and any fixed ordering on $\mathbf{CI}(\mathcal{K})$ and on the rules). However, these differences are not relevant: each \mathcal{F} represents a *universal* model $\mathcal{I}_{\mathcal{F}}$ (defined as its standard unravelling [8]) that is embeddable into every model of \mathcal{K} , and can be used for query answering. The following is shown by a straightforward induction on the construction of $\mathcal{I}_{\mathcal{F}}$:

Theorem 5. Let \mathcal{I} be a model of \mathcal{K} , let \mathcal{F} be a complete and clash-free completion forest for \mathcal{K} , and let $\mathcal{I}_{\mathcal{F}}$ be the model of \mathcal{K} represented by \mathcal{F} . Then there is a homomorphic embedding of $\mathcal{I}_{\mathcal{F}}$ into \mathcal{I} . Hence, for any CQ q , $\mathcal{K} \models q$ iff $\mathcal{I}_{\mathcal{F}} \models q$.

⁵ Note that the TBox internalization of [8] is not adequate for Horn-*SHIQ*, and that the other rules of [8] are not necessary due to the normal form of the KB.

\mathcal{T} -rule:	if $A \sqsubseteq D \in \mathcal{T}$, $A \in \mathcal{L}(u)$, and $D \notin \mathcal{L}(u)$, then $\mathcal{L}(u) := \mathcal{L}(u) \cup \{D\}$.
$\overline{\mathcal{T}}_{\sqcap}$ -rule:	if $A \sqcap B \sqsubseteq C \in \mathcal{T}$, $\{A, B\} \subseteq \mathcal{L}(u)$, u is not indirectly blocked, and $C \notin \mathcal{L}(u)$, then $\mathcal{L}(u) := \mathcal{L}(u) \cup \{C\}$.
$\overline{\mathcal{T}}_{\exists}$ -rule:	if $\exists R.A \sqsubseteq B \in \mathcal{T}$, $B \notin \mathcal{L}(u)$, u is not indirectly blocked, and u has an R -neighbor u' with $A \in \mathcal{L}(u')$, then $\mathcal{L}(u) := \mathcal{L}(u) \cup \{B\}$.
\exists -rule:	if $\exists R.A \in \mathcal{L}(u)$, u is not blocked, and u has no R -neighbor u' with $A \in \mathcal{L}(u')$, then set $\mathcal{N} = \mathcal{N} \cup \{u'\}$, $\mathcal{E} = \mathcal{E} \cup \{(u, u')\}$, $\mathcal{L}(u, u') := \{R\}$ and $\mathcal{L}(u') := \{A\}$ for some u' new in \mathcal{F} .
\forall -rule:	if $\forall R.A \in \mathcal{L}(u)$, u is not indirectly blocked, and u has an R -neighbour u' with $A \notin \mathcal{L}(u')$, then $\mathcal{L}(u') := \mathcal{L}(u') \cup \{A\}$.
\forall_+ -rule:	if $\forall R.A \in \mathcal{L}(u)$, u is not indirectly blocked, there is some R' with $\text{Trans}_{\mathcal{R}}(R')$ and $R' \sqsubseteq_{\mathcal{R}}^* R$, and there is an R' -neighbour u' of u with $\forall R'.A \notin \mathcal{L}(u')$, then $\mathcal{L}(u') := \mathcal{L}(u') \cup \{\forall R'.A\}$.
\geq -rule:	if $\geq m S.A \in \mathcal{L}(u)$, u is not blocked, and there are no m S -neighbours u_1, \dots, u_m of u such that $A \in \mathcal{L}(u_i)$ and $u_i \not\approx u_j$ for $1 \leq i < j \leq m$, then set $\mathcal{N} = \mathcal{N} \cup \{u_1, \dots, u_m\}$, $\mathcal{E} = \mathcal{E} \cup \{(u, u_1), \dots, (u, u_m)\}$, $\mathcal{L}(u, u_i) := \{S\}$, $\mathcal{L}(u_i) := \{A\}$ and $u_i \not\approx u_j$ for $1 \leq i < j \leq m$ and u_1, \dots, u_m new in \mathcal{F} .
\leq -rule:	if $\leq 1 S.A \in \mathcal{L}(u)$, u is not indirectly blocked, N is the set of all S -neighbours u' of u with $A \in \mathcal{L}(u')$, $ N > 1$ and there is no pair u', u'' in N with $u' \not\approx u''$, then $\text{merge}(u, N)$.

Fig. 3. Tableaux expansion rules

5 Conjunctive Queries over Horn-SHIQ

To answer CQs over Horn-SHIQ KBs, we exploit the method for answering tree-shaped queries over domino systems. For this section, we assume that $\mathcal{K} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ is a consistent Horn-SHIQ KB, and q is an arbitrary CQ.⁶ From a complete and clash-free completion forest $\mathcal{F}_{\mathcal{K}}$ for \mathcal{K} , we extract a domino system $\mathcal{D}_{\mathcal{F}_{\mathcal{K}}}$ that encodes a forest-shaped universal model of \mathcal{K} for query answering. We then rewrite q into a set of tree-shaped queries which can be posed separately over $\mathcal{D}_{\mathcal{F}_{\mathcal{K}}}$, such that $\mathcal{K} \models q$ iff one of the generated queries is entailed by $\mathcal{D}_{\mathcal{F}_{\mathcal{K}}}$.

The transformation of the completion forest into $\mathcal{D}_{\mathcal{F}_{\mathcal{K}}}$, which we now present, eliminates the ‘graph part’ of the forest by encoding it into the initial domino.

Definition 15. *Let $\mathcal{F} = \langle \mathcal{N}, \mathcal{E}, \mathcal{L}, \not\approx \rangle$ be a complete and clash-free completion forest for \mathcal{K} . For every $u \in \mathcal{N}$, let $\mathcal{L}'(u) = \mathcal{L}(u) \cap \mathbf{C}(\mathcal{K})$.*

⁶ Note that in case of inconsistent KBs, query entailment is trivial.

- (1) let u_0 be the \leftarrow -minimal element of N ;
- (2) let $N' = N \setminus \{u_0\}$; let N'' be the minimal set containing N' , each unnamed successor u' of a node in N' , and all descendants of u' ;
- (3) if $(u', n) \in \mathcal{E}$ for some $u' \in N'$ and some named n , then $\mathcal{E} := \mathcal{E} \cup (u_0, n)$ and $\mathcal{L}(u_0, n) := \mathcal{L}(u_0, n) \cup \mathcal{L}(u', n)$;
if $(n, u') \in \mathcal{E}$ for some $u' \in N'$ and some named n , then $\mathcal{E} := \mathcal{E} \cup (n, u_0)$ and $\mathcal{L}(n, u_0) := \mathcal{L}(n, u_0) \cup \mathcal{L}(n, u')$;
- (4) set $\mathcal{N} := \mathcal{N} \setminus N''$, $\mathcal{E} := \mathcal{E} \setminus \{(v, u') \mid u' \in N''\}$, restrict \mathcal{L} and $\not\approx$ to the new \mathcal{N} , \mathcal{E} ;
- (5) add $u_0 \not\approx v$ for every $v \in \mathcal{N}$ such that $u' \not\approx v$ for some $u' \in N'$;
- (6) set $\mathcal{L}(u_0) := \mathcal{L}(u_0) \cup \mathcal{L}(N')$, where $\mathcal{L}(N') = \bigcup_{u' \in N'} \mathcal{L}(u')$;
- (7) if $(u_0, u) \in \mathcal{E}$ then $\mathcal{L}(u_0, u) := \mathcal{L}(u_0, u) \cup \mathcal{L}(N', u)$, else $\mathcal{L}(u, u_0) := \mathcal{L}(u, u_0) \cup \mathcal{L}(u, N')$, where $\mathcal{L}(u, N') = \bigcup_{u' \in N} \mathcal{L}(u, u')$ and $\mathcal{L}(N', u) = \{Inv(R) \mid R \in \mathcal{L}(u, N')\}$.

Fig. 4. The $merge(u, N)$ operation on $\mathcal{F} = \langle \mathcal{N}, \mathcal{E}, \mathcal{L}, \not\approx \rangle$

Let $t_0 = \langle \emptyset, \emptyset, c \rangle$ be the domino where c is the smallest set of fresh concept names such that $Root \in c$ and, for each pair n_a, n_b of named nodes in \mathcal{N} , (i) $A \in \mathcal{L}'(n_a)$ implies $A_a \in c$, (ii) if n_b is an R -neighbour of n_a , then $R_{a,b} \in c$, (iii) $R_{a,b} \in c$ and $R \sqsubseteq R' \in \mathcal{R}$ implies $R'_{a,b} \in c$, (iv) $R_{a,b} \in c$, $R_{b,d} \in c$ and $Trans_{\mathcal{R}}(R)$ implies $R_{a,d} \in c$, and (v) $R_{a,b} \in c$ implies $Inv(R)_{b,a} \in c$.

For each named node $n_a \in \mathcal{N}$, let t_a denote the domino $t_a = \langle c, \{Q_a\}, c' \rangle$, where Q_a is fresh and $c' = \mathcal{L}'(n_a)$. For a pair $(u, u') \in \mathcal{E}$, let $t(u, u')$ denote the domino $\langle \mathcal{L}'(u), \{R \mid u' \text{ is an } R\text{-neighbour of } u\}, \mathcal{L}'(u') \rangle$. Then $\mathcal{D}_{\mathcal{F}} = \langle D, \triangleright, \mathcal{R} \rangle$ is the domino system with initial domino t_0 , where

- D is the smallest domino set containing (i) t_0 , (ii) t_a for each named $n_a \in \mathcal{N}$, and (iii) each $t(u, u')$ such that $(u, u') \in \mathcal{E}$ and u' is unnamed and not blocked.
- \triangleright is the smallest relation s.t. (i) for all named $n_a \in \mathcal{N}$, $t_0 \triangleright t_a$ and $t_a \triangleright t(n_a, u)$ for every $t(n_a, u) \in D$; and (ii) if $t(u, u'), t(u', v) \in D$ for some $(u, u'), (u', u'') \in \mathcal{E}$ such that either $u'' = v$ is not blocked or u'' is blocked by v , then $t(u, u') \triangleright t(u', v)$.

Since the specific complete and clash-free \mathcal{F} does not matter, we assume in what follows a fixed arbitrary $\mathcal{F}_{\mathcal{K}}$ and denote its domino system $\mathcal{D}_{\mathcal{F}_{\mathcal{K}}}$ simply by $\mathcal{D}_{\mathcal{K}}$. As easily seen, we can reconstruct a universal model of \mathcal{K} from the domino tree of $\mathcal{D}_{\mathcal{K}}$. However, for querying $\mathcal{D}_{\mathcal{K}}$, we need to rewrite q in order to handle the links between individuals encoded as concept names in the initial domino.

Definition 16. A link rewriting of q w.r.t. \mathcal{K} is a CQ obtained from q as follows:

1. Exhaustively replace, one by one, $R(y, x)$ by $Inv(R)(x, y)$ whenever there are atoms of the form $R(y, x)$ and $S(x, y)$ in q .
2. Let $\mu: \mathbf{V}(q) \rightarrow \mathbf{I}(\mathcal{K})$ be a partial function, and let $\nu(x) \in \{r \text{ (root)}, i \text{ (inside)}\}$ be a choice for each $x \in \text{dom}(\mu)$. Let $\{z\} \cup \{x' \mid x \in \mathbf{V}(q)\}$ be fresh variables. Then, for each $R(x, y) \in q$ with $\{x, y\} \subseteq \text{dom}(\mu)$, let $S \sqsubseteq_{\mathcal{R}}^* R$ be arbitrary such that $Trans_{\mathcal{R}}(S)$ holds if either $\nu(x) = i$ or $\nu(y) = i$, and (i) replace $R(x, y)$ in q by $Root(z)$, $S_{a,b}(z)$, where $\mu(x) = a$ and $\mu(y) = b$, and (ii) add in q , depending on the choice $[\nu(x), \nu(y)]$, the following atoms:

$$\begin{array}{ll}
 [r, r]: Q_a(z, x), Q_b(z, y); & [i, i]: Q_a(z, x'), Q_b(z, y'), S(x, x'), S(y', y); \\
 [i, r]: Q_a(z, x'), Q_b(z, y), S(x, x'); & [r, i]: Q_a(z, x), Q_b(z, y'), S(y', y).
 \end{array}$$

Roughly speaking, possible R -connections between matches for x and y in $\mathcal{I}_{\mathcal{F}_K}$ via two individuals a, b are reflected in the query by the atoms $Root(z), S_{a,b}(z)$ and $[\nu(x), \nu(y)]$. For example, if $\mathcal{R} = \{Trans(S), S \sqsubseteq R\}$, the CQ $q' = \{Root(z), S_{a,b}(z), Q_a(z, x'), S(x, x'), A(x), Q_b(z, y), B(y)\}$ is a link rewriting of $q = \{A(x), R(x, y), B(y)\}$, obtained by choosing $\mu(x) = a, \mu(y) = b, \nu(x) = i, \nu(y) = r$ and S . A match for q' in $\mathcal{T}_{\mathcal{D}_K}$ corresponds to a match for q in $\mathcal{I}_{\mathcal{F}_K}$ mapping x to a descendant of a (i.e., *inside* the tree rooted at a) and y to b (i.e., the *root* of b 's tree), which are connected via an S -path and thus in the extension of R . Note that, as $\nu(x) = i$ was chosen, the non-transitive R can not link the matches of x and y . Choosing $\mu(x) = a, \mu(y) = b, \nu(x) = r, \nu(y) = r$ and R we obtain a rewriting $q'' = \{Root(z), R_{a,b}(z), Q_a(z, x), A(x), Q_b(z, y), B(y)\}$ that captures the matches for q which map x and y to a and b if they are R -neighbors in $\mathcal{I}_{\mathcal{F}_K}$.

Theorem 6. $\mathcal{K} \models q$ iff $\mathcal{D}_K \models q'$ for some link rewriting q' of q . Hence, due to Theorem 3, $\mathcal{K} \models q$ iff $\mathcal{D}_K \models_o tq$, for some treeification tq of a link rewriting of q .

Theorem 6 suggests a procedure for deciding $\mathcal{K} \models q$: it suffices to verify the existence of a treeification of a link rewriting of q that has an ordered match in the domino tree of \mathcal{D}_K . The latter can be verified using the method from Section 3.

6 Computational Complexity

We now show that CQ entailment in Horn- $SHIQ$ is decidable in exponential time. The presented method relies on the extraction of a domino system from a complete and clash-free completion forest. Hence, the following is important.

Theorem 7. *The tableau algorithm for Horn- $SHIQ$ in Section 4 decides consistency of Horn- $SHIQ$ KBs in single exponential time. For a consistent KB, it constructs a complete and clash-free completion forest of at most exponential size.*

Proof (Sketch). Definition 14 ensures that if a completion forest \mathcal{F} contains two pairs of nodes with the same node-arc-node label combination, one of them is blocked. The number of such combinations, and thus of nodes in a forest, is single exponential in the input KB \mathcal{K} (in fact, it is bounded by $2^{2|C1(\mathcal{K})| \times 2|R(\mathcal{K})|}$). Using the usual arguments [8], it can be shown that the number of rule applications needed to generate \mathcal{F} is polynomially bounded by the maximal number of nodes it can have, as the shrinking rules do not cause repeated rule applications. \square

We are ready to formulate the main complexity results of this paper.

Theorem 8. *Conjunctive query entailment $\mathcal{K} \models q$ in Horn- $SHIQ$ is EXPTIME-complete in combined complexity, i.e., in the size of the KB \mathcal{K} and the query q .*

Proof (Sketch). By Theorem 7, we can check the consistency of \mathcal{K} using the tableau-based algorithm in exponential time. If \mathcal{K} is found inconsistent, then $\mathcal{K} \models q$ trivially holds. Otherwise, we can extract the domino system $\mathcal{D}_{\mathcal{K}}$ from the completion forest $\mathcal{F}_{\mathcal{K}}$ that was constructed, in time polynomial in $|\mathcal{F}_{\mathcal{K}}|$.

Each link rewriting q' of q , as well as each treeification tq of q' , has size polynomial w.r.t. $|\mathcal{K}| + |q|$. There are at most exponentially many q' and, for each q' , at most exponentially many tq ; hence, there are at most exponentially many tq in total, and they can be traversed in polynomial space. To show EXPTIME membership of $\mathcal{K} \models q$, it is thus sufficient to show that $\mathcal{D}_{\mathcal{K}} \models_o tq$ is decidable in exponential time w.r.t. $|\mathcal{K}| + |q|$. Indeed, `checkRoleSucc` runs in NPSpace w.r.t. $|\mathcal{K}| + |q|$ if $\mathcal{D}_{\mathcal{K}}$ is precomputed (note that the counter i needs only polynomial space). The procedure `assocDominoes` runs in NP (w.r.t. $|\mathcal{K}| + |q|$) using `checkRoleSucc` as an oracle. Hence, $\mathcal{D}_{\mathcal{K}} \models_o tq$ is in $\text{NP}^{\text{NPSpace}} = \text{PSPACE}$ w.r.t. $|\mathcal{K}| + |q|$, if $\mathcal{D}_{\mathcal{K}}$ is precomputed. As computing $\mathcal{D}_{\mathcal{K}}$ is feasible in exponential time, it follows that deciding $\mathcal{K} \models q$ is in EXPTIME. The matching lower bound follows from the EXPTIME-hardness of consistency checking in Horn-*SHIQ* [11]. \square

The next result shows that CQs in Horn-*SHIQ* are tractable in data complexity.

Theorem 9. *Conjunctive query entailment $\mathcal{K} \models q$ in Horn-*SHIQ* is P-complete in data complexity, i.e., in the size of the ABox \mathcal{A} of the KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$.*

Proof (Sketch). As in Theorem 8, we can check the consistency of \mathcal{K} when we construct the completion forest $\mathcal{F}_{\mathcal{K}}$. As \mathcal{T} and \mathcal{R} are fixed, $|\mathcal{F}_{\mathcal{K}}|$ is polynomial w.r.t. \mathcal{A} , so $\mathcal{F}_{\mathcal{K}}$ and $\mathcal{D}_{\mathcal{K}}$ can be constructed in time polynomial w.r.t. \mathcal{A} . Next, for fixed q , \mathcal{T} and \mathcal{R} , there are polynomially many treeifications tq of link rewritings q' of q w.r.t. \mathcal{A} , and they can be traversed in polynomial time. By Theorem 1, it remains to show that the existence of a domino association $\mu : \mathbf{V}(tq) \rightarrow D$, where D is the domino set of $\mathcal{D}_{\mathcal{K}}$, is decidable in polynomial time w.r.t. \mathcal{A} . Since $|\mathbf{V}(tq)|$ is bounded by a constant w.r.t. \mathcal{A} and $|D|$ is polynomial w.r.t. \mathcal{A} , there are polynomially many candidate μ w.r.t. \mathcal{A} . We can check r -successorship between dominoes t_1, t_2 of $\mathcal{D}_{\mathcal{K}}$ in time polynomial in $|\mathcal{D}_{\mathcal{K}}|$, i.e., polynomial w.r.t. \mathcal{A} . Hence, we can check whether μ satisfies Definition 6 in polynomial time w.r.t. \mathcal{A} . The resulting P membership bound is tight, as consistency checking in any DL allowing for conjunction on the left hand side and quantified universal restrictions on the right hand side of GCIs is P-hard in data complexity [3]. \square

The source of EXPTIME-hardness of consistency testing in Horn-*SHIQ*, and hence of query entailment, are inverse roles and concepts of the form $\exists R.A$ on the left hand side of the GCIs. Intuitively, both constructs allow to propagate information from a node to its ancestors in a completion forest; any one of them allows for an encoding of a generic Alternating PSPACE Turing machine. If we disallow both, obtaining the DL Horn-*SHQ*⁻, consistency testing and CQ entailment drop to PSPACE-completeness. Roughly, the direct successors of a node in a completion forest can be inferred in polynomial time from its label. Hence, the existence of a complete and clash-free completion forest \mathcal{F} is refutable in PSPACE without building it, by non-deterministically following a path in \mathcal{F} (of

at most exponential length) that leads to a clash. CQ entailment is decidable in PSPACE by supplying `checkRoleSucc` with a PSPACE oracle for navigating the domino system \mathcal{D}_K (note that the explicit construction of \mathcal{D}_K may require exponential space). This procedure is worst-case optimal, as consistency checking in Horn-SHQ^- is PSPACE-hard (provable, e.g., by a generic Turing machine encoding). Finally, also our P-completeness result for data complexity of CQs carries over from Horn-SHIQ to Horn-SHQ^- . Details will be given in the full paper.

7 Related Work and Conclusion

As found recently, answering CQs in some expressive DLs, including \mathcal{ALCH} [16] and \mathcal{ALCHQ} [14], is not harder than standard reasoning, and in fact EXPTIME-complete in combined complexity. Horn-SHIQ is another such DL but orthogonal to those mentioned, as it offers transitive and inverse roles but excludes disjunction (we note that one can infer from [2] the EXPTIME-completeness result for the DL Horn-ALCHI , i.e., Horn-SHIQ without transitive roles and number restrictions). Moreover, the data complexity of CQs is polynomial in Horn-SHIQ but intractable in \mathcal{ALCH} and \mathcal{ALCHQ} (in fact, it is CONP-hard already for \mathcal{AL} [3]).

Different approaches have been recently used to show that CQs have tractable data complexity in some DLs. A large class of such DLs are extensions of \mathcal{EL} [1], considered e.g. in [17,12,10], of which \mathcal{ELH} , \mathcal{ELI}^f , and \mathcal{EL}^{++} are particularly noticeable. For \mathcal{EL} and \mathcal{ELH} , which are subsumed by Horn-SHIQ but not by Horn-SHQ^- (due to the absence of existential restrictions on the LHS of the GCIs), a reduction to Datalog has been given in [17]. In both \mathcal{EL} and \mathcal{ELH} , CQs have CONP-complete combined complexity and P-complete data complexity. For \mathcal{ELI}^f , which is also strictly subsumed by Horn-SHIQ (as the latter offers qualified universal quantification on the RHS of axioms and more general number restrictions) an explicit (partial) construction of a universal model was used in [10]. Like in Horn-SHIQ and Horn-SHQ^- , CQs have P-complete data complexity in \mathcal{ELI}^f . Finally, for \mathcal{EL}^{++} , which is orthogonal to both Horn-SHIQ and Horn-SHQ^- (\mathcal{EL}^{++} has nominals and regular role hierarchies, but lacks universal quantification), special proof-graphs with automata were used in [12]. Noticeably, CQs in \mathcal{EL}^{++} have PSPACE-complete combined and P-complete data complexity respectively, and thus the same complexity as in Horn-SHQ^- .

Another prominent family for which data complexity has been deeply investigated is DL-Lite [5]. For the core DL-Lite and its extension with functionality and conjunction, which is subsumed by Horn-SHIQ but not by Horn-SHQ^- , query rewriting into first-order logic over relational databases has been employed. CQ answering has very low data complexity (inside logarithmic space), and its CONP-complete combined complexity is also much lower than for Horn-SHQ^- .

Our ongoing and future work is devoted to the following issues. The first concerns richer query syntax. As the normal form and the universal model property of Horn-SHIQ carry over to unions of CQs and the more general positive existential queries (PQs), our results can be immediately extended to them. In fact, answering PQs in Horn-SHIQ is reducible to answering at most exponentially

many (in the size of the PQ) CQs. Further, since the universal model property allows us to precompile a knowledge base \mathcal{K} into a (query-independent) domino system $\mathcal{D}_{\mathcal{K}}$ for on-line query answering, the identification of cases in which $\mathcal{D}_{\mathcal{K}}$ is small would be beneficial. Finally, an obvious issue is a detailed study of other fragments of Horn-*SHIQ* besides Horn-*SHQ*⁻. The effect of syntactic restrictions similarly as in [3] on data complexity is here of particular interest.

References

1. Baader, F.: Terminological cycles in a description logic with existential restrictions. In: IJCAI 2003, pp. 325–330 (2003)
2. Cali, A., Gottlob, G., Kifer, M.: Taming the Infinite Chase: Query Answering under Expressive Relational Constraints. In: KR 2008 (to appear, 2008)
3. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Data complexity of query answering in description logics. In: KR 2006 (2006)
4. Calvanese, D., De Giacomo, G., Lenzerini, M.: On the decidability of query containment under constraints. In: PODS 1998, pp. 149–158 (1998)
5. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *DL-lite* family. *J. Autom. Reasoning* 39(3), 385–429 (2007)
6. Glimm, B., Horrocks, I., Lutz, C., Sattler, U.: Conjunctive Query Answering for the Description Logic *SHIQ*. In: IJCAI 2007, pp. 399–404 (2007)
7. Horrocks, I.: Optimising Tableaux Decision Procedures for Description Logics. PhD thesis, University of Manchester (1997)
8. Horrocks, I., Sattler, U.: A tableaux decision procedure for *SHOIQ*. In: IJCAI 2005, pp. 448–453 (2005)
9. Hustadt, U., Motik, B., Sattler, U.: Data complexity of reasoning in very expressive description logics. In: IJCAI 2005, pp. 466–471 (2005)
10. Krisnadhi, A., Lutz, C.: Data complexity in the \mathcal{EL} family of description logics. In: DL 2007, Bressanone, Italy. CEUR-WS, vol. 250, pp. 88–99 (2007)
11. Krötzsch, M., Rudolph, S., Hitzler, P.: Complexity boundaries for Horn description logics. In: AAI 2007, pp. 452–457 (2007)
12. Krötzsch, M., Rudolph, S., Hitzler, P.: Conjunctive queries for a tractable fragment of OWL 1.1. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007/ISWC 2007. LNCS, vol. 4825, pp. 310–323. Springer, Heidelberg (2007)
13. Lutz, C.: Inverse roles make conjunctive queries hard. In: DL 2007 (2007)
14. Lutz, C.: Two upper bounds for conjunctive query answering in *SHIQ*. In: DL 2008 (2008)
15. Motik, B., Shearer, R., Horrocks, I.: Optimized reasoning in description logics using hypertableaux. In: Pfenning, F. (ed.) CADE 2007. LNCS (LNAI), vol. 4603, pp. 67–83. Springer, Heidelberg (2007)
16. Ortiz, M., Šimkus, M., Eiter, T.: Worst-case optimal conjunctive query answering in description logics without inverses. In: AAI 2008 (to appear, 2008)
17. Rosati, R.: On conjunctive query answering in \mathcal{EL} . In: DL 2007 (2007)
18. Rudolph, S., Krötzsch, M., Hitzler, P.: Terminological reasoning in *SHIQ* with ordered binary decision diagrams. In: AAI 2008 (to appear, 2008)