From Horn-$\mathcal{SRIQ}$ to Datalog: A Data-Independent Transformation That Preserves Assertion Entailment

David Carral,[1] **Larry González,**[1] **Patrick Koopmann**[2]
[1]Center for Advancing Electronics Dresden (cfaed), Technische Universität Dresden, Germany
[2]Institute for Theoretical Computer Science, Technische Universität Dresden, Germany
firstname.lastname@tu-dresden.de

Abstract

Ontology-based access to large data-sets has recently gained a lot of attention. To access data efficiently, one approach is to rewrite the ontology into Datalog, and then use powerful Datalog engines to compute implicit entailments. Existing rewriting techniques support Description Logics (DLs) from $\mathcal{ELH}$ to Horn-$\mathcal{SHIQ}$. We go one step further and present one such data-independent rewriting technique for Horn-$\mathcal{SRIQ}_{\sqcap}$, the extension of Horn-$\mathcal{SHIQ}$ that supports role chain axioms, an expressive feature prominently used in many real-world ontologies. We evaluated our rewriting technique on a large known corpus of ontologies. Our experiments show that the resulting rewritings are of moderate size, and that our approach is more efficient than state-of-the-art DL reasoners when reasoning with data-intensive ontologies.

Introduction

Assertion retrieval (AR)—i.e., the task of inferring implicit assertions from a Description Logics (DL) knowledge base (KB)—is an important reasoning task with many applications in knowledge representation and data management. For instance, the computation of AR can be used to solve SPARQL query answering, and to compute statistics on the implicit inferences of data-intensive ontologies such as in (Callahan, Cruz-Toledo, and Dumontier 2013; Vrandečić and Krötzsch 2014). For these tasks, both the concepts an object satisfies and the relations between objects are relevant. Typical DL ontologies focus on providing axioms about concepts, but expressive ontologies also allow to make inferences about roles, e.g., through the use of logical constructors such as inverse roles and role chains.

Efficient AR on large datasets requires the use of "one-pass" algorithms that compute the full set of entailed assertions as part of a saturation procedure. Although many customised algorithms and implementations of this type have been developed in the past, to the best of our knowledge, either these procedures do not support role chains, or they are not complete for deriving role assertions. Indeed, the retrieval of roles in the presence of role chains is a rather challenging task, as it may require reasoning about paths involving objects not explicit in the data.

Example 1. *Let $\mathcal{T}_{ex}$ be the TBox with the following axioms modelling conflicts of interests between researchers.*

$$\mathsf{ResearchGroup} \sqsubseteq \forall \mathsf{hasMember}.\mathsf{Researcher}$$
$$\mathsf{Researcher} \sqsubseteq \exists \mathsf{hasMember}^{-}.\mathsf{ResearchGroup}$$
$$\mathsf{collaborated} \circ \mathsf{hasMember}^{-} \circ \mathsf{hasMember} \sqsubseteq \mathsf{hasConflict}$$
$$\mathsf{hasMember} \circ \mathsf{supervises} \sqsubseteq \mathsf{hasMember}$$

Here, the third axiom uses a role chain to express that, if a researcher collaborated with someone who is a member of a research group, then he has a conflict of interest with everyone from that group. Using $\mathcal{T}_{ex}$, we can infer from the ABox $\mathcal{A}_{ex} = \{\mathsf{collaborated}(\mathsf{gottlob}, \mathsf{alonzo}), \mathsf{supervises}(\mathsf{alonzo}, \mathsf{alan}), \mathsf{Researcher}(\mathsf{alonzo})\}$ the two assertions $\mathsf{Researcher}(\mathsf{alan})$ and $\mathsf{hasConflict}(\mathsf{gottlob}, \mathsf{alan})$. Both entailments depend on the existence of a research group which has both alan *and* alonzo *as members, the existence of which is implied but not explicit. Specifically,* gottlob *has a conflict of interest with* alan *because there is a path via* alonzo *and this research group connecting* gottlob *with* alan*, which corresponds to the role chain in the third axiom.*

We propose a technique for AR from KBs formulated in Horn-$\mathcal{SRIQ}_{\sqcap}$—a DL fragment that supports complex roles and role conjunctions (Krötzsch, Rudolph, and Hitzler 2013)—based on data-independent rewritings into Datalog rule sets. Specifically, given a TBox $\mathcal{T}$, we describe how to construct a Datalog rule set $\mathcal{R}_{\mathcal{T}}$ s.t., for every ABox $\mathcal{A}$ and assertion α only using symbols occurring in $\mathcal{T}$, we have $\langle \mathcal{T}, \mathcal{A} \rangle \models \alpha$ iff $\langle \mathcal{R}_{\mathcal{T}}, \mathcal{A} \rangle \models \alpha$.

To show practical feasibility, we implemented and evaluated our transformation, showing that Datalog rewritings for many real-world Horn-$\mathcal{SRIQ}_{\sqcap}$ TBoxes are of moderate size. Moreover, we computed our Datalog rewritings for two real-world ontologies, and performed AR over the resulting Datalog KBs. Our results show that our approach can outperform *Konclude* (Steigmiller, Liebig, and Glimm 2014)—considered as one of the leading DL reasoners (Parsia et al. 2017)—when solving AR over data-intensive ontologies. This is rather noteworthy, since (unlike Konclude) our rewritings are complete for role retrieval.

In summary, our contributions are as follows.

- We present a worst-case optimal transformation of Horn-

$\mathcal{SRIQ}_\sqcap$ TBoxes into Datalog rule sets that preserves satisfiability and assertion entailment.

- We show that the resulting rule sets can be transformed into equivalent DLP ontologies (Grosof et al. 2003)—the DL fragment underlying the OWL RL standard.
- We empirically show that our rewriting technique produces Datalog rule sets of moderate size for many real-world Horn-$\mathcal{SRIQ}_\sqcap$ TBoxes.
- We empirically show that the resulting Datalog programs can be used to solve AR more efficiently than DL reasoners when dealing with data-intensive ontologies.

Formal proofs and arguments for the results in this paper, as well as evaluation details, are in the extended version of this paper (Carral, González, and Koopmann 2018).

Related Work

Even though there are many algorithms and implementations for AR on DL KBs, we find that none of them can satisfactorily handle role retrieval, i.e., the retrieval of role assertions, in the presence of role chains.

There are many approaches that can efficiently perform AR for DLs which do not support role chains, and which are similar in spirit to our approach. Hustadt et al. (2004) reduce standard reasoning tasks in the DL $\mathcal{SHIQ}^-$ to reasoning over disjunctive query Datalog programs. Eiter et al. (2012) propose a method that combines materialisation—a step that can be repurposed to solve role retrieval—and rewriting to solve conjunctive query answering over Horn-$\mathcal{SHIQ}$ ontologies. A similar method tailored for the DL Horn-$\mathcal{ALCHOIQ}$ is presented by Carral et al. (2018). Recently, Ahmetaj et al. (2016) proposed Datalog rewritings to perform instance queries over $\mathcal{ALCHIO}$ KBs extended with closed predicates.

State-of-the-art DL reasoners such as Fact++ (Tsarkov and Horrocks 2006), HermiT (Motik, Shearer, and Horrocks 2009), Pellet (Sirin et al. 2007) and Konclude (Steigmiller, Liebig, and Glimm 2014) support $\mathcal{SROIQ}$ KBs. However, while the former three do not perform that well on data-intensive ontologies (Parsia et al. 2017), Konclude does not support role retrieval as part of its one-pass algorithm. As our results indicate, Datalog rewritings have the potential to outperform all these approaches.

Regarding less expressive DLs, despite the fact that there are theoretical algorithms for $\mathcal{EL}^{++}$ that can deal with role chains (Krötzsch 2011), leading profile reasoners such as ELK (Kazakov, Krötzsch, and Simančík 2014) do not support this expressive feature yet.

Preliminaries

We consider logical theories based on finite signatures consisting of mutually disjoint sets $\mathsf{N_c}$ of *concepts* (unary predicates), $\mathsf{N_r}$ of *roles* (binary predicates), $\mathsf{N_v}$ of *variables*, and $\mathsf{N_i}$ of *individuals* (constants), as well as an unbounded set $\mathsf{N_0}$ of *nulls* disjoint with all of the above. There is a bijective and irreflexive function $\cdot^- : \mathsf{N_r} \to \mathsf{N_r}$ with $R^{--} = R$ for all $R \in \mathsf{N_r}$, and $\bot, \top \in \mathsf{N_c}$. For a formula or set thereof φ, we use $\mathsf{sig}(\varphi)$ to denote the set of all concepts and roles in φ.

$\bigwedge_{i=1}^{n} A_i(x) \to B(x)$	$\prod_{i=1}^{n} A_i \sqsubseteq B$	$(\sqcap)$
$A(x) \wedge R(x,y) \to B(y)$	$A \sqsubseteq \forall R.B$	$(\forall)$
$A(x) \to \exists y.R(x,y) \wedge B(y)$	$A \sqsubseteq \exists R.B$	$(\exists)$
$A(x) \wedge R(x,y) \wedge B(y)$ $\wedge R(x,z) \wedge B(z) \to y \approx z$	$A \sqsubseteq \leqslant 1\, R.B$	$(\leq)$
$\bigwedge_{i=1}^{n} R_i(x_{i-1}, x_i) \to S(x_0, x_n)$	$R_1 \circ \ldots \circ R_n \sqsubseteq S$	$(\circ)$
$\bigwedge_{i=1}^{m} R_i(x,y) \to S(x,y)$	$\prod_{i=1}^{m} R_i \sqsubseteq S$	$(\sqcap_\mathsf{r})$

Figure 1: Horn-$\mathcal{SRIQ}_\sqcap$ Axioms, where $A_{(i)}, B \in \mathsf{N_c}$, $R_{(i)}, S \in \mathsf{N_r}$, $x_{(i)}, y, z \in \mathsf{N_v}$, $n \geq 1$, and $m > 1$

The sets of *terms* and *ground terms* are $\mathsf{N_t} = 2^{\mathsf{N_i}} \cup \mathsf{N_0} \cup \mathsf{N_v}$ and $\mathsf{N_{gt}} = 2^{\mathsf{N_i}} \cup \mathsf{N_0}$, respectively. The use of $2^{\mathsf{N_i}}$ rather than $\mathsf{N_i}$ in the definition of terms is for convenience of the definition of the chase later in this section. Thus, we henceforth identify every $a \in \mathsf{N_i}$ with the singleton set $\{a\}$.

Existential Rules We write tuples of terms $t_1, \ldots, t_n$ as $\vec{t}$, and treat such tuples as sets when the order is irrelevant. An *atom* is a formula of the form $C(t)$ or $R(t,u)$ with $C \in \mathsf{N_c}$, $R \in \mathsf{N_r}$, and $t, u \in \mathsf{N_t}$. We identify a binary atom $R(t,u)$ with $R^-(u,t)$. A formula or set thereof is *ground* if it only contains ground terms. For a formula φ, we write $\varphi[\vec{x}]$ to indicate that $\vec{x}$ is the set of all free variables occurring in φ.

An *(existential) rule* is a formula of one of the forms:

$$\forall \vec{x}, \vec{z}.\big(B[\vec{x}, \vec{z}] \to \exists \vec{y}.H[\vec{x}, \vec{y}]\big) \qquad (\to)$$

$$\forall \vec{x}.\big(B[\vec{x}] \to x \approx y\big) \qquad (\approx)$$

Where B and H are non-empty, null-free conjunctions of atoms, and $x, y \in \vec{x}$. A *Datalog rule* is a rule without existentially quantified variables. A *fact* is a ground atom. We identify facts and sets thereof if they are identical up to bijective renaming of nulls. A *knowledge base (KB)* is a tuple $\langle \mathcal{R}, \mathcal{A} \rangle$ with $\mathcal{R}$ a rule set and $\mathcal{A}$ an *ABox*—a set of facts without nulls, i.e., *assertions*. We treat KBs as first-order theories and define semantical notions such as entailment and satisfiability in the usual way. To axiomatise the semantics of $\top$, we assume that $\{A(x) \to \top(x) \mid A \in \mathsf{N_c}\} \cup \{R(x,y) \to \top(x) \wedge \top(y) \mid R \in \mathsf{N_r}\} \subseteq \mathcal{R}$ for every rule set $\mathcal{R}$.

The DL Horn-$\mathcal{SRIQ}_\sqcap$. Without loss of generality (Krötzsch, Rudolph, and Hitzler 2013), we define Horn-$\mathcal{SRIQ}_\sqcap$ using a restricted set of normalised axioms, which we introduce in the right hand side of Figure 1. We identify each of these axioms with the corresponding rule in the left hand side of Figure 1, and alternate between these two syntaxes whenever this is convenient.

For an axiom set $\mathcal{R}$, let $\prec^+_\mathcal{R}$ be the minimal transitive relation over roles s.t. $R \prec^+_\mathcal{R} S$ iff $R^- \prec^+_\mathcal{R} S$; for every axiom in $\mathcal{R}$ of Type $(\sqcap_\mathsf{r})$, $R_i \prec^+_\mathcal{R} S$ for all $i \in [\![1, m]\!]$; and, for every axiom in $\mathcal{R}$ of Type $(\circ)$,

- if $n = 1$ and $R_1 \neq S^-$, then $R_1 \prec^+_\mathcal{R} S$, and
- if $n > 1$ and $R_1 \circ \ldots \circ R_n \neq S \circ S$, then

– if $R_n = S$, then $R_i \prec^+_{\mathcal{R}} S$ for all $i \in \{1, \ldots, n-1\}$,
– if $R_1 = S$, then $R_i \prec^+_{\mathcal{R}} S$ for all $i \in \{2, \ldots, n\}$, and
– if $R_1 \neq S \neq R_n$, then $R_i \prec^+_{\mathcal{R}} S$ for all $i \in \{1, \ldots, n\}$.

A role V is *complex* wrt. $\mathcal{R}$ if there is an axiom in $\mathcal{R}$ of Type ($\circ$) with $n > 1$ and $S \prec^*_{\mathcal{R}} V$ with $\prec^*_{\mathcal{R}}$ the reflexive closure of $\prec^+_{\mathcal{R}}$. Otherwise, V is *simple*.

Definition 1. *An axiom set $\mathcal{T}$ is a (Horn-$\mathcal{SRIQ}_\sqcap$) TBox if $\prec^+_{\mathcal{T}}$ is irreflexive, and all roles occurring in an axiom of Type ($\leq$), or in the left hand side of an axiom of Type ($\sqcap_\mathsf{r}$) in $\mathcal{T}$ are simple. A KB $\langle \mathcal{T}, \mathcal{A} \rangle$ is Horn-$\mathcal{SRIQ}_\sqcap$ if $\mathcal{T}$ is a Horn-$\mathcal{SRIQ}_\sqcap$ TBox.*

The Chase A well-known way of characterising entailments from KBs is the chase, which we introduce next.

A *substitution* σ is a partial function over $\mathsf{N_t}$. We use $[t_1/u_1, \ldots, t_n/u_n]$ to denote the substitution σ s.t. $\sigma(t_i) = u_i$ for all $i \in [\![1, n]\!]$. For a formula φ, we write $\varphi\sigma$ to denote the formula obtained by replacing all occurrences of a term t in φ with $\sigma(t)$ if t is in the domain of σ. For a tuple $\vec{t}$ of terms, $\sigma_{\vec{t}} \subseteq \sigma$ is the restriction of σ to the domain $\vec{t}$.

To handle rules of Type ($\approx$), we represent individuals as sets, which is why we used $2^{\mathsf{N_i}}$ in the definition of terms. For a given substitution σ and two variables x, y, we define $\sigma^{\mathsf{rn}}_{x,y}$ by $\sigma^{\mathsf{rn}}_{x,y}(x) = \sigma^{\mathsf{rn}}_{x,y}(y) = \sigma(x)$ if $\sigma(x), \sigma(y) \in \mathsf{N_0}$, and $\sigma^{\mathsf{rn}}_{x,y}(x) = \sigma^{\mathsf{rn}}_{x,y}(y) = (\sigma(x) \cup \sigma(y)) \cap \mathsf{N_i}$ otherwise. Intuitively, $\sigma^{\mathsf{rn}}_{x,y}$ is the substitution identifying $\sigma(x)$ and $\sigma(y)$.

A tuple $\langle \rho, \sigma \rangle$ with $\rho = B[\vec{x}, \vec{z}] \to \exists \vec{y}.H[\vec{x}, \vec{y}]$ a rule and σ a substitution is *applicable* to a set of facts $\mathcal{F}$ if $B\sigma \subseteq \mathcal{F}$, and $H\sigma' \not\subseteq \mathcal{F}$ for all $\sigma' \supseteq \sigma_{\vec{x}}$. The *application* of $\langle \rho, \sigma \rangle$ on $\mathcal{F}$, written $\mathcal{F}\langle \rho, \sigma \rangle$, is the set of facts $\mathcal{F} \cup H\sigma'$ with $\sigma' \supseteq \sigma_{\vec{x}}$ a substitution mapping every variable in $\vec{y}$ to a fresh null. If ρ is of the form $B[\vec{x}] \to x \approx y$, then $\langle \rho, \sigma \rangle$ is applicable to $\mathcal{F}$ if $B\sigma \subseteq \mathcal{F}$ and $\sigma(x) \neq \sigma(y)$. In this case, the application of $\langle \rho, \sigma \rangle$ on $\mathcal{F}$, also denoted by $\mathcal{F}\langle \rho, \sigma \rangle$, is the set $\mathcal{F}\sigma^{\mathsf{rn}}_{x,y}$.

We introduce this non-standard approach of rule applications with equality to ensure that the forest-model property of Horn-$\mathcal{SRIQ}_\sqcap$ ontologies is reflected in the structure of the chase, which will later be useful to show completeness of our Datalog rewritings

Definition 2. *A* chase sequence *for a KB $\mathcal{K} = \langle \mathcal{R}, \mathcal{A} \rangle$ is a sequence $\mathcal{F}^0 = \mathcal{A}, \mathcal{F}^1, \ldots$ of sets of facts s.t.*

- *for all $i \geq 1$, $\mathcal{F}^i = \mathcal{F}^{i-1}\langle \rho, \sigma \rangle$ for a rule $\rho \in \mathcal{R}$ and some substitution σ s.t. $\langle \rho, \sigma \rangle$ is applicable, and*
- *for all $\langle \rho, \sigma \rangle$ with $\rho \in \mathcal{R}$, there is some $k \geq 0$ s.t. $\langle \rho, \sigma \rangle$ is not applicable to $\mathcal{F}^i$ for all $i \geq k$ (**fairness**).*

The chase *of $\mathcal{K}$, denoted by $\mathcal{K}^\infty$, is the is the union of all sets in some (arbitrarily chosen) chase sequence of $\mathcal{K}$.*

For the rest of the paper, we fix a Horn-$\mathcal{SRIQ}_\sqcap$ KB $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ and some (possibly infinite) chase sequence $\mathcal{O}^0, \mathcal{O}^1, \ldots$ for $\mathcal{O}$. For all $i \geq 1$, let $\rho_i \in \mathcal{T}$ be an axiom and σ_i a substitution s.t. $\mathcal{O}^i = \mathcal{O}^{i-1}\langle \rho_i, \sigma_i \rangle$. By abuse of notation, we write $P(a_1, \ldots, a_n) \in \mathcal{F}$, with $\mathcal{F}$ a set of facts, $P \in \mathsf{N_c} \cup \mathsf{N_r}$, and $a_1, \ldots, a_n \in \mathsf{N_i}$, if $P(b_1, \ldots, b_n) \in \mathcal{F}$ for some $b_1, \ldots, b_n \in 2^{\mathsf{N_i}}$ with $a_i \in b_i$ for all $i \in [\![1, n]\!]$.

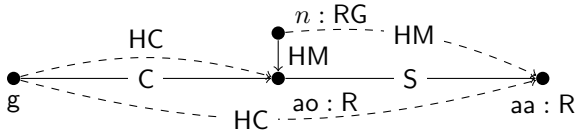

Figure 2: Chase of $\mathcal{O}_x = \langle \mathcal{T}_{ex}, \mathcal{A}_{ex} \rangle$ from Example 1

Theorem 1. *A KB $\mathcal{K}$ is satisfiable iff $\bot(t) \notin \mathcal{K}^\infty$ for all $t \in \mathsf{N_{gt}}$. If $\mathcal{K}$ is satisfiable, $\mathcal{K} \models \alpha$ iff $\alpha \in \mathcal{K}^\infty$ for every assertion α.*

We later show that the every chase step in a chase sequence of a Horn-$\mathcal{SRIQ}_\sqcap$ ontology reflects the "forest-shaped" when we restrict to facts containing at least one null, which corresponds to the well-known forest-model property of Horn-$\mathcal{SRIQ}_\sqcap$. In the presence of complex roles, the forest-model property is not entirely apparent in the chase steps of an ontology. To characterise this property, we distinguish binary facts in the chase that are not produced via the application of axioms of the Type ($\circ$) with $n \geq 2$, or the propagation of such facts.

All binary facts in $\mathcal{O}^0$ are *direct*. For all $i \geq 1$, a binary fact $\phi \in \mathcal{O}^i \setminus \mathcal{O}^{i-1}$ is *direct* iff ρ_i is of Type ($\exists$) or ($\sqcap_\mathsf{r}$); ρ_i is of Type ($\circ$) with $n = 1$ and $R_1(\sigma_i(x_0), \sigma_i(x_1)) \in \mathcal{O}^{i-1}$ is direct; or ρ_i is of Type ($\leq$), and there is a direct fact $\phi' \in \mathcal{O}^{i-1}$ s.t. $\phi'(\sigma_i)^{\mathsf{rn}}_{x,y} = \phi$. For $i \geq 0$, we write $\mathsf{D}(\mathcal{O}^i)$ to denote the set of all direct facts in $\mathcal{O}^i$.

Example 2. *Consider the TBox $\mathcal{T}_{ex}$ and ABox $\mathcal{A}_{ex}$ from Example 1. The chase of $\mathcal{O}_x = \langle \mathcal{T}_{ex}, \mathcal{A}_{ex} \rangle$ is depicted in Figure 2, where direct and not direct facts are represented using full and dashed arrows, respectively. Note that n is a null introduced by the chase.*

If we consider only the direct facts that occur in the chase sequence of an ontology, we can establish the "forest model property" reflected in every chase step of this sequence. For all $i \geq 0$, let $\mathsf{F}(\mathcal{O}^i)$ be the graph s.t. every $a \in 2^{\mathsf{N_i}}$ in $\mathcal{O}^i$ is a node in $\mathsf{F}(\mathcal{O}^i)$, and $t_{n-1} \to t_n \in \mathsf{F}(\mathcal{O}^i)$ if there is a sequence of facts $R_1(t_0, t_1), \ldots, R_n(t_{n-1}, t_n) \in \mathsf{D}(\mathcal{O}^i)$ with $t_0 \in 2^{\mathsf{N_i}}$ and $t_i \neq t_j$ for all $0 \leq i < j \leq n$.

Lemma 1. *For all $i \geq 0$,*

- *all nulls in $\mathcal{O}^i$ occur as nodes in $\mathsf{F}(\mathcal{O}^i)$, and*
- *$\mathsf{F}(\mathcal{O}^i)$ is a rooted forest where every individual node is a root, and every null node is not.*

Non-Deterministic Automata In our approach, we need to trace the paths of complex roles in the chase of a Horn-$\mathcal{SRIQ}_\sqcap$ KB that traverse only direct facts. To do so, we make use of well-known automata techniques from (Horrocks, Kutz, and Sattler 2006; Kazakov 2010). Here, we use non-deterministic finite automata (NFAs) in a rather informal way, and use the notation $p \to_R q \in \mathcal{N}$ to denote that, in the NFA $\mathcal{N}$, there is a transition from a state p to a state q with the letter R, instead of introducing transition relations formally.

Definition 3. *For a TBox $\mathcal{T}$, let $\mathcal{T}_- \supseteq \mathcal{T}$ be the TBox with $R_n^- \circ \ldots \circ R_1^- \sqsubseteq S^- \in \mathcal{T}_-$ for every axiom of Type ($\circ$) in $\mathcal{T}$.*

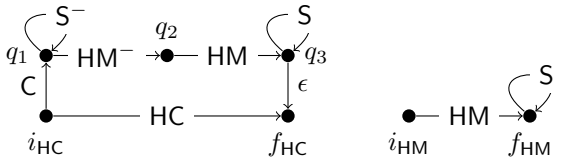

Figure 3: The NFA $\mathcal{N}_{\mathcal{T}_{ex}}(\mathsf{HC})$ and $\mathcal{N}_{\mathcal{T}_{ex}}(\mathsf{HM})$

For every $V \in \mathsf{N_r}$, the NFA $\mathcal{N}_\mathcal{T}(V)$ is the smallest NFA s.t. $i_V \to_V f_V \in \mathcal{N}_\mathcal{T}(V)$ with i_V and f_V the only initial and final states; and for every transition $q \to_S \hat{q} \in \mathcal{N}_\mathcal{T}(V)$ and every axiom in $\mathcal{T}_-$ of the form ($\circ$), we have

- *if $n = 1$ and $R_1 = S^-$, then $q \to_{S^-} \hat{q} \in \mathcal{N}_\mathcal{T}(V)$,*
- *if $n = 2$, $R_1 = S$, and $R_2 = S$, then $\hat{q} \to_\epsilon q \in \mathcal{N}_\mathcal{T}(V)$,*
- *Otherwise,*
  - *if $R_1 \neq S = R_n$, then $q \to_\epsilon q_0 \to_{R_1} q_1 \to_{R_2} q_2 \to_{R_3} \ldots \to_{R_{n-1}} q_{n-1} \to_\epsilon q \in \mathcal{N}_\mathcal{T}(V)$,*
  - *if $R_1 = S \neq R_n$, then $\hat{q} \to_\epsilon q_1 \to_{R_2} q_2 \to_{R_3} q_3 \to_{R_4} \ldots \to_{R_n} q_n \to_\epsilon \hat{q} \in \mathcal{N}_\mathcal{T}(V)$, and*
  - *if $R_1 \neq S \neq R_n$, then $q \to_\epsilon q_0 \to_{R_1} q_1 \to_{R_2} q_2 \to_{R_3} \ldots \to_{R_n} q_n \to_\epsilon \hat{q} \in \mathcal{N}_\mathcal{T}(V)$.*

In the above, states q_i are assumed to be fresh and distinct.

Our definition of NFA coincides with that from (Horrocks, Kutz, and Sattler 2006) in the sense that the resulting NFA $\mathcal{N}_\mathcal{T}(R)$ for any $R \in \mathsf{N_r}$ does recognise the same language. With analogous arguments to those presented by Horrocks et al., we can show the following claim.

Lemma 2. *For all $i \geq 0$, if $\mathcal{O}^i$ is closed under the application of axioms of Type ($\sqcap_\mathsf{r}$), there is a binary fact $R(t, u) \in \mathcal{O}^i$ iff there are some $S_1(t, t_1), \ldots, S_n(t_{n-1}, u) \in \mathsf{D}(\mathcal{O}^i)$ with $S_1 \cdot \ldots \cdot S_n \in \mathcal{N}_\mathcal{T}(R)$.*

Given a $\mathsf{P} = R_1 \cdot \ldots \cdot R_n$ with $R_1, \ldots, R_n \in \mathsf{N_r}$, we write $q \to^*_\mathsf{P} \hat{q} \in \mathcal{N}_\mathcal{T}(R)$ (resp. $\mathsf{P} \in \mathcal{N}_\mathcal{T}(R)$) to indicate that there is a path P from q to $\hat{q}$ (resp. i_R to f_R) in $\mathcal{N}_\mathcal{T}(R)$.

Example 3. *Consider $\mathcal{O}_x = \langle \mathcal{T}_{ex}, \mathcal{A}_{ex} \rangle$ with TBox $\mathcal{T}_{ex}$ and ABox $\mathcal{A}_{ex}$ from Example 1. The NFA $\mathcal{N}_{\mathcal{T}_{ex}}(\mathsf{HC})$ and $\mathcal{N}_{\mathcal{T}_{ex}}(\mathsf{HM})$ are depicted in Figure 3 (for the sake of clarity, we have removed some ϵ-transitions). As implied by Lemma 2 and since $\mathsf{HC}(\mathsf{g}, \mathsf{aa})$, we have $\mathsf{C}(\mathsf{g}, \mathsf{ao})$, $\mathsf{HM}^-(\mathsf{ao}, n)$, $\mathsf{HM}(n, \mathsf{ao}), \mathsf{S}(\mathsf{ao}, \mathsf{aa}) \in \mathsf{D}(\mathcal{O}_x^\infty)$ such that $\mathsf{C} \cdot \mathsf{HM}^- \cdot \mathsf{HM} \cdot \mathsf{S} \in \mathcal{N}_{\mathcal{T}_{ex}}(\mathsf{HC})$ (see Figure 2).*

Datalog Rewritings in Horn-$\mathcal{SRIQ}_\sqcap$

In this section, we define the Datalog AR-rewriting $\mathcal{R}_\mathcal{T}$ for the TBox $\mathcal{T}$ and discuss complexity results.

Definition 4. *A rule set $\mathcal{R}$ is an* AR-rewriting *for $\mathcal{T}$ iff, for every ABox $\mathcal{A}$ and assertion α over $\mathsf{sig}(\mathcal{T})$, $\langle \mathcal{T}, \mathcal{A} \rangle$ and $\langle \mathcal{R}, \mathcal{A} \rangle$ are equi-satisfiable and $\langle \mathcal{T}, \mathcal{A} \rangle \models \alpha$ iff $\langle \mathcal{R}, \mathcal{A} \rangle \models \alpha$.*

Let $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ and $\mathcal{K}_\mathcal{O} = \langle \mathcal{R}_\mathcal{T}, \mathcal{A} \rangle$. By Theorem 1, $\mathcal{R}_\mathcal{T}$ is an AR-rewriting only if the chase of $\mathcal{K}_\mathcal{O}$ coincides with the chase of $\mathcal{O}$ on all assertions over $\mathsf{sig}(\mathcal{T})$. The challenge in constructing Datalog AR-rewritings is that assertions in the $\mathcal{O}^\infty$ might be introduced by rule applications on facts with nulls, whilst no Datalog rule can introduce such terms.

$$\frac{\mathbb{A} \sqsubseteq \exists \mathbb{R}.(\mathbb{B} \sqcap \mathbb{C}) \qquad \mathbb{C} \sqsubseteq A}{\mathbb{A} \sqsubseteq \exists \mathbb{R}.(\mathbb{B} \sqcap \mathbb{C} \sqcap A)} \tag{1}$$

$$\frac{\mathbb{A} \sqsubseteq \exists (\mathbb{R} \sqcap \mathbb{S}).\mathbb{B} \qquad \mathbb{S} \sqsubseteq R}{\mathbb{A} \sqsubseteq \exists (\mathbb{R} \sqcap \mathbb{S} \sqcap R).\mathbb{B}} \tag{2}$$

$$\frac{\mathbb{A} \sqsubseteq \exists \mathbb{R}.(\mathbb{B} \sqcap \bot)}{\mathbb{A} \sqsubseteq \bot} \tag{3}$$

$$\frac{\mathbb{A} \sqsubseteq \exists (\mathbb{R} \sqcap R).\mathbb{B} \qquad A \sqsubseteq \forall R.B}{\mathbb{A} \sqcap A \sqsubseteq \exists (\mathbb{R} \sqcap R).(\mathbb{B} \sqcap B)} \tag{4}$$

$$\frac{\mathbb{A} \sqsubseteq \exists (\mathbb{R} \sqcap R^-).(\mathbb{B} \sqcap A) \qquad A \sqsubseteq \forall R.B}{\mathbb{A} \sqsubseteq B} \tag{5}$$

$$\frac{\begin{array}{l}\mathbb{A} \sqsubseteq \exists (\mathbb{R} \sqcap R).(\mathbb{B} \sqcap B) \qquad A \sqsubseteq \leqslant 1\, R.B \\ \mathbb{C} \sqsubseteq \exists (\mathbb{S} \sqcap R).(\mathbb{D} \sqcap B)\end{array}}{\mathbb{A} \sqcap \mathbb{C} \sqcap A \sqsubseteq \exists (\mathbb{R} \sqcap \mathbb{S} \sqcap R).(\mathbb{B} \sqcap \mathbb{D} \sqcap B)} \tag{6}$$

$$\frac{\begin{array}{l}\mathbb{A} \sqsubseteq \exists (\mathbb{R} \sqcap R^-).(\mathbb{B} \sqcap \mathbb{C} \sqcap A) \qquad A \sqsubseteq \leqslant 1\, R.B \\ \mathbb{C} \sqsubseteq \exists (\mathbb{S} \sqcap R).(\mathbb{D} \sqcap B \sqcap C)\end{array}}{\begin{array}{l}\mathbb{A} \sqcap B \sqsubseteq C \\ \mathbb{A} \sqcap B \sqsubseteq \exists (\mathbb{R} \sqcap R^- \sqcap \mathbb{S}^-).(\mathbb{B} \sqcap \mathbb{C} \sqcap A)\end{array}} \tag{7}$$

Figure 4: Derivation Rules where $A, B \in \mathsf{N_c}$, $R \in \mathsf{N_r}$, and $\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}$ and $\mathbb{R}, \mathbb{S}$ are conjunctions of elements in $\mathsf{N_c}$ and $\mathsf{N_r}$, respectively

Example 4. *Let $\mathcal{O}_x$ be the ontology from Example 1. Then, the assertion $\mathsf{HC}(\mathsf{g}, \mathsf{aa})$ is in $\mathcal{O}_x^\infty$ because $\mathsf{HC}(\mathsf{g}, \mathsf{ao})$, $\mathsf{HM}(n, \mathsf{ao}), \mathsf{HM}(n, \mathsf{aa}) \in \mathcal{O}_x$ (see Figure 2). Analogously, $\mathsf{R}(\mathsf{aa}) \in \mathcal{O}_x^\infty$ because $\mathsf{RG}(n), \mathsf{HM}(n, \mathsf{aa}) \in \mathcal{O}_x^\infty$. Note that the facts $\mathsf{HM}(n, \mathsf{ao}), \mathsf{HM}(n, \mathsf{aa})$, and $\mathsf{RG}(n)$ cannot occur in the case of a Datalog AR-rewriting, since $n \in \mathsf{N_0}$.*

To replicate assertion entailments in $\mathcal{K}_\mathcal{O}^\infty$ such as the ones highlighted in the previous example, we encode information in $\mathcal{K}_\mathcal{O}^\infty$ about the null successors of an individual in $\mathcal{O}^\infty$ using fresh concepts and roles. For all $R \in \mathsf{N_r}$ and states $q, \hat{q} \in \mathcal{N}_\mathcal{T}(R)$, we introduce the fresh concepts A_q and $R_{q,\hat{q}}$, and the fresh role R_q. Intuitively, these are used to encode the following information about $\mathcal{O}^\infty$ in $\mathcal{K}_\mathcal{O}^\infty$.

1. If $A_q(a) \in \mathcal{K}_\mathcal{O}^\infty$, then there are some $A(t_0) \in \mathcal{O}^\infty$, and some $R_1(t_0, t_1), \ldots, R_n(t_{n-1}, a) \in \mathsf{D}(\mathcal{O}^\infty)$ with $q \to^*_{R_1 \cdot \ldots \cdot R_n} \hat{q} \in \mathcal{N}_\mathcal{T}(R)$.
2. If $R_{q,\hat{q}}(a) \in \mathcal{K}_\mathcal{O}^\infty$, then there are some $R_1(a, t_1), \ldots, R_n(t_{n-1}, a) \in \mathsf{D}(\mathcal{O}^\infty)$ with $t_1, \ldots, t_{n-1} \in \mathsf{N_0}$ and $q \to^*_{R_1 \cdot \ldots \cdot R_n} \hat{q} \in \mathcal{N}_\mathcal{T}(R)$.
3. If $R_q(a, b) \in \mathcal{K}_\mathcal{O}^\infty$, then $S_1(a, t_1), \ldots, S_n(t_{n-1}, b) \in \mathsf{D}(\mathcal{O}^\infty)$ with $i_R \to^*_{S_1 \cdot \ldots \cdot S_n} q \in \mathcal{N}_\mathcal{T}(R)$.

Note that all terms t_i may possibly be nulls that do not appear in the chase of $\mathcal{K}_\mathcal{O}$.

To ascertain when information about one of these predicates needs to be used in $\mathcal{K}_\mathcal{O}$, we make use of a sound saturation calculus from (Eiter et al. 2012), shown in Figure 4, which we also use to infer further axioms relevant to our Datalog program. Since this calculus was originally designed for Horn-$\mathcal{SHIQ}$, we first need to extend

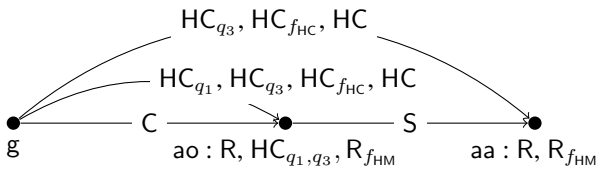

Figure 5: Representation of $\mathcal{K}_{\mathcal{O}}^{\infty}$ with $\mathcal{O}$ from Example 1

our input TBox $\mathcal{T}$ to a TBox $\mathcal{T}_+$ in which the behaviour of axioms of Type ($\circ$) is sufficiently simulated. For instance, if the calculus derives from $\mathcal{T}_+$ an axiom of the form $\mathbb{A} \sqsubseteq A_q$, then we can conclude that, for every term t s.t. $B(t) \in \mathcal{O}^{\infty}$ for every $B \in \mathbb{A}$, there is a set of direct facts $A(t_0), R_1(t_0, t_1), \ldots, R_n(t_{n-1}, a) \in \mathcal{O}^{\infty}$ with a corresponding path in the automata, irrespectively of the ABox $\mathcal{A}$. We further augment $\mathcal{T}_+$ to a TBox $\mathcal{T}_\times$ that allows us to trace paths in possible chases for $\mathcal{T}$. Using the inferences from this calculus, we then describe the rewriting $\mathcal{R}_{\mathcal{T}}$.

Definition 5. *Let $\mathsf{B}(\mathcal{T})$ be the set of axioms that, for every axiom $\rho \in \mathcal{T}$ of Type ($\forall$), contains $A \sqsubseteq A_{i_R}, A_{f_R} \sqsubseteq B$, and $A_q \sqsubseteq \forall S.A_{\hat{q}} \in \mathsf{B}(\mathcal{T})$ for every $q \to_S^* \hat{q} \in \mathcal{N}_{\mathcal{T}}(R)$ with $S \in \mathsf{N_r}$. Let $\mathcal{T}_+ = \mathcal{T}_- \cup \mathsf{B}(\mathcal{T})$, and $\mathcal{T}_\times = \mathcal{T}_- \cup \mathsf{B}(\mathcal{T} \cup \bigcup_{R \in \mathsf{N_r}} \{X \sqsubseteq \forall R.Y\})$, with X and Y fresh concepts.*

Then, $\mathcal{R}_{\mathcal{T}}$ is the Datalog rule set that contains every axiom in $\mathcal{T}_+$ that is not of Type ($\exists$), and every axiom that can be inferred using the implications described in Table 1.

Theorem 2. *The rule set $\mathcal{R}_{\mathcal{T}}$ is an AR-rewriting of $\mathcal{T}$.*

Example 5. *Let $\mathcal{O}_x$ be the ontology from Example 1. Then, the Datalog rule set $\mathcal{R}_{\mathcal{T}_{ex}}$ contains (amongst others) all the rules in $\mathcal{T}_{ex}$ that are not of Type ($\exists$), as well as the following.*

$$\mathsf{R}(x) \to \mathsf{R}_{f_{\mathsf{HM}}}(x) \qquad \mathsf{R}_{f_{\mathsf{HM}}}(x) \to \mathsf{R}(x)$$
$$\mathsf{R}_{f_{\mathsf{HM}}}(x) \wedge \mathsf{S}(x, y) \to \mathsf{R}_{f_{\mathsf{HM}}}(y)$$
$$\mathsf{C}(x, y) \to \mathsf{HC}_{q_1}(x, y) \qquad \mathsf{R}(x) \to \mathsf{HC}_{q_1,q_3}(x)$$
$$\mathsf{HC}_{q_1}(x, y) \wedge \mathsf{HC}_{q_1,q_3}(x) \to \mathsf{HC}_{q_3}(x, y)$$
$$\mathsf{HC}_{q_3}(x, y) \to \mathsf{HC}_{f_{\mathsf{HC}}}(x, y) \qquad \mathsf{HC}_{f_{\mathsf{HC}}}(x, y) \to \mathsf{HC}(x, y)$$
$$\mathsf{HC}_{f_{\mathsf{HC}}}(x, y) \wedge \mathsf{S}(y, z) \to \mathsf{HC}_{f_{\mathsf{HC}}}(x, z)$$

The chase of $\mathcal{K}_{\mathcal{O}_x}$ is depicted in Figure 5. Note that $\mathcal{K}_{\mathcal{O}_x}^{\infty}$ contains every assertion in $\mathcal{O}^{\infty}$.

While we provide for full proofs of Theorem 2 in the extended version of the paper, we give an overview of some of the main technical ideas in this section. While showing soundness of our approach is not as challenging, we focus on the argument showing completeness of the AR-rewriting $\mathcal{R}_{\mathcal{T}}$. Before discussing this proof, we give an intermediate result.

Lemma 3. *For a TBox $\mathcal{T}$, an ABox $\mathcal{A}$ and a fact set $\mathcal{F}$ defined over $\mathsf{sig}(\mathcal{T})$, $\langle \mathcal{T}, \mathcal{A} \rangle$ is satisfiable iff $\langle \mathcal{T}_+, \mathcal{A} \rangle$ is, and $\langle \mathcal{T}, \mathcal{A} \rangle \models \mathcal{F}$ iff $\langle \mathcal{T}_+, \mathcal{A} \rangle \models \mathcal{F}$.*

Since $\mathcal{T}_+ \supseteq \mathcal{T}$, the "if" direction of this lemma follows trivially from monotonicity of logical entailment. The "only if" direction is proven in the extended version of the paper.

By Lemma 3, it suffices to show that our Datalog rewritings entail the same assertions as $\mathcal{T}_+$ in order to show completeness of our rewriting, which by Theorem 1 is consequence of the following lemma.

Lemma 4. *For a TBox $\mathcal{T}$, an ABox $\mathcal{A}$ and an assertion α over $\mathsf{sig}(\mathcal{T})$,*

- *if $\bot(t) \in \langle \mathcal{T}_+, \mathcal{A} \rangle^{\infty}$ with $t \in \mathsf{N_{gt}}$, then $\bot(u) \in \langle \mathcal{R}_{\mathcal{T}}, \mathcal{A} \rangle^{\infty}$ for some $u \in 2^{\mathsf{N_i}}$, and*
- *if $\alpha \in \langle \mathcal{T}_+, \mathcal{A} \rangle^{\infty}$, then $\alpha \in \langle \mathcal{R}, \mathcal{A} \rangle^{\infty}$.*

Let $\mathcal{O}_+^0, \mathcal{O}_+^1, \ldots$ be a chase sequence for the ontology $\mathcal{O}_+ = \langle \mathcal{T}_+, \mathcal{A} \rangle$ where axioms of Type ($\sqcap_{\mathsf{r}}$) are applied with the highest priority. For every $i \in [\![1, n]\!]$, we select an axiom $\rho_i \in \mathcal{T}_+$ and a substitution σ_i s.t. $\mathcal{O}_+^i = \mathcal{O}_+^{i-1} \langle \rho_i, \sigma_i \rangle$.

To prove Lemma 4, we show via induction that for every $i \geq 1$ and every assertion $\alpha \in \mathcal{O}_+^i$, we have $\alpha \in \mathcal{K}_{\mathcal{O}}^{\infty}$. The base case of this induction is trivial, since $\mathcal{O}_+^0 = \mathcal{A}$ and $\mathcal{A} \subseteq \mathcal{K}_{\mathcal{O}}^{\infty}$ by Definition 2. For the induction step, we provide a thorough case analysis based on the type of the axiom ρ_i, and the type of the elements occurring in the range of σ_i. Since $\alpha \in \mathcal{K}_{\mathcal{O}}^{\infty}$ for every assertion $\alpha \in \mathcal{O}^{i-1}$ by the induction hypothesis, many cases follow trivially. The more challenging cases are the following.

1. ρ_i is of Type ($\circ$), $\sigma_i(x_0), \sigma_i(x_n) \in 2^{\mathsf{N_i}}$ and $\sigma_i(x_j) \in \mathsf{N_0}$ for $j \in \{1, \ldots, n-1\}$.
2. ρ_i is of Type ($\forall$), $\sigma_i(x) \in \mathsf{N_0}$ and $\sigma_i(y) \in \mathsf{N_i}$.
3. ρ_i is of Type ($\leq$), and a) $\sigma_i(y) \in 2^{\mathsf{N_i}}$ and $\sigma_i(x), \sigma_i(z) \in \mathsf{N_0}$, or b) $\sigma_i(x), \sigma_i(y) \in 2^{\mathsf{N_i}}$ and $\sigma_i(z) \in \mathsf{N_0}$.

Cases in which ρ_i is of Type ($\leq$), and either $\sigma_i(z) \in 2^{\mathsf{N_i}}$ and $\sigma_i(x), \sigma_i(y) \in \mathsf{N_0}$, or $\sigma_i(x), \sigma_i(z) \in 2^{\mathsf{N_i}}$ and $\sigma_i(y) \in \mathsf{N_0}$, are also non-trivial, but analogous to Cases 3a) and 3b).

In all of the challenging cases, the occurrence of facts containing nulls in $\mathcal{O}_+^{i-1}$ results in the introduction of new assertions in $\mathcal{O}_+^i$—a situation previously illustrated in Example 4. To illustrate our completeness argument, we give a brief proof sketch that shows that induction step for Case (1). First, we introduce a preliminary lemma, which ensures that an axiom as used for Rule ($\circlearrowleft$) is derived by the calculus if there is a corresponding cyclic path along nulls in $\mathcal{O}^{\infty}$.

Lemma 5. *Let $i \geq 1$, $R_1(t_0, t_1), \ldots, R_n(t_{n-1}, t_n) \in \mathsf{D}(\mathcal{O}_+^i)$, and $q, \hat{q} \in \mathcal{N}_{\mathcal{T}}(R)$ with $q \neq \hat{q}$. If*

- *$q \to_{\mathsf{P}}^* \hat{q} \in \mathcal{N}_{\mathcal{T}}(R)$ with $\mathsf{P} = R_1 \cdot \ldots \cdot R_n$, and*
- *$t_0 \in 2^{\mathsf{N_i}}$, and $t_1, \ldots, t_{n-1} \in \mathsf{N_0}$;*

then there exists $\mathbb{A} \subseteq \{A \mid A(t_0) \in \mathcal{O}_+^i\}$ s.t. $\mathbb{A} \sqcap X_q \sqsubseteq X_{\hat{q}} \in \Gamma(\mathcal{T}_\times)$.

This result can be shown via induction on the depth of the sequence $R_1(t_0, t_1), \ldots, R_n(t_{n-1}, t_n)$—the maximum minus the mimimum depth of a term in $t_0, \ldots, t_n$ in the rooted forest $\mathsf{F}(\mathcal{O}_+^i)$. We proceed with the proof for case (1).

Proof (Sketch). Let ρ_i be an axiom of the form $R_1 \circ \ldots \circ R_n \sqsubseteq S \in \mathcal{T}_+$. Then, $R_1(\sigma_i(x_0), \sigma_i(x_1)), \ldots, R_n(\sigma_i(x_{n-1}), \sigma_i(x_n)) \in \mathcal{O}_+^{i-1}$.

By Lemma 2 and the fact that $\mathcal{O}_+^i$ is closed under application of rules of Type ($\sqcap_{\mathsf{r}}$), there is a sequence

$$\bigwedge_{D\in\mathbb{D}} D(x) \to A(x) \quad \Longleftarrow \quad \mathbb{D} \sqsubseteq A \in \Gamma(\mathcal{T}_\times) \qquad (\sqcap)$$

$$\bigwedge_{D\in\mathbb{D}} D(x) \to R_{q,\hat{q}}(x) \quad \Longleftarrow \quad R \in \mathsf{N_r},\, q,\hat{q} \in \mathcal{N}_R(\mathcal{T}), \text{ and } \mathbb{D} \sqcap X_q \sqsubseteq X_{\hat{q}} \in \Gamma(\mathcal{T}_\times) \qquad (\circlearrowleft)$$

$$A(x) \wedge \bigwedge_{D\in\mathbb{D}\cup A} D(x) \wedge R(x,y) \wedge B(y) \to C(y) \quad \Longleftarrow \quad A \sqsubseteq \leqslant 1\, R.B, \mathbb{D} \sqsubseteq \exists(\mathbb{R} \sqcap R).(\mathbb{A} \sqcap B \sqcap C) \in \Gamma(\mathcal{T}_\times) \qquad (\lhd 1)$$

$$A(x) \wedge \bigwedge_{D\in\mathbb{D}} D(x) \wedge R(x,y) \wedge B(y) \to S(x,y) \quad \Longleftarrow \quad A \sqsubseteq \leqslant 1\, R.B, \mathbb{D} \sqsubseteq \exists(\mathbb{R} \sqcap R \sqcap S).(\mathbb{A} \sqcap B) \in \Gamma(\mathcal{T}_\times) \qquad (\lhd 2)$$

$$S(x,y) \to R_q(x,y) \quad \Longleftarrow \quad R, S \in \mathsf{N_r} \text{ and } i_R \to_S^* q \in \mathcal{N}_\mathcal{T}(R) \qquad (\text{R } 1)$$

$$R_{i_R,q}(x) \to R_q(x,x) \quad \Longleftarrow \quad R \in \mathsf{N_r} \text{ and } R_{i_R,q} \in \mathcal{R}_\mathcal{T} \qquad (\text{R } 2)$$

$$R_q(x,y) \wedge S(y,z) \to R_{\hat{q}}(x,z) \quad \Longleftarrow \quad R, S \in \mathsf{N_r} \text{ and } q \to_S^* \hat{q} \in \mathcal{N}_\mathcal{T}(R) \qquad (\text{R } 3)$$

$$R_q(x,y) \wedge R_{q,\hat{q}}(y) \to R_{\hat{q}}(x,y) \quad \Longleftarrow \quad R \in \mathsf{N_r} \text{ and } R_{q,\hat{q}} \in \mathcal{R}_\mathcal{T} \qquad (\text{R } 4)$$

$$R_{f_R}(x,y) \to R(x,y) \quad \Longleftarrow \quad R \in \mathsf{N_r} \qquad (\text{R } 5)$$

Table 1: Rules to construct $\mathcal{R}_\mathcal{T}$, where $\Gamma(\mathcal{T}_\times)$ is the saturation of $\mathcal{T}_\times$ by the rules in Figure 4 and all concepts A and B and those in the conjunctions $\mathbb{D}$ and $\mathbb{A}$ occur $\mathcal{T}_+$.

$V_1(t_0,t_1),\dots,V_m(t_{m-1},t_m) \in \mathsf{D}(\mathcal{O}_+^{i-1})$ with $\sigma_i(x_{j-1}) = t_0$, $\sigma_i(x_j) = t_m$, and $V_1 \cdot \ldots \cdot V_m \in \mathcal{N}_\mathcal{T}(R_j)$ for every $j \in \{1,\dots,n\}$ (note that possibly $m = 1$). By concatenating these sequences, we can construct a sequence $V_1(t_0,t_1),\dots,V_m(t_{m-1},t_m) \in \mathsf{D}(\mathcal{O}_+^{i-1})$ s.t. $\sigma_i(x_0) = t_0$, $\sigma_i(x_n) = t_m$, and $V_1 \cdot \ldots \cdot V_m \in \mathcal{N}_\mathcal{T}(S)$. Hence, there are states $q_0,\dots,q_m$ s.t. $q_0 = i_V$, $q_m = f_V$, and $q_0 \to_{W_1} q_1 \to_{W_2} q_2 \dots \to_{W_m} q_m \in \mathcal{N}_\mathcal{T}(V)$. Let $k_0,\dots,k_o$ be the longest sorted sequence of natural numbers with $t_{k_j} \in 2^{\mathsf{N_i}}$ for all $j \in \{0,\dots,o\}$. We show via induction that $S_{q_{k_j}}(t_0,t_{k_j}) \in \mathcal{K}_\mathcal{O}^\infty$ for all $j \in \{1,\dots,o\}$. In turn, this implies $S(\sigma_i(x),\sigma_i(y)) \in \mathcal{K}_\mathcal{O}^\infty$ since $S_{q_{f_S}}(x,y) \to S(x,y) \in \mathcal{R}_\mathcal{T}$ as $k_0 = t_0 = \sigma_i(x)$, $t_{k_o} = t_m = \sigma_i(y)$, and $q_{k_m} = f_S$.

To show the base case, we check that $S_{q_{k_1}}(t_0,t_{k_1}) \in \mathcal{K}_\mathcal{O}^\infty$. We consider two possible cases a) and b) depending on whether $k_1 = 1$. a) Let $k_1 = 1$. Then, $W_1(t_0,t_1) \in \mathcal{K}_\mathcal{O}^\infty$ by the inductive hypothesis. Since $W_1(x,y) \to S_{q_1}(x,y) \in \mathcal{R}_\mathcal{T}$, $S_{q_1}(t_0,t_1) \in \mathcal{K}_\mathcal{O}^\infty$. b) Let $k_1 > 1$. By Lemma 1, $t_{k_1} = t_0$. As shown in the extended version of the paper $\mathbb{A} \sqcap X_{i_S} \sqsubseteq X_{q_{k_1}} \in \Gamma(\mathcal{T}_\times)$ with $\mathbb{A} \subseteq \mathsf{N_c}^{i-1}(t_0)$ and hence, $\mathbb{A}(x) \to S_{i_S,q_{k_1}}(x) \in \mathcal{R}_\mathcal{T}$. By the inductive hypothesis, $\mathbb{A}(t_0) \in \mathcal{K}_\mathcal{O}^\infty$ and hence, $S_{i_S,q_{k_1}}(t_0) \in \mathcal{K}_\mathcal{O}^\infty$. Since $S_{i_S,q_{k_1}}(x) \to S_{q_{k_1}}(x,x) \in \mathcal{R}_\mathcal{T}^\infty$, $S_{q_{k_1}}(t_0,t_{k_1}) \in \mathcal{K}_\mathcal{O}^\infty$.

To show the induction step, we verify that, for all $j \in \{2,\dots,o\}$, $S_{q_{k_j}}(t_0,t_{k_j}) \in \mathcal{K}_\mathcal{O}^\infty$ provided that $S_{q_{k_{j-1}}}(t_0,t_{k_{j-1}}) \in \mathcal{K}_\mathcal{O}^\infty$. We consider two possible cases a) and b) depending on whether $k_1 = 1$. Let $k_j = k_{j-1} + 1$. Then, $W_{k_j}(t_{k_{j-1}},t_{k_j}) \in \mathcal{K}_\mathcal{O}^\infty$ by the inductive hypothesis. Since $S_{q_{k_{j-1}}}(x,y) \wedge W_{k_j}(y,z) \to S_{q_{k_j}}(x,z) \in \mathcal{R}_\mathcal{T}$, $S_{q_{k_j}}(t_0,t_{k_j}) \in \mathcal{K}_\mathcal{O}^\infty$. Let $k_j > k_{j-1} + 1$. Then, $t_{k_j} = t_{k_{j-1}}$ by Lemma 1. This case is analogous to the second case considered in the proof of the base case. □

In addition to showing correctness, we can show that our approach is worst-case optimal for Horn-$\mathcal{SRIQ}_\sqcap$ and even for less expressive DLs such as $\mathcal{ELH}$ and Horn-$\mathcal{SHIQ}$.

Definition 6. *An axiom set is a* Horn-$\mathcal{SHIQ}$ TBox *if, for every axiom $\rho \in \mathcal{T}$ of Type* ($\circ$)*, we have that a) $n = 1$ or b) $n = 2$, and $R_1 = R_2 = S$.*

A $\mathcal{ELH}$ TBox *$\mathcal{T}$ is a set containing axioms of Type* ($\sqcap$)*,* ($\exists$)*,* ($\circ$)*, and of the form $\exists R.A \sqsubseteq B$ with $A, B \in \mathsf{N_c}$ and $R \in \mathsf{N_r}$ s.t. a) $n = 1$ for every axiom of the form* ($\circ$) *and b) for every $R \in \mathsf{N_r}$, $\mathcal{T}$ uses R or R^-, but not both.*

Axioms of the form $\exists R.A \sqsubseteq B$ are equivalent to $A \sqsubseteq \forall R^-.B$, which is why $\mathcal{ELH}$ is included in Horn-$\mathcal{SRIQ}_\sqcap$.

Theorem 3. *Let $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ be an ontology. If $\mathcal{T}$ is Horn-$\mathcal{SRIQ}_\sqcap$/Horn-$\mathcal{SHIQ}$/$\mathcal{ELH}$, then we can compute $\mathcal{R}_\mathcal{T}$ and $\langle \mathcal{R}_\mathcal{T}, \mathcal{A} \rangle^\infty$ in* 2ExpTime/ExpTime/PTime*, respectively.*

Finally, we show that our rewritings can be transformed into DLP TBoxes. This feature may prove useful for users that want to produce KBs that are expressible using the OWL standard.

Definition 7. *A* DLP TBox *is an axiom set that a) does not contain axioms of Type* ($\exists$) *and b) may contain axioms of the form $\bigsqcap_{i=1}^n A_i \sqsubseteq \exists R.Self$ with $A \in \mathsf{N_c}$ and $R \in \mathsf{N_r}$.*

Definition 8. *Given a TBox $\mathcal{T}$, the* DLP-rewriting $\mathcal{T}_{\text{dlp}}$ of $\mathcal{T}$ *is the TBox containing every DLP axiom in $\mathcal{R}_\mathcal{T}$ which additionally satisfies all of the following.*

1. *If $\bigwedge_{A\in\mathbb{A}} A(x) \wedge R(x,y) \wedge B(y) \to C(y) \in \mathcal{R}_\mathcal{T}$, then $\mathbb{A} \sqsubseteq X_\mathbb{A}$, $X_\mathbb{A} \sqsubseteq \forall R.X_{R^-,\mathbb{A}}$, $X_{R^-,\mathbb{A}} \sqcap B \sqsubseteq C \in \mathcal{T}_{dlp}$.*
2. *If $\bigwedge_{A\in\mathbb{A}} A(x) \wedge R(x,y) \wedge B(y) \to S(x,y) \in \mathcal{R}_\mathcal{T}$, then $\mathbb{A} \sqsubseteq \exists W_\mathbb{A}.Self$, $B \sqsubseteq \exists W_B.Self$, $W_\mathbb{A} \circ R \circ W_B \sqsubseteq S \in \mathcal{T}_{dlp}$.*
3. *If $R_q(x,y) \wedge R_{q,\hat{q}}(y) \to R_{\hat{q}}(x,y) \in \mathcal{R}_\mathcal{T}$, then $R_{q,\hat{q}} \sqsubseteq \exists W_{q,\hat{q}}.Self, R_q \circ W_{q,\hat{q}} \sqsubseteq R_{\hat{q}} \in \mathcal{T}_{dlp}$.*

In the above, all $X_\mathbb{A}$ and $R.X_{R^-,\mathbb{A}}$ are fresh concepts unique for every $\mathbb{A} \subseteq \mathsf{N_c}$ and $R \in \mathsf{N_r}$, and all $W_\mathbb{A}$ and $W_{q,\hat{q}}$ are fresh roles unique for every $W \in \mathsf{N_r}$ and the states q and $\hat{q}$.

The rules introduced in (1)–(3) in Definition 8 correspond to consequence-preserving transformations from rules to axioms described in (Krötzsch, Rudolph, and Hitzler 2008). From this, it follows that $\mathcal{T}_{\text{dlp}}$ is an AR-rewriting of $\mathcal{T}$.

Evaluation

We implement our rewriting technique in Java using the OWL-API (Horridge and Bechhofer 2011) to handle OWL ontology files, and Clipper (Eiter et al. 2012) to apply the calculus from Figure 4. We performed two different experiments to validate the practical usefulness of our approach.

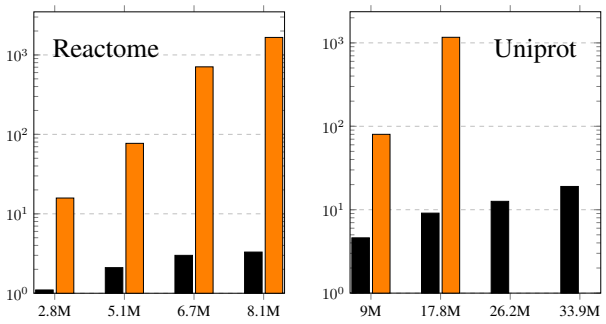

Figure 6: Times in seconds for RDFox (dark) and Konclude (bright), each over four ABoxes with increasing numbers of assertions.

AR on Data-Intensive Ontologies We compared the performance of performing AR using our Datalog rewritings versus using the DL reasoner Konclude. We considered two real-world, data-intensive ontologies from the biological domain, *Reactome* and *Uniprot*, which were used in the evaluation of PAGOdA (Zhou et al. 2015). We have normalised these ontologies and removed axioms not expressible in Horn-$\mathcal{SRIQ}_\sqcap$. Also, we enriched Reactome and Uniprot with three and five axioms of Type ($\circ$), respectively, as neither ontology contained axioms of this form. These axioms are listed in the last section of the appendix which is contained in the extended version of this paper (Carral, González, and Koopmann 2018). The resulting ontologies contained 485 (Reactome), and 304 (Uniprot) terminological axioms, respectively. For each ontology, we considered ABoxes of various sizes, generated by sampling the real-world ABoxes using the method by Zhou et al. (2015).

The rewritten Datalog programs for the Reactome and Uniprot TBoxes contained 539 and 367 rules, and were computed in 221 and 182 seconds, respectively. We used RDFox (SVN version 2776) as Datalog engine for computing the chase of our rewritings (Motik et al. 2014), and compared its performance with that of Konclude v0.6.2. We performed all experiments and computed both rewritings on a MacBook Pro with a 2,4 GHz Intel Core i5 and 8GB of RAM. Figure 6 shows the wall-clock times measured in this experiment, ignoring the time used for parsing and loading, in logarithmic scale. While Konclude reports detailed times, for RDFox we have measured the time from within our prototype. For more information, see the logs with the resulting evaluation can be found online. Konclude timed-out (with a one hour time limit) for the two largest of the Uniprot samples. Hence no times are reported there. Note that our implementation is performing full AR, whilst Konclude only performed class retrieval.

Size of Rewritings Computed To get an idea on how our approach would perform on other data-intensive real-world ontologies, we computed rewritings for a selected set of TBoxes from MOWLCorp (Matentzoglu et al. 2014). From each ontology in this corpus of DL ontologies, we removed axioms that, after normalisation, were not in Horn-$\mathcal{SRIQ}_\sqcap$, and selected from the resulting ontology set those which contained role chain axioms, and removed TBoxes with more than 1,000 axioms, since TBoxes with smaller sizes are more likely to be used on large data sets. Furthermore, we removed all those ontologies which belong to any of the profiles OWL EL, OWL RL, and OWL QL, since they admit polynomial reasoning even without a Datalog rewriting. This resulted in a set of 187 ontologies on which we applied our implemented rewriting procedure.

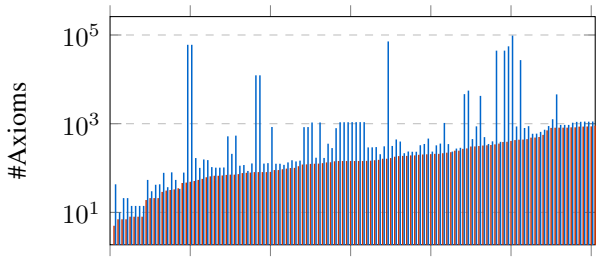

Figure 7: Sizes of TBoxes and their rewritings.

For 121 ontologies, rewritings could be computed without memory errors. Often, memory errors were caused by complex role chains in the TBox which lead to an explosion of the resulting automata. For instance, we found one degenerate ontology in the corpus with only 10 axioms, 4 of which were role chain axioms with 8 roles each. For this TBox, $\mathcal{T}_\times$ contained 86,264 axioms, which Clipper could not handle. We believe that ontologies of this form are unlikely to be used in practice to reason about large ABoxes.

The sizes of the successful rewritings are shown in Figure 7, where the red bars correspond to the number of axioms in the input ontologues, and the blue bars to the number of rules in the resulting Datalog rewritings. For some ontologies, the rewritings got substantially larger. This was expected, and in theory unavoidable, due to the double exponential time complexity of assertion entailment in Horn-$\mathcal{SRIQ}_\sqcap$: for Datalog, this complexity is only polynomial, which is why our rewritings are in the worst case double exponential in the size of the input. Our evaluation confirms that these blow-ups are not only of theoretical nature, but do happen for the considered ontologies. On the other hand, in a lot of cases, the size of the computed rule sets was still of similar dimensions: in 59% of cases, the the increase was at most by 100%, and in 74% of cases, it was at most by 200%.

Conclusions and Future Work

To the best of our knowledge, we present the first data-independent Datalog transformation for Horn-$\mathcal{SRIQ}_\sqcap$, an expressive DL that allows for the use of the role chain constructor. Furthermore, we show that our transformation is worst-case optimal for $\mathcal{ELH}$, Horn-$\mathcal{SHIQ}$, and Horn-$\mathcal{SRIQ}_\sqcap$, and that the resulting Datalog programs can be translated into DLP ontologies. We empirically show that a) the use of Datalog rewritings can outperform state-of-the-art reasoners and that b) we can construct rewritings of moderate sizes for many real-world ontologies.

As for future work, we aim to develop a rewriting technique for expressive DLs language that allows for the use of non-deterministic role constructors and role chains based on the calculi from (Cucala, Cuenca Grau, and Horrocks 2018; Bate et al. 2016). Also, we intend to further optimise our prototype implementation, in order to produce even smaller rewritings and show that these can be efficiently computed.

Acknowledgements

We thank Irina Dragoste for assisting us with the execution of the experiments, for which we used the servers from the Centre for Information Services and High Performance Computing (ZIH) at the Technische Universität Dresden. This work is partly funded by the DFG within the Center for Advancing Electronics Dresden (cfaed), the Collaborative Research Center CRC 912 (HAEC), and Emmy Noether grant KR 4381/1-1 (DIAMOND).

References

Ahmetaj, S.; Ortiz, M.; and Šimkus, M. 2016. Polynomial datalog rewritings for expressive dls with closed predicates. In *Proc. IJCAI'16*, 878–885. AAAI Press.

Bate, A.; Motik, B.; Cuenca Grau, B.; Simančík, F.; and Horrocks, I. 2016. Extending consequence-based reasoning to $\mathcal{SRIQ}$. In *Proc. KR'16*, 187–196. AAAI Press.

Callahan, A.; Cruz-Toledo, J.; and Dumontier, M. 2013. Ontology-based querying with Bio2RDF's linked open data. *J. Biomedical Semantics* 4(S-1):S1.

Carral, D.; Krötzsch, M.; Marx, M.; Ozaki, A.; and Rudolph, S. 2018. Preserving constraints with the stable chase. In *Proc. ICDT'18*, 12:1–12:19. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

Carral, D.; González, L.; and Koopmann, P. 2018. From Horn-$\mathcal{SRIQ}$ to Datalog: A data-independent transformation that preserves assertion entailment (extended version). LTCS-Report 18-14, Chair for Automata Theory, Institute for Theoretical Computer Science, Technische Universität Dresden. See https://lat.inf.tu-dresden.de/research/reports.html.

Cucala, D. T.; Cuenca Grau, B.; and Horrocks, I. 2018. Consequence-based reasoning for description logics with disjunction, inverse roles, number restrictions, and nominals. In *Proc. IJCAI'18*, 1970–1976. AAAI Press.

Eiter, T.; Ortiz, M.; Šimkus, M.; Tran, T.; and Xiao, G. 2012. Query rewriting for Horn-$\mathcal{SHIQ}$ plus rules. In *Proc. AAAI'12*. AAAI Press.

Grosof, B. N.; Horrocks, I.; Volz, R.; and Decker, S. 2003. Description logic programs: combining logic programs with description logic. In *Proc. WWW'03*, 48–57. ACM.

Horridge, M., and Bechhofer, S. 2011. The OWL API: A java API for OWL ontologies. *Semantic Web* 2(1):11–21.

Horrocks, I.; Kutz, O.; and Sattler, U. 2006. The irresistible $\mathcal{SRIQ}$. In *Proc. OWLED'05*, volume 188 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Hustadt, U.; Motik, B.; and Sattler, U. 2004. Reducing $\mathcal{SHIQ}^-$ description logic to disjunctive datalog programs. In *Proc. KR'2004*, 152–162. AAAI Press.

Kazakov, Y.; Krötzsch, M.; and Simančík, F. 2014. The incredible ELK: From polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies. *J. Autom. Reasoning* 53(1):1–61.

Kazakov, Y. 2010. An extension of complex role inclusion axioms in the description logic $\mathcal{SROIQ}$. In *Proc. IJCAR'10*, volume 6173 of *Lecture Notes in Computer Science*, 472–486. Springer.

Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2008. Description logic rules. In *Proc. ECAI'08*, 80–84. IOS Press.

Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2013. Complexities of Horn dls. *ACM Trans. Comput. Logic* 14(1):2:1–2:36.

Krötzsch, M. 2011. Efficient rule-based inferencing for OWL EL. In *Proc. IJCAI'11*, 2668–2673. AAAI Press/IJCAI.

Matentzoglu, N.; Tang, D.; Parsia, B.; and Sattler, U. 2014. The Manchester OWL repository: System description. In *Proc. ISWC'2014 (Posters & Demonstrations)*, volume 1272 of *CEUR Workshop Proceedings*, 285–288. CEUR-WS.org.

Motik, B.; Nenov, Y.; Piro, R.; Horrocks, I.; and Olteanu, D. 2014. Parallel materialisation of Datalog programs in centralised, main-memory RDF systems. In *Proc. AAAI'14*, 129–137. AAAI Press.

Motik, B.; Shearer, R.; and Horrocks, I. 2009. Hypertableau reasoning for description logics. *J. of Artificial Intelligence Research* 36:165–228.

Parsia, B.; Matentzoglu, N.; Gonçalves, R. S.; Glimm, B.; and Steigmiller, A. 2017. The OWL reasoner evaluation (ORE) 2015 competition report. *J. of Autom. Reasoning* 59(4):455–482.

Sirin, E.; Parsia, B.; Grau, B. C.; Kalyanpur, A.; and Katz, Y. 2007. Pellet: A practical OWL-DL reasoner. *J. of Web Semantics* 5(2):51–53.

Steigmiller, A.; Liebig, T.; and Glimm, B. 2014. Konclude: System description. *J. of Web Semantics* 27:78–85.

Tsarkov, D., and Horrocks, I. 2006. FaCT++ description logic reasoner: System description. In *Proc. IJCAR'06*, volume 4130 of *Lecture Notes in Computer Science*, 292–297. Springer.

Vrandečić, D., and Krötzsch, M. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM* 57(10).

Zhou, Y.; Cuenca Grau, B.; Nenov, Y.; Kaminski, M.; and Horrocks, I. 2015. PAGOdA: Pay-as-you-go ontology query answering using a Datalog reasoner. *J. of Artif. Intell. Res.* 54:309–367.