Ontology-based Data Access: Theory and Practice

Guohui Xiao

KRDB Research Centre

Free University of Bozen-Bolzano

Roman Kontchakov

Department of Computer Science & Inf. Systems

Birkbeck, University of London

http://ontop.inf.unibz.it/ijcai-2018-tutorial

The "simple" axiom

$$\mathsf{CheckingAccount} \sqcap \exists \mathsf{name}.\mathsf{Person} \sqsubseteq \mathsf{SimpleAccount}$$

is NOT in OWL 2 QL.

Recall that OWL 2 QL does not contain:

- **existential quantification** (ObjectSomeValuesFrom) **on the left**
- **conjunction** (intersectionOf) **on the left**
- **universal quantification** (ObjectAllValuesFrom, DataAllValuesFrom)
- **enumeration** of individuals and literals (ObjectOneOf, DataOneOf)
- **disjunction** (ObjectUnionOf, DisjointUnion and DataUnionOf)
- individual equality assertions (**sameAs**) (SameIndividual)
- ...

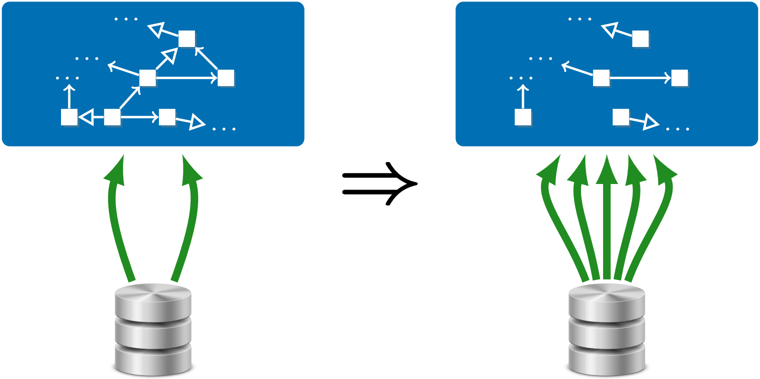
The "simple" axiom

$$\mathsf{CheckingAccount} \sqcap \exists \mathsf{name}.\mathsf{Person} \sqsubseteq \mathsf{SimpleAccount}$$

is NOT in OWL 2 QL.

Recall that OWL 2 QL does not contain:

- **existential quantification** (ObjectSomeValuesFrom) **on the left**
- **conjunction** (intersectionOf) **on the left**
- **universal quantification** (ObjectAllValuesFrom, DataAllValuesFrom)
- **enumeration** of individuals and literals (ObjectOneOf, DataOneOf)
- **disjunction** (ObjectUnionOf, DisjointUnion and DataUnionOf)
- individual equality assertions (**sameAs**) (SameIndividual)
- ...



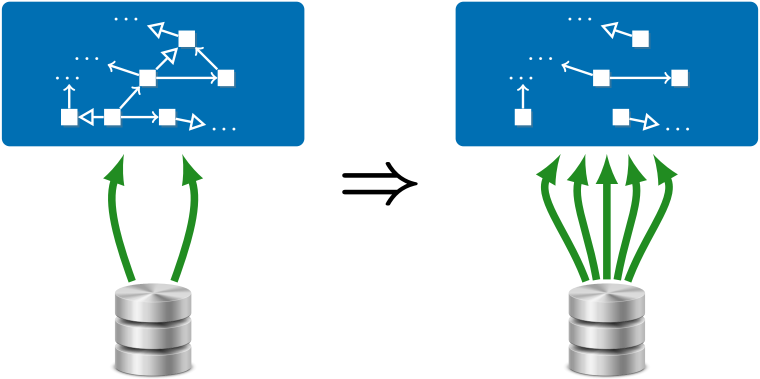
$\langle \mathcal{T}, \mathcal{M}, \mathcal{S} \rangle \Rightarrow \langle \mathcal{T}', \mathcal{M}', \mathcal{S} \rangle$

Rewriting The new specification is equivalent to the original one w.r.t. query answering (**query-inseparable**).

Approximation The new specification is a sound approximation of the original one w.r.t. query answering.

NB: This idea is inspired by the mapping saturation technique

Framework for Rewriting and Approximation of OBDA [Botoeva et al. 2016]



$\langle \mathcal{T}, \mathcal{M}, \mathcal{S} \rangle$ $\qquad\qquad$ $\langle \mathcal{T}', \mathcal{M}', \mathcal{S} \rangle$

Rewriting The new specification is equivalent to the original one w.r.t. query answering (**query-inseparable**).

Approximation The new specification is a sound approximation of the original one w.r.t. query answering.

NB: This idea is inspired by the mapping saturation technique

Example: compiling ontology constraints into the mapping

Example

$$\mathcal{T} = \{\ A \sqcap B \sqsubseteq C\ \}$$
$$\mathcal{M} = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x),\ \mathsf{SQL}_B(x) \rightsquigarrow B(x)\ \}$$
$$\Rightarrow$$
$$\mathcal{T}' = \{\ \ \}$$
$$\mathcal{M}' = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x),\ \mathsf{SQL}_B(x) \rightsquigarrow B(x),\ \mathsf{SQL}_A(x) \land \mathsf{SQL}_B(x) \rightsquigarrow C(x)\}$$

Example

$$\mathcal{T} = \{\ \exists R.A \sqsubseteq C\ \}$$
$$\mathcal{M} = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x),\ \mathsf{SQL}_R(x,y) \rightsquigarrow R(x,y)\ \}$$
$$\Rightarrow$$
$$\mathcal{T}' = \{\ \ \}$$
$$\mathcal{M}' = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x),\ \mathsf{SQL}_R(x,y) \rightsquigarrow R(x,y),\ \mathsf{SQL}_R(x,y) \land \mathsf{SQL}_A(y) \rightsquigarrow C(x)\}$$

Example: compiling ontology constraints into the mapping

Example

$$\mathcal{T} = \{\ A \sqcap B \sqsubseteq C\ \} \qquad \Rightarrow \qquad \mathcal{T}' = \{\ \ \}$$

$$\mathcal{M} = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x),\ \mathsf{SQL}_B(x) \rightsquigarrow B(x)\ \} \qquad \Rightarrow \qquad \mathcal{M}' = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x),\ \mathsf{SQL}_B(x) \rightsquigarrow B(x),\ \mathsf{SQL}_A(x) \land \mathsf{SQL}_B(x) \rightsquigarrow C(x)\}$$

Example

$$\mathcal{T} = \{\ \exists R.A \sqsubseteq C\ \} \qquad \Rightarrow \qquad \mathcal{T}' = \{\ \ \}$$

$$\mathcal{M} = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x),\ \mathsf{SQL}_R(x,y) \rightsquigarrow R(x,y)\ \} \qquad \Rightarrow \qquad \mathcal{M}' = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x),\ \mathsf{SQL}_R(x,y) \rightsquigarrow R(x,y),\ \mathsf{SQL}_R(x,y) \land \mathsf{SQL}_A(y) \rightsquigarrow C(x)\}$$

Example

$$\begin{array}{ll} \mathcal{T} = \{\ A \sqcap B \sqsubseteq C\ \} & \mathcal{T}' = \{\ \ \} \\ \mathcal{M} = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x), & \Rightarrow\ \mathcal{M}' = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x), \\ \qquad \mathsf{SQL}_B(x) \rightsquigarrow B(x)\ \} & \qquad \mathsf{SQL}_B(x) \rightsquigarrow B(x), \\ & \qquad \mathsf{SQL}_A(x) \wedge \mathsf{SQL}_B(x) \rightsquigarrow C(x)\} \end{array}$$

Example

$$\begin{array}{ll} \mathcal{T} = \{\ \exists R.A \sqsubseteq C\ \} & \mathcal{T}' = \{\ \ \} \\ \mathcal{M} = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x), & \Rightarrow\ \mathcal{M}' = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x), \\ \qquad \mathsf{SQL}_R(x,y) \rightsquigarrow R(x,y)\ \} & \qquad \mathsf{SQL}_R(x,y) \rightsquigarrow R(x,y), \\ & \qquad \mathsf{SQL}_R(x,y) \wedge \mathsf{SQL}_A(y) \rightsquigarrow C(x)\} \end{array}$$

Example: compiling ontology constraints into the mapping

Example

$$\begin{array}{l} \mathcal{T} = \{\ A \sqcap B \sqsubseteq C\ \} \\ \mathcal{M} = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x), \\ \qquad \mathsf{SQL}_B(x) \rightsquigarrow B(x)\ \} \end{array} \Rightarrow \begin{array}{l} \mathcal{T}' = \{\ \ \} \\ \mathcal{M}' = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x), \\ \qquad \mathsf{SQL}_B(x) \rightsquigarrow B(x), \\ \qquad \mathsf{SQL}_A(x) \wedge \mathsf{SQL}_B(x) \rightsquigarrow C(x) \} \end{array}$$

Example

$$\begin{array}{l} \mathcal{T} = \{\ \exists R.A \sqsubseteq C\ \} \\ \mathcal{M} = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x), \\ \qquad \mathsf{SQL}_R(x,y) \rightsquigarrow R(x,y)\ \} \end{array} \Rightarrow \begin{array}{l} \mathcal{T}' = \{\ \ \} \\ \mathcal{M}' = \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x), \\ \qquad \mathsf{SQL}_R(x,y) \rightsquigarrow R(x,y), \\ \qquad \mathsf{SQL}_R(x,y) \wedge \mathsf{SQL}_A(y) \rightsquigarrow C(x) \} \end{array}$$

Example: Recursion

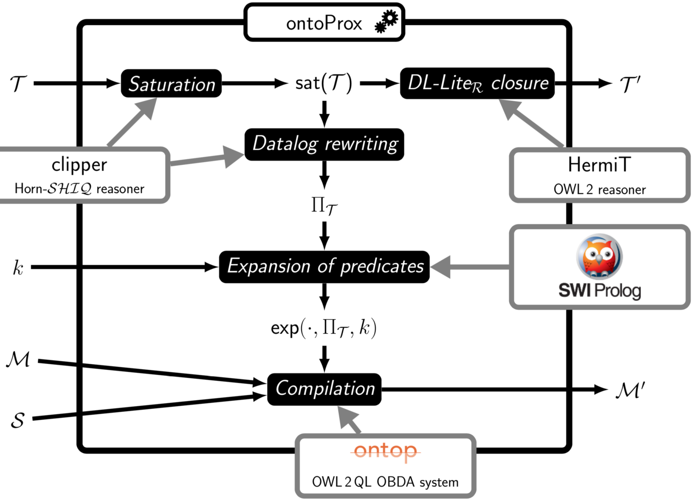
Recursion cannot be fully captured via the mapping.
$\rightsquigarrow$ We use approximation, by setting **a bound on the depth** of the Datalog expansion of queries.

Example

$$
\begin{aligned}
\mathcal{T} &= \{\ \exists R.A \sqsubseteq A\ \} \\
\mathcal{M} &= \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x), \\
&\qquad \mathsf{SQL}_R(x,y) \rightsquigarrow R(x,y)\ \}
\end{aligned}
$$

$\Rightarrow$

$$
\begin{aligned}
\mathcal{T}' &= \{\ \ \} \\
\mathcal{M}' &= \{\ \mathsf{SQL}_A(x) \rightsquigarrow A(x), \\
&\qquad \mathsf{SQL}_R(x,y) \rightsquigarrow R(x,y), \\
&\qquad \mathsf{SQL}_R(x,y) \land \mathsf{SQL}_A(y) \rightsquigarrow A(x) \\
&\qquad \mathsf{SQL}_R(x,y) \land \mathsf{SQL}_R(y,z) \land \mathsf{SQL}_A(z) \rightsquigarrow A(x) \\
&\qquad \mathsf{SQL}_R(x,y) \land \mathsf{SQL}_R(y,z) \land \mathsf{SQL}_R(z,w) \land \mathsf{SQL}_A(w) \rightsquigarrow A(x) \\
&\qquad \ldots\ \}
\end{aligned}
$$



Ontoprox: $\langle \mathcal{T}, \mathcal{M}, \mathcal{S} \rangle, k \;\rightarrow\; \langle \mathcal{T}', \mathcal{M}', \mathcal{S} \rangle$
where $\mathcal{T}$ is a Horn-$\mathcal{SHIQ}$ ontology and $\mathcal{T}'$ is a OWL 2 QL ontology.
`https://github.com/ontop/ontoprox`

[1] E. Botoeva et al. "Beyond OWL 2 QL in OBDA: Rewritings and Approximations". In: *Proc. of AAAI*. 2016, pp. 921–928.