Rewriting the Description Logic ALCHIQ to Disjunctive Existential Rules

David Carral and Markus Krötzsch

Knowledge-Based Systems Group

September 14, 2020

Check out the chat for a link to download the slides!
Rewritings

Definition

Consider fragments $\mathcal{L}$ and $\mathcal{L}'$ of FOL. An $\mathcal{L}'$-theory $\mathcal{T}'$ is a rewriting of an $\mathcal{L}$-theory $\mathcal{T}$ if, for all fact sets $\mathcal{F}$ over the signature of $\mathcal{T}$, we have that $\mathcal{T} \cup \mathcal{F}$ and $\mathcal{T}' \cup \mathcal{F}$ are equisatisfiable. If we can always compute such a rewriting, $\mathcal{L}$ is rewriteable to $\mathcal{L}'$.

Motivation

- Theoretical: understand the expressivity of FOL fragments.
- Practical: reuse existing reasoners across FOL fragments.
  - Assume that $\mathcal{L}$ is rewriteable to $\mathcal{L}'$.
  - Consider $\mathcal{T} \cup \mathcal{F}$ with $\mathcal{T}$ an $\mathcal{L}$-theory and $\mathcal{F}$ a fact set.
  - Compute an equisatisfiable theory $\mathcal{T}' \cup \mathcal{F}$ with $\mathcal{T}' \in \mathcal{L}'$.
  - Use an $\mathcal{L}'$-reasoner to decide if $\mathcal{T}' \cup \mathcal{F}$ is satisfiable.
  - The result determines whether $\mathcal{T} \cup \mathcal{F}$ is satisfiable.
**Rewritings**

**Definition**
Consider fragments $\mathcal{L}$ and $\mathcal{L}'$ of FOL. An $\mathcal{L}'$-theory $\mathcal{T}'$ is a rewriting of an $\mathcal{L}$-theory $\mathcal{T}$ if, for all fact sets $\mathcal{F}$ over the signature of $\mathcal{T}$, we have that $\mathcal{T} \cup \mathcal{F}$ and $\mathcal{T}' \cup \mathcal{F}$ are equisatisfiable. If we can always compute such a rewriting, $\mathcal{L}$ is rewriteable to $\mathcal{L}'$.

**Motivation**
- Theoretical: understand the expressivity of FOL fragments.
- Practical: reuse existing reasoners across FOL fragments.

**Contribution**
Establish that $\textsc{ALCHIQ}$ is rewriteable into rule-based languages.
**Definition: ALC\textsc{HQ}**

<table>
<thead>
<tr>
<th>Inference Rule</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \sqcap B \sqsubseteq C$</td>
<td>$A(x) \land B(x) \rightarrow C(x)$</td>
</tr>
<tr>
<td>$A \sqsubseteq B \sqcup C$</td>
<td>$A(x) \rightarrow B(x) \lor C(x)$</td>
</tr>
<tr>
<td>$A \sqsubseteq \forall ! R . B$</td>
<td>$A(x) \land R(x, y) \rightarrow B(y)$</td>
</tr>
<tr>
<td>$A \sqsubseteq \exists ! R . B$</td>
<td>$A(x) \rightarrow \exists y . R(x, y) \land B(y)$</td>
</tr>
<tr>
<td>$A \sqsubseteq \leq_{1} R . B$</td>
<td>$A(x) \land R(x, y) \land B(y) \land R(x, z) \land B(z) \rightarrow y \approx z$</td>
</tr>
<tr>
<td>$R \sqcap S \sqsubseteq V$</td>
<td>$R(x, y) \land S(x, y) \rightarrow V(x, y)$</td>
</tr>
<tr>
<td>$R \sqsubseteq S \sqcup V$</td>
<td>$R(x, y) \rightarrow S(x, y) \lor V(x, y)$</td>
</tr>
<tr>
<td>$R^{-} \sqsubseteq S$</td>
<td>$R(y, x) \rightarrow S(x, y)$</td>
</tr>
</tbody>
</table>

In the above, $A$, $B$, and $C$ are unary predicates (i.e., concept names) and $R$, $S$, and $V$ are binary predicates (i.e., role names).
Datalog $\lor \exists$: Syntax and Semantics

**Definition**

A disjunctive existential rule is a FOL formula of the form

$$\forall \bar{x}. \left( \beta[\bar{x}] \rightarrow \bigvee_{i=1}^{n} \exists \bar{y}_i \cdot \eta_i[\bar{x}_i, \bar{y}_i] \right).$$

where $\beta[\bar{x}]$ and $\eta_i[\bar{x}_i, \bar{y}_i]$ are atom conjunctions using variables in the lists $\bar{x}_i$ and $\bar{y}_i$, such that $\bar{x}_i \subseteq \bar{x}$ and $\bar{x} \cap \bar{y}_i = \emptyset$ for all $1 \leq i \leq n$.

**Definition**

- **Datalog $\lor \exists$**: all sets of disjunctive existential rules.
- **Datalog $\exists$**: Datalog $\lor \exists$ without disjunction.
- **Datalog $\lor$**: Datalog $\lor \exists$ without existential quantifiers.
- **Datalog**: Datalog $\lor$ without disjunctions.
### Rewritings of DL-Type Logics to Rule Languages

<table>
<thead>
<tr>
<th>Reference</th>
<th>Logic</th>
<th>Rule Language</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Hustadt et al., 2007]</td>
<td>ALCHIQ</td>
<td>Datalog(^\lor) exp. †</td>
<td></td>
</tr>
<tr>
<td>[Eiter et al., 2012]</td>
<td>Horn-SHIQ</td>
<td>Datalog exp. †</td>
<td></td>
</tr>
<tr>
<td>[Rudolph et al., 2012]</td>
<td>SHIQ(_b_s)</td>
<td>Datalog(^\lor) exp. †</td>
<td></td>
</tr>
<tr>
<td>[Bienvenu et al., 2014]</td>
<td>SHIQ</td>
<td>Datalog(^\lor) exp. †</td>
<td></td>
</tr>
<tr>
<td>[Carral et al., 2018]</td>
<td>Horn-ALC（带）</td>
<td>Datalog exp. †</td>
<td></td>
</tr>
<tr>
<td>[Carral et al., 2019b]</td>
<td>Horn-SHIQ</td>
<td>Datalog exp. †</td>
<td></td>
</tr>
<tr>
<td>[Ortiz et al., 2010]</td>
<td>Horn-ALC（带）</td>
<td>Datalog poly.</td>
<td></td>
</tr>
<tr>
<td>[Ahmetaj et al., 2016]</td>
<td>Alc（带）</td>
<td>Datalog(^\lor) poly.</td>
<td></td>
</tr>
<tr>
<td>[Krötzsch, 2011]</td>
<td>EL(^++)</td>
<td>Datalog poly. †</td>
<td></td>
</tr>
<tr>
<td>[Carral et al., 2019a]</td>
<td>Horn-ALC</td>
<td>Datalog(^3) poly. †</td>
<td></td>
</tr>
</tbody>
</table>

†: rules of bounded size that does not depend on input

### Remark

All rewriting techniques for expressive DLs produce rule sets of exponential size or rules of unbounded arity.
**Theorem 1**

\(\text{ALCHIQ}\) is poly-time rewritable into terminating Datalog\(^{\lor \exists}\) rules \textbf{of bounded size}.

**Definition: Terminating Datalog\(^{\lor \exists}\)**

Language of all sets \(R\) of disjunctive existential rules that terminate with respect to the Datalog-first restricted chase.

**Theorem 2**

\(\text{ALCHIQ}\) is poly-time rewritable to Datalog\(^{\lor}\) rules (of unbounded size).
**Results**

**Simplified Theorem**

\( \mathcal{ALC} \) is poly-time rewritable into terminating Datalog\( ^{\vee \exists} \) rules of bounded size.

**Definition: \( \mathcal{ALC} \)**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A \sqcap B \sqsubseteq C )</td>
<td>( A(x) \land B(x) \rightarrow C(x) )</td>
</tr>
<tr>
<td>( A \sqsubseteq B \sqcup C )</td>
<td>( A(x) \rightarrow B(x) \lor C(x) )</td>
</tr>
<tr>
<td>( A \sqsubseteq \forall R.B )</td>
<td>( A(x) \land R(x, y) \rightarrow B(y) )</td>
</tr>
<tr>
<td>( A \sqsubseteq \exists R.B )</td>
<td>( A(x) \rightarrow \exists y. R(x, y) \land B(y) )</td>
</tr>
</tbody>
</table>

In the above, \( A, B, \) and \( C \) are unary predicates (i.e., concept names) and \( R \) is a binary predicate (i.e., role name)
Definition: $\mathcal{ALC}$ Rewritings

Consider a theory $T$ of $\mathcal{ALC}$ axioms and a sequence $A_1, \ldots, A_n$ containing all of the classes in $T$. Then, the following set of Datalog $v^3$ rules is a terminating rewriting for $T$:

$$
\{A(x) \land B(x) \rightarrow C(x) \mid A \sqcap B \sqsubseteq C \in T\} \cup \{A(x) \rightarrow B(x) \lor C(x) \mid A \sqsubseteq B \sqcup C \in T\} \cup \\
\{A(x) \land R(x, y) \rightarrow B(y) \mid A \sqsubseteq \forall R.B \in T\} \cup \\
\{A(x) \rightarrow \exists y.R(x, y) \land B(y) \land \text{Succ}(x, y) \mid A \sqsubseteq \exists R.B \in T\} \cup \\
\{\rightarrow \triangleleft A(x) \lor A^\neg(x), \rightarrow \neg A(x) \land A^\neg(x) \rightarrow \bot \mid A \in \text{Classes}(T)\} \cup \\
\{A_1(x) \land A_1(z) \rightarrow \text{SameClasses}_{1}(x, z), A_1^\neg(x) \land A_1^\neg(z) \rightarrow \text{SameClasses}_{1}(x, z)\} \cup \\
\{\text{SameClasses}_{i-1}(x, z) \land A_i(x) \land A_i(z) \rightarrow \text{SameClasses}_{i}(x, z), \\
\quad \text{SameClasses}_{i-1}(x, z) \land A_i^\neg(x) \land A_i^\neg(z) \rightarrow \text{SameClasses}_{i}(x, z) \mid 2 \leq i \leq n\} \cup \\
\{\text{SameClasses}_{n}(x, y) \rightarrow \text{SameType}(x, y)\} \cup \\
\{\text{SameType}(x, z) \land \text{Succ}(x, y) \land R(x, y) \rightarrow \text{Succ}(z, y) \land R(z, y) \mid R \in \text{Roles}(T)\}
$$
Thank you for your attention!

Link to the paper:
iccl.inf.tu-dresden.de/web/Inproceedings3244/en


CARRAL, D., GONZÁLEZ, L., AND KOOPMANN, P. (2019B). **From Horn-SRIQ to Datalog: A data-independent transformation that preserves assertion entailment.**

