Towards Polynomial-Time Forgetting and Query Rewriting

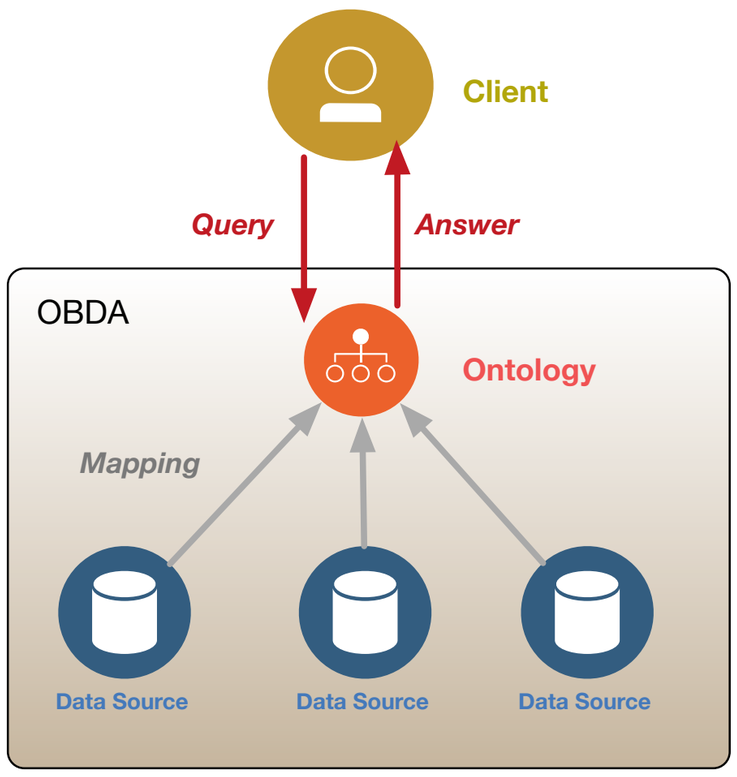
Student: Sen Zheng Supervisor: Renate A.Schmidt

Research Group: Formal Methods Group

Introduction

- Query rewriting. Rewrite queries against an ontology is widely studied in ontology-based data access (OBDA) systems [4]. The recent research focus is on:
  - what kind of query?
  - how to rewrite?
  - what kind of ontologies?

  Our research shows that it has **polynomial-time** complexity to use an **Ackermann's Lemma(AL)**-based forgetting approach to rewrite **role symbols** using **instance queries** against **non-nested ontologies** of $\mathcal{ALCOI}$.



- Forgetting guard predicates in the guarded fragment.
  As a translation of $\mathcal{ALCOI}$ in the first-order logic, the **guarded fragment (GF)**'s guard predicates can not be rewritten (forgotten) [2].

  Motivated by [3] and [5], we show that the guard predicates can be forgotten without losing semantic equivalence.

  The result is a more generalised result of the previous query rewriting for $\mathcal{ALCOI}$.

An example

Knowledge base(KB):
Postdoc $\sqsubseteq$ Researcher
Researcher $\sqsubseteq \exists$worksFor

KB in first-order logic:
Postdoc(x) $\rightarrow$ Researcher(x)
Researcher(x) $\rightarrow \exists y$worksFor(x, y)

Data source:
worksFor(Alice, WebCure)
worksFor(Bob, AniFur)
Postdoc(Dva)
Researcher(Cook)

Query:
Who works for any project? $q(x) = \exists y \text{worksFor}(x, y)$
{Alice, Bob} is a direct answer set to this query.

Our approach:
Rewrite $\exists y \text{worksFor}(x, y)$ against the knowledge base:

KB:	$\neg Postdoc(x) \vee Researcher(x)$	1
KB:	$\neg Researcher(x) \vee \exists y\, worksFor(x, y)$	2
q:	$\neg worksFor(x, y)$	3
AL on 2, 3:	$\neg Researcher(x)$	4
AL on 1, 4:	$\neg Postdoc(x)$	5

4 and 5 can be derived from the given query q and the knowledge base KB. 4 can be seen as a query $q_1(x) = \text{Researcher}(x)$ and 5 is $q_2(x) = \text{Postdoc}(x)$.

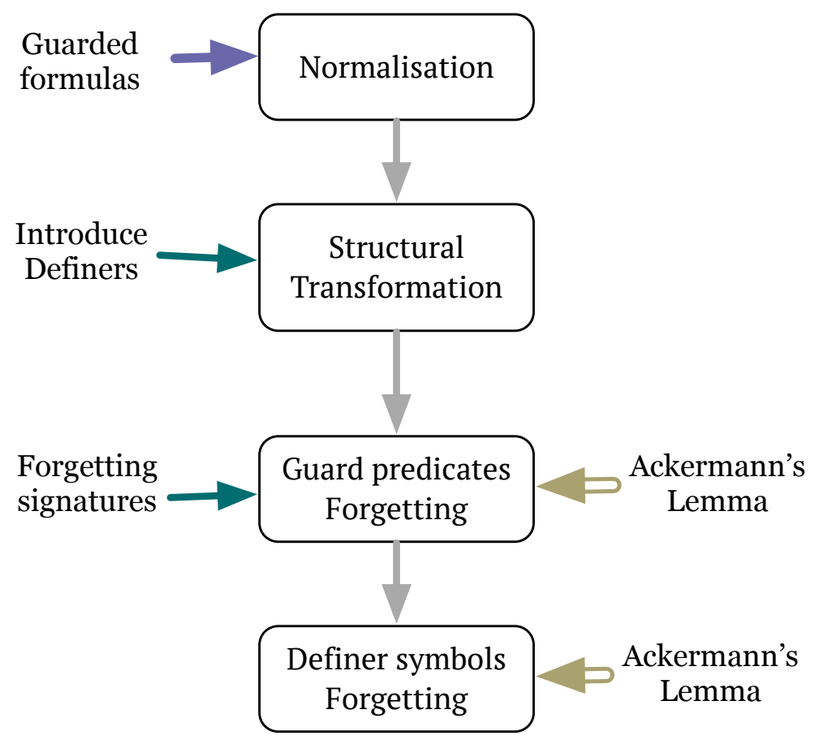
Having q_1 and q_2, Cook and Dva will be included in the final answer set {Alice, Bob, Cook, Dva}.

Forgetting guard predicates in GF

- **Input**: A set N_0 of non-nested guarded formulas.
- **Assumptions**:
  - Guard predicates do not occur at non-guard positions.
  - Equality and constants are allowed.
  - The forgetting symbol $\mathcal{F}$ only contains guard predicates.
- **Target**: Forget guard predicates in the non-nested guarded formulas.
- **Steps**:
  - Normalisation: Universal quantifiers are added for free variables in N_0. The formulas in N_0 are then transformed into a set N_1 of their negation normal forms.
  - Structural transformation: New predicates (definers) are introduced to transform different formulas into a set N_2.
    In particular, for constants and equalities, we introduce the extended term abstraction rule as follows.

$$\frac{N \cup \{C(x, a)\}}{N \cup \{C(x, y) \vee Q(y), \neg Q(a)\}}$$

    where y is a fresh variable and Q is a fresh predicate.
  - Forgetting guard predicates: Ackermann's Lemma is used for N_2 to eliminate guard predicates in $\mathcal{F}$ one at a time.
  - Eliminating definers: Ackermann's Lemma is also used to eliminate the definers introduced by structural transformation.



Conclusion and Ongoing Work

- Forgetting techniques can be used to rewrite instance queries and to forget guard predicates in the non-nested guarded fragment.
- Now we are working on proofs of soundness and forgetting completeness of this approach. The next step will be evaluating our query rewriting approach and looking for other possible applications.

References

Andréka, Hajnal and Németi, István and van Benthem, Johan
Modal languages and bounded fragments of predicate logic
Journal of Philosophical Logic

Hoogland, Eva and Marx, Maarten
Interpolation and definability in guarded fragments
Studia Logica

Zhao, Yizheng and Schmidt, Renate A
Role Forgetting for $\mathcal{ALCOQH}$(universal role)-Ontologies Using an Ackermann-Based Approach
Proceedings of the 26th International Joint Conference on Artificial Intelligence

Rodriguez-Muro, Mariano and Kontchakov, Roman and Zakharyaschev, Michael
International Semantic Web Conference
Springer

Koopmann, Patrick and Schmidt, Renate A
Forgetting Concept and Role Symbols in $\mathcal{ALCH}$-Ontologies
Logic for Programming, Artificial Intelligence, and Reasoning LNCS 8312