Saturation-Based Querying Procedures for the Clique-Guarded Negation Fragment

Sen Zheng, Renate A. Schmidt
Department of Computer Science, The University of Manchester, UK

***Abstract*—The clique-guarded negation fragment (CGNF) is one of the most expressive decidable fragments in first-order logic. This paper considers two types of querying for the CGNF: i) Boolean conjunctive query (BCQ) answering and a novel notion of ii) saturation-based BCQ rewriting. Though BCQ answering for CGNF is decidable, insufficient effort has been devoted to developing automated deduction methods for this problem. We close this gap by providing saturation-based decision procedures for both types of querying for CGNF. Unlike conventional query rewriting, saturation-based query rewriting compiles a saturation of the schema and query, which has the advantage that the saturation is reusable with heterogeneous data sources, enabling distributed querying and is well-suited for use cases of ontology-based data access.**

***Index Terms*—Boolean conjunctive querying, saturation-based query rewriting, guarded negation fragments, saturation-based decision procedure**

I. Introduction

As a generalisation of modal logics to first-order logic (FOL) [1, 8], the guarded fragment (GF) inherits both robust decidability [9] from modal logics and expressiveness of quantifications from FOL. By now GF has been generalised to a large family of decidable fragments in FOL, including the *guarded quantification fragments* (i.e., GF, the loosely guarded fragment (LGF) and the clique guarded fragment (CGF) [1, 6, 8]) and the *guarded negation fragments* (i.e., the unary negation fragment (UNF), the guarded negation fragment (GNF) and the *clique-guarded negation fragment (CGNF)* [4, 5, 7]). Fig. 1 depicts the relationship among these fragments and FOL. In this paper, we focus on the CGNF as it subsumes the aforementioned guarded fragments.

In this paper we will use the following notation: Σ for formulas in CGNF, D for ground atoms, and q for a union of Boolean conjunctive queries (BCQs). The problems of BCQ answering for CGNF and saturation-based BCQ rewriting for CGNF, respectively, are:

Question I. *Does a* saturation-based *procedure exist that decides whether* $\Sigma \cup D \models q$*?*

Question II. *Let* N *be a saturation of* $\{\neg q\} \cup \Sigma$*. Can* N *be* back-translated *to a (Skolem-symbol-free) first-order formula* Σ_q *such that* $D \cup \Sigma \models q$ *iff* $D \models \Sigma_q$*?*

Our saturation-based decision procedure extends and improves the decision procedures for the guarded quantification fragments [10, 11].

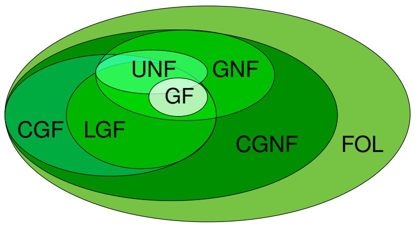

Fig. 1. The relationship among the guarded fragments and FOL

II. BCQ answering procedure

Fig. 2 depicts how our procedure answers Question I. It proceeds in three main steps, see Fig. 2:

1) **Trans**: A clausification process that reduces checking entailment of $\Sigma \cup D \models q$ to checking unsatisfiability of a set of *loosely guarded clauses with equality (*$LG_{\approx}$ *clauses)* and *query clauses*.
2) **Q-Sep**: Use *separation rules* to simplify *inequality-free query clauses* by $LG_{\approx}$ clauses and *indecomposable chained-only query clauses (ICQ clauses)*.

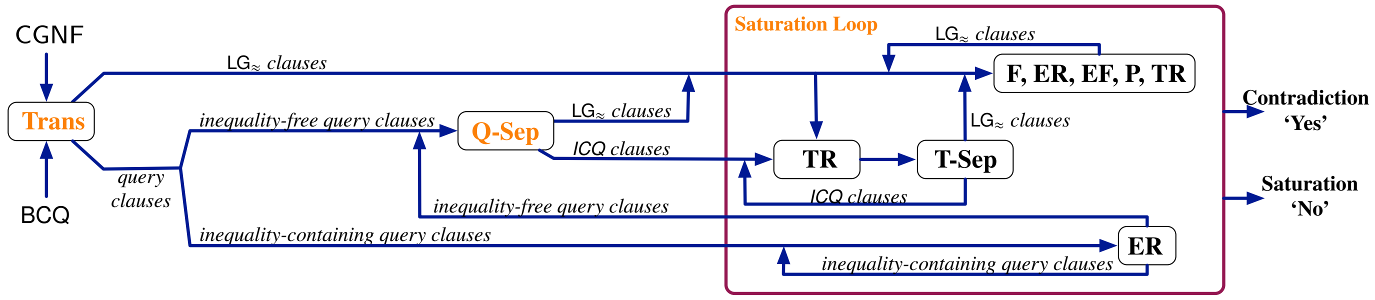

Fig. 2. Our saturation-based query answering procedure for CGNF with rules and input/output clauses specified

Let $C_1 = B_1 \vee D_1, \ldots, C_n = B_n \vee D_n$ be *side premises* and $C = \neg A_1 \vee \ldots \vee \neg A_n \vee\ D$ a *main premise* with $\neg A_1, \ldots, \neg A_n$ selected in a selection-based resolution step. *Top-variable resolution* is applied as follows.

1) Without producing or adding the resolvent, compute an mgu σ' for $C_1, \ldots, C_n$ and C such that $\sigma' = \text{mgu}(A_1 \doteq B_1, \ldots, A_n \doteq B_n)$.
2) Compute the *variable ordering* $>_v$ and $=_v$ over the variables of $\neg A_1 \vee \ldots \vee \neg A_n$. By definition, $x >_v y$ and $x =_v y$ with respect to σ', if $\text{dep}(x\sigma') > \text{dep}(y\sigma')$ and $\text{dep}(x\sigma') = \text{dep}(y\sigma')$, respectively.
3) The maximal variables under $>_v$ and $=_v$ in $\neg A_1 \vee \ldots \vee \neg A_n$ are the *top variables*. A literal in $\neg A_1, \ldots, \neg A_n$ is called a *top-variable literal* if the literal contains at least one top variable.

Then, the top-variable resolvents are computed using:

$$\textbf{TR}: \ \frac{B_1 \vee D_1, \ldots, B_n \vee D_n, \quad \neg A_1 \vee \ldots \vee \neg A_n \vee D}{(D_1 \vee \ldots \vee D_m \vee \neg A_{m+1} \vee \ldots \vee \neg A_n \vee D)\sigma}$$

if the following conditions are satisfied.

1. $B_1, \ldots, B_n$, respectively, are strictly maximal with respect to $D_1, \ldots, D_n$.

2a. If $n = 1$, then i) either $\neg A_1$ is selected, or nothing is selected in $\neg A_1 \vee D$ and $\neg A_1$ is maximal with respect to D, and ii) $\sigma = \text{mgu}(A_1 \doteq B_1)$, or

2b. there exist an mgu σ' such that $\sigma' = \text{mgu}(A_1 \doteq B_1, \ldots, A_n \doteq B_n)$, then $\neg A_1, \ldots, \neg A_m$ are the *top-variable literals* and $\sigma = \text{mgu}(A_1 \doteq B_1, \ldots, A_m \doteq B_m)$ where $1 \leq m \leq n$.

3. All premises are variable disjoint.

A clause is *decomposable* if it can be divided into two variable-disjoint subclauses, otherwise, the clause is *indecomposable*. Let C be a clause. A literal L is a *surface literal* in C if there exists no other literal L' in C such that $\text{var}(L) \subset \text{var}(L')$. Let L_1 and L_2 be two surface literals in C such that $\text{var}(L_1) \neq \text{var}(L_2)$. Then, x is a *chained variable* in C if $x \in \text{var}(L_1) \cap \text{var}(L_2)$. The other non-chained variables in C are the *isolated variables*.
A clause can be separated using:

$$\textbf{SepDe}: \ \frac{N \cup \{C \vee D\}}{N \cup \{C \vee \neg p_1, \neg p_2 \vee D, p_1 \vee p_2\}}$$

if the following conditions are satisfied.

1) $C \vee D$ is a decomposable clause.
2) C and D are non-empty subclauses of $C \vee D$.
3) $\text{var}(C) \cap \text{var}(D) = \emptyset$.
4) p_1 and p_2 are propositional variables that do not occur in $N \cup \{C \vee D\}$, and p_1 and p_2 are smaller than the predicate symbols in $N \cup \{C \vee D\}$.

$$\textbf{SepIn}: \ \frac{N \cup \{C \vee \neg A(\bar{x}, \bar{y}) \vee D\}}{N \cup \{C \vee \neg A(\bar{x}, \bar{y}) \vee P(\bar{x}), \neg P(\bar{x}) \vee D\}}$$

if the following conditions are satisfied.

1) $C \vee \neg A(\bar{x}, \bar{y}) \vee D$ is an indecomposable clause.
2) $\neg A(\bar{x}, \bar{y})$ is a surface literal and $\text{var}(C) \subseteq \bar{x} \cup \bar{y}$.
3) $\bar{x}$ are chained variables, $\bar{x} \neq \emptyset$, and $\bar{x} \subseteq \text{var}(D)$.
4) $\bar{y}$ are isolated variables and $\bar{y} \cap \text{var}(D) = \emptyset$.
5) P is a predicate symbol that does not occur in $N \cup \{C \vee \neg A(\bar{x}, \bar{y}) \vee D\}$, and P is smaller than the predicate symbols in $N \cup \{C \vee \neg A(\bar{x}, \bar{y}) \vee D\}$.

Fig. 3. The top-variable resolution rule **TR** (on the left) and the separation rules **SepDe** and **SepIn** (on the right)

3) The saturation loop saturates $\mathsf{LG}_{\approx}$ clauses, ICQ clauses and *inequality-containing query clauses*.

Our procedure uses three customised rules, namely the top-variable resolution rule **TR** and the separation rules **SepDe** and **SepIn**. **TR** is an on-the-fly macro resolution rule that aims to avoid term depth increase in the resolvents. This is achieved by first identifying the side premises that contain the potentially deepest terms during unification, and then resolving the main premise with only these side premises. **SepDe** and **SepIn** are simplification rules that replace a clause by multiple clauses that contain fewer variables. Fig. 3 describes these rules.

Beside these special rules, our saturation-based system contains the following standard inference rules from the superposition framework of [2, 3]: the factoring rule **F**, the equality resolution rule **ER**, the equality factoring rule **EF** and the paramodulation rule **P**.

Theorem 1. *The saturation-based procedure of Fig. 2 is a decision procedure for answering* BCQ*s for* CGNF*.*

III. Saturation-based BCQ rewriting procedure

The main challenge in answering Question II is to eliminate Skolem symbols in $\mathsf{LG}_{\approx}$ clauses and query clauses. This is solved by three steps:

1) Use *term abstraction rules* to remove constants and duplicate variables in the compound terms of $\mathsf{LG}_{\approx}$ clauses.
2) Align variables in compound terms that are under the same function symbols across all clauses.
3) Unskolemise and then negate the result.

Theorem 2. *Let N be a saturation computed by applying the procedure of Fig. 2 to $\{\neg q\} \cup \Sigma$. Applying the above steps then to N back-translates it to a (Skolem-symbol-free) first-order formula Σ_q such that $D \cup \Sigma \models q$ iff $D \models \Sigma_q$.*

IV. Conclusions

We provide data-independent non-mainstream saturation-based decision procedures for classical querying problems. This makes our methods attractive for the emerging distributed querying scenarios such as ontology-based data access.

References

[1] H. Andréka, I. Németi, and J. van Benthem, “Modal Languages and Bounded Fragments of Predicate Logic,” *J. Philos. Logic*, vol. 27, no. 3, pp. 217–274, 1998.

[2] L. Bachmair and H. Ganzinger, “Equational Reasoning in Saturation-based Theorem Proving,” in *Automated Deduction: A Basis for Applications*, W. Bibel and P. H. Schmitt, Eds. Kluwer, 1998, pp. 353–397.

[3] ——, “Resolution Theorem Proving,” in *Handbook of Automated Reasoning*, J. A. Robinson and A. Voronkov, Eds. Elsevier and MIT Press, 2001, pp. 19–99.

[4] V. Bárány, B. ten Cate, and L. Segoufin, “Guarded Negation,” in *Proc. ICALP’11*, ser. LNCS, vol. 6756. Springer, 2011, pp. 356–367.

[5] V. Bárány, B. ten Cate, and L. Segoufin, “Guarded Negation,” *J. ACM*, vol. 62, no. 3, pp. 22:1–22:26, 2015.

[6] E. Grädel, “Decision Procedures for Guarded Logics,” in *Proc. CADE’16*, ser. LNCS, vol. 1632. Springer, 1999, pp. 31–51.

[7] B. ten Cate and L. Segoufin, “Unary negation,” *Logic Methods Comput. Sci.*, vol. 9, no. 3, 2013.

[8] J. van Benthem, “Dynamic Bits and Pieces,” Univ. Amsterdam, Research Report LP-97-01, 1997.

[9] M. Y. Vardi, “Why is Modal Logic So Robustly Decidable?” in *Proc. DIMACS Workshop’96*. DIMACS/AMS, 1996, pp. 149–183.

[10] S. Zheng and R. A. Schmidt, “Deciding the Loosely Guarded Fragment and Querying Its Horn Fragment Using Resolution,” in *Proc. AAAI’20*. AAAI, 2020, pp. 3080–3087.

[11] ——, “Resolution-based Boolean Conjunctive Query Answering and Rewriting for Guarded Quantification Fragments,” 2023, accepted to J. Autom. Reason. The technical report version is at https://arxiv.org/abs/2208.05365.