



**TECHNISCHE
UNIVERSITÄT
DRESDEN**

Faculty of Computer Science Institute of Artificial Intelligence Knowledge Representation and Reasoning

Second-Order Characterizations of Definientia in Formula Classes

Christoph Wernhard

KRR Report 14-03

Mail to
Technische Universität Dresden
01062 Dresden

Bulk mail to
Technische Universität Dresden
Helmholtzstr. 10
01069 Dresden

Office
Technische Universität Dresden Room 2006
Nöthnitzer Straße 46
01187 Dresden

Internet
<http://www.wv.inf.tu-dresden.de>



Second-Order Characterizations of Definientia in Formula Classes

Christoph Wernhard
Technische Universität Dresden

Abstract. Predicate quantification can be applied to characterize definientia of a given formula that are in terms of a given set of predicates. Methods for second-order quantifier elimination and the closely related computation of forgetting, projection and uniform interpolants can then be applied to compute such definientia. Here we address the question, whether this principle can be transferred to definientia in given classes that allow efficient processing, such as Horn or Krom formulas. Indeed, if propositional logic is taken as basis, for the class of all formulas that are equivalent to a conjunction of atoms and the class of all formulas that are equivalent to a Krom formula, the existence of definientia as well as representative definientia themselves can be characterized in terms of predicate quantification. For the class of formulas that are equivalent to a Horn formula, this is possible with a special further operator. For first-order logic as basis, we indicate guidelines and open issues.

1 Introduction

Tasks in knowledge processing such as view-based query rewriting with exact rewritings [2, 20, 22, 8, 28] involve the *computation of a definiens* R of a given “query” formula Q within a second given “background” formula F , such that R meets certain conditions, for example, that it is expressed in terms of a given set S of predicates. That is, for given Q, F and S , a formula R must be computed, such that $F \models (R \leftrightarrow Q)$ and only predicates from S do occur in R . If the requested property of R is indeed that it involves only predicates from a restricted set, then the class of solution formulas R can be characterized straightforwardly with predicate quantification. This allows to relate the computation of formulas R to the various advances concerning applications of and methods for second-order quantifier elimination and its variants, the computation of forgetting, of projection, and of uniform interpolants, in particular with respect to knowledge representation in first-order logic and description logics [10, 3, 9, 13, 18, 25, 4, 26, 14], and in the preprocessing of propositional formulas [1, 7, 11, 19]. The underlying basic principle is that the second-order formula $\exists P F$, where F is a first-order formula and P is a predicate, is – if it admits elimination of the predicate quantifier – equivalent to a first-order formula that does not involve P but is equivalent to F with respect to the other predicates.

The question addressed here is whether also definientia in given classes of formulas that are typically *characterized by other means than vocabulary restrictions*, such as Horn formulas, conjunctions of atoms or Krom formulas, can be specified with predicate quantification and can thus be computed by second-order quantifier elimination methods. The envisaged main application is to compute such definientia as query rewritings that are in restricted classes which allow further processing in particularly efficient ways or by engines with limited deductive capability. It seems that also the requirement that the *negation* of a

definiens R is in some given formula class can be useful: R might be evaluated by proving that a given knowledge base is unsatisfiable when conjoined with $\neg R$. The case of negated definiens is subsumed by the general case: The requirement that R is a definiens for Q , where $\neg R$ is in some formula class, can be expressed just as the requirement that R' is a definiens in some formula class for $\neg Q$ and letting R be $\neg R'$.

A starting point for the second-order characterizations of the considered formula classes is the semantic characterization of the class of propositional Horn formulas [5, 21]: A propositional formula is equivalent to a Horn formula if and only if it has the *model intersection property*. Literal projection [15, 24] is a generalization of predicate quantification that allows to specify that only positive or negative predicate occurrences are affected. It can be combined with the semantic characterization of Horn formulas to express the requirement that the definiens is a conjunction of atoms. This restriction can be – up to equivalence – characterized purely by second-order operators that can be defined in terms of predicate quantification. It can be applied to express restrictions to further classes as vocabulary restrictions by meta-level encodings. We show this for Krom formulas, which can be represented as conjunctions of “meta-level” atoms of the form $\text{clause}(L, M)$, defined in the background formula with equivalences ($\text{clause}(L, M) \leftrightarrow L \vee M$) for literals L, M of the original vocabulary.

The rest of the paper is structured as follows: The background framework of classical logic extended by second-order operators is introduced in Sect. 2. Then the class of formulas that are equivalent to a Horn formula is considered: A semantic characterization and a notion of approximation, stemming from the literature on knowledge compilation, are rendered, along with a further weaker notion of approximation that is expressible in terms of predicate quantification. In Sect. 4 two further classes are considered, formulas that are equivalent to a conjunction of atoms and formulas that are equivalent to a Krom formula. For all three considered classes, characterizations of definability and of a definiens in the respective class, provided definability holds, are given. For the latter two classes, these characterizations can be expressed by predicate quantification. In the conclusion (Sect. 5), inherent features of the approach are summarized and possible generalizations and ways to implement it are indicated.

2 Classical Logic Extended by Second-Order Operators

2.1 Notation and Preliminaries

Although the envisaged underlying logic is classical first-order logic, we consider here technically just propositional logic as basis, for simplicity of presentation and because the considered formula classes have in propositional logic immediate correspondence to syntactic restrictions. With the first-order generalization in mind, we often speak of *predicates* instead of *atoms*. Actually, the background framework, definitions, propositions and theorems given in the paper transfer to a large extent easily to first-order logic, following the principles in [24, 25, 28]. When there are particularities to consider, these will be discussed.

We assume a fixed finite set of atoms, denoted by **ATOMS**. An *interpretation* is a set of literals that contains each atom either in a positive or negative literal. (This representation of interpretations facilitates the set-oriented specification of

second-order operators shown further down below.) The satisfaction relation \models between interpretations and formulas is defined with a clause for atoms and one for each logical operator. For instance, for all interpretations I , atoms P and formulas F, G , define: $I \models P \text{ iff}_{def} P \in I$; $I \not\models \perp$; $I \models \neg F \text{ iff}_{def} I \not\models F$; $I \models (F \wedge G) \text{ iff}_{def} I \models F \text{ and } I \models G$. Entailment and equivalence are defined as usual: $F \models G \text{ iff}_{def}$ for all interpretations I it holds that if $I \models F$, then $I \models G$; $F \equiv G \text{ iff}_{def} F \models G \text{ and } G \models F$. We call a set of literals a *scope*. The complement of a literal L is denoted by \bar{L} . The set of complement literals of the members of a scope S is denoted by \bar{S} . As special cases of scopes, the sets of all positive and all negative literals are denoted by POS and NEG, respectively.

2.2 Literal Projection

We now add a second-order operator to classical propositional logic, defined semantically by a clause analogously to the standard connectives. The introduced operator has, aside of a formula, a scope as argument, written as subscript:

Definition 1 (Second-Order Operator project). For all interpretations I , scopes S and formulas F define:

$$I \models \text{project}_S(F) \text{ iff}_{def} \text{ There exists an interpretation } J \text{ such that} \\ J \models F \text{ and } J \cap S \subseteq I.$$

The formula $\text{project}_S(F)$ expresses the *literal projection* of formula F onto scope S [15, 24]. It generalizes existential Boolean quantification: If $S = \bar{S}$, then $\text{project}_S(F)$ is equivalent to $\exists P_1, \dots, \exists P_n F$, where P_1, \dots, P_n are those atoms in F that do not occur in S . Literal projection can also express that quantification affects only positive or only negative atom occurrences (considering also implicit negation through implication and biconditional as well as cancellation of negation in even nestings). For example, the “forgetting” just about the negative occurrences of atom P can be expressed as $\text{project}_{(\text{POS} \cup \text{NEG}) \setminus \{\neg P\}}(F)$, which is equivalent to $((P \wedge F) \vee F[P \mapsto \perp])$, where $F[P \mapsto \perp]$ denotes F with all occurrences of P replaced by \perp . The following notation serves to express that a formula is semantically “in” a scope, or, in other words, just “about” literals in a scope:

Definition 2 (\in). For all formulas F and scopes S define:

$$F \in S \text{ iff}_{def} F \equiv \text{project}_S(F).$$

The statement $F \in S$ holds if and only if F is equivalent to a formula in negation normal form whose literals are all in S .

2.3 Definientia and Definability

The second-order operators gsnc and gWSC , defined in the following in terms of projection, express the *globally strongest necessary condition (GSNC)* and the *globally weakest sufficient condition (GWSC)* of formula G on scope S within formula F [25], which are variants of the strongest necessary condition and weakest sufficient condition [17, 6]. As shown in [25], aside of the possibility to differentiate between negative and positive predicate occurrences, the main difference to the variants introduced in [17] is that for a given formula and scope only the “global” variants are unique up to equivalence. This justifies speaking of *the* GSNC and *the* GWSC.

Definition 3 (Second-Order Operators \mathbf{gsnc} , \mathbf{gwsc}). For all scopes S and formulas F, G define:

- (i) $\mathbf{gsnc}_S(F, G) \stackrel{\text{def}}{=} \text{project}_S(F \wedge G)$.
- (ii) $\mathbf{gwsc}_S(F, G) \stackrel{\text{def}}{=} \neg \text{project}_{\bar{S}}(F \wedge \neg G)$.

The GSNC as well as the GWSC on scope S are in scope S :

Proposition 4 (Scope of \mathbf{gsnc} and \mathbf{gwsc}). For all scopes S and formulas F, G it holds that: (i) $\mathbf{gsnc}_S(F, G) \in S$. (ii) $\mathbf{gwsc}_S(F, G) \in S$.

The dual concepts GSNC and GWSC are application patterns of projection that arise in many contexts such as non-monotonic reasoning [25], abductive reasoning [26], and provide a basis to characterize the notions of *definiens* and *definability* [28]:

Definition 5 (Definiens, Definability). For all formulas F, G, H and scopes S define:

- (i) H is a *definiens* of G in terms of S within F iff_{def}
 $\mathbf{gsnc}_S(F, G) \models H \models \mathbf{gwsc}_S(F, G)$ and $H \in S$.
- (ii) G is *definable* in terms of S within F iff_{def}
 $\mathbf{gsnc}_S(F, G) \models \mathbf{gwsc}_S(F, G)$.

This characterization states that definiens are exactly the formulas in the given scope S that are stronger than the GSNC and weaker than the GWSC of the definiendum G on S within the background formula F . It can be shown that for all formulas F, G, H and scopes S it holds that H is a definiens of G in terms of S within F in the sense of the above characterization if and only if $F \models H \leftrightarrow G$ and $H \in S$ [28]. A formula is definable if and only if there exists a definiens of it in terms of the given scope within the given formula.

2.4 Circumscription

We define a further basic second-order operator \mathbf{raise} and on its basis another second-order operator \mathbf{circ} , which allows to express a generalization of predicate circumscription. These operators have been introduced and discussed in more detail in [25]:

Definition 6 (Second-Order Operators \mathbf{raise} , \mathbf{circ}). For all interpretations I , scopes S and formulas F define:

- (i) $I \models \mathbf{raise}_S(F)$ iff_{def} There exists an interpretation J such that
 $J \models F$ and $J \cap S \subset I \cap S$.
- (ii) $\mathbf{circ}_S(F) \stackrel{\text{def}}{=} F \wedge \neg \mathbf{raise}_S(F)$.

As special cases, the models of $\mathbf{circ}_{\text{POS}}(F)$ and $\mathbf{circ}_{\text{NEG}}(F)$ are the minimal and maximal models, respectively, of F , with respect to the following partial order on interpretations:

$$I \leq_{\text{POS}} J \stackrel{\text{def}}{=} I \cap \text{POS} \subseteq J \cap \text{POS}. \quad (\text{i})$$

It is well known that, if first-order logic is used as basis, there are satisfiable sentences F and scopes S , such that $\mathbf{circ}_S(F)$ is unsatisfiable, where *well-foundedness* is a sufficient condition for the satisfiability of $\mathbf{circ}_S(F)$, given that F is satisfiable [16]. Well-foundedness can be defined as follows [25]:

Definition 7 (Well-Founded Formula). A formula F is called *well-founded* with respect to scope S if and only if $F \models \text{project}_S(\text{circ}_S(F))$.

Well-foundedness holds for all universal first-order formulas with respect to all scopes and thus also for propositional formulas. Nevertheless, we make this property explicit here, since it allows generalize the definitions and properties discussed here straightforwardly to first-order logic. It is referenced below in Prop. 8 and Prop. 16.ix, which are applied in the proof of Thm. 23.i. The following proposition shows a precondition under which entailment from a circumscribed formula reduces to entailment from the original formula [25]:

Proposition 8 (Consequences of Scope-Determined Circumscriptions). For all formulas F, G and scopes S such that F is well-founded with respect to S and $G \in S$ it holds that $\text{circ}_S(F) \models G$ iff $F \models G$.

2.5 The diff Second-Order Operator and the Greatest Lower Bound

We define a third basic second-order operator, *diff* (“scoped difference”), which is defined like *project*, except that the condition $J \cap S \subseteq I$ is replaced by its negation:

Definition 9 (Second-Order Operator diff). For all interpretations I , scopes S and formulas F define:

$$I \models \text{diff}_S(F) \quad \text{iff}_{\text{def}} \quad \text{There exists an interpretation } J \text{ such that} \\ J \models F \text{ and } J \cap S \not\subseteq I.$$

The *diff* second-order operator can be applied, for example, to express the following: If $S = \overline{S}$, then the statement $I \models \neg \text{diff}_S(F)$ holds if and only if all models of F satisfy exactly those of the atoms occurring in S that are satisfied by I . The statement $I \models \neg \text{diff}_{\text{NEG}}(F)$ means that I is a lower bound w.r.t. \leq_{POS} of the set of models of F . Based on *diff*, the greatest lower bound of the models of a formula can be expressed with the following second-order operator:

Definition 10 (Second-Order Operator glb). For all formulas F define:

$$\text{glb}(F) \stackrel{\text{def}}{=} \text{circ}_{\text{NEG}}(\neg \text{diff}_{\text{NEG}}(F)).$$

For all formulas F , the formula $\text{glb}(F)$ has a single model that is this greatest lower bound of the set of models of F . A precise characterization of that model, based on entailment of atoms, is implied by the following proposition:

Proposition 11 (Characterization of the Greatest Lower Bound). For all sentences F and atoms P it holds that (i) $\text{glb}(F) \models P$ iff $F \models P$. (ii) $\text{glb}(F) \models \neg P$ iff $F \not\models P$.

2.6 Second-Order Operators in Terms of Predicate Quantification

Actually, all the three considered basic second-operators, *project*, *raise* and *diff* can be expressed in terms of predicate quantification only, analogously to the common way in which predicate circumscription is expressed by means of second-order quantification [16]. Recall that, as sketched in Sect. 2.2, if $S = \overline{S}$, then $\text{project}_S(F)$ is equivalent to $\exists P_1, \dots, \exists P_n F$, where P_1, \dots, P_n are those atoms in F that do not occur in S . We thus show the expressibility in terms of

project onto a scope S such that $S = \bar{S}$. We assume that ATOMS can be partitioned into two disjoint subsets of equal cardinality $\text{ATOMS}^0 = \{P_1, \dots, P_n\}$ and $\text{ATOMS}^1 = \{P_1^1, \dots, P_n^1\}$. Only atoms from ATOMS^0 are allowed in the user input formulas. Their correspondents ATOMS^1 are auxiliary atoms used to encode the second-order operators. For formulas F in which only atoms from ATOMS^0 do occur, let F^1 denote F after renaming each atom $P_i \in \text{ATOMS}^0$ to its correspondent $P_i^1 \in \text{ATOMS}^1$. The expression of **project**, **raise** and **diff** in terms of predicate quantification is then justified by the following equivalences, which hold for all finite scopes S and formulas F in which only atoms from ATOMS^0 do occur:

$$\text{project}_S(F) \equiv \text{project}_{S \cup \bar{S}}(F^1 \wedge \bigwedge_{L \in S}(L^1 \rightarrow L)). \quad (\text{ii})$$

$$\text{raise}_S(F) \equiv \text{project}_{S \cup \bar{S}}(F^1 \wedge \bigwedge_{L \in S}(L^1 \rightarrow L) \wedge \neg \bigwedge_{L \in S}(L^1 \leftarrow L)). \quad (\text{iii})$$

$$\text{diff}_S(F) \equiv \text{project}_{S \cup \bar{S}}(F^1 \wedge \neg \bigwedge_{L \in S}(L^1 \rightarrow L)). \quad (\text{iv})$$

Also the systematic renaming of atoms to obtain F^1 from F can be expressed in terms of predicate quantification: Let LITS^1 denote the set of all literals whose atom is in ATOMS^1 . Then $F^1 \equiv \text{project}_{\text{LITS}^1}(F \wedge \bigwedge_{P \in \text{ATOMS}^0}(P^1 \leftrightarrow P))$.

These encodings can be straightforwardly lifted to first-order logic if the scope S meets a certain restriction: On a first-order basis, scopes are sets of ground literals, which are possibly infinite [24, 25]. The requirement for lifting the encodings is that the scope contains for each predicate in some finite set of predicates either all ground literals, all positive ground literals or all negative ground literals with that predicate, and it does not contain any other literals. The scope then represents a finite set of predicates, possibly with associated signs. The first-order versions of (ii)–(iv) can be obtained from these equivalences by replacing $\bigwedge_{L \in S}(L^1 \rightarrow L)$ with $\forall x_1, \dots, \forall x_n \bigwedge_{L \in \text{predlits}(S)}(L^1 \rightarrow L)$, where n is the maximal arity of the predicates occurring in S and $\text{predlits}(S)$ denotes the set of first-order literals that contains $P(x_1, \dots, x_{\text{arity}(P)})$ ($\neg P(x_1, \dots, x_{\text{arity}(P)})$, resp.) for all predicates P such that all positive (negative, resp.) ground literals with P are in S and contains no other literals. Note that if all positive as well as all negative ground literals with predicate P are in S , then both $P(x_1, \dots, x_{\text{arity}(P)})$ and $\neg P(x_1, \dots, x_{\text{arity}(P)})$ are in $\text{predlits}(S)$. The required assumption on the signature is that the set of predicates can be partitioned analogously to the set of atoms in the propositional case: Literal L^1 denotes literal L with its predicate P replaced by P^1 , formula F^1 denotes F with all predicates P replaced by their correspondents P^1 . Equivalence (iii) also requires to replace $\bigwedge_{L \in S}(L^1 \leftarrow L)$ in an analogous way.

3 MIP Formulas and MIP Approximations

Recall that a *Horn formula* is a conjunction of Horn clauses, where a Horn clause is a clause with at most a single positive literal and an arbitrary number, including zero, of negative literals. We consider here just propositional Horn formulas. An important feature of them is that satisfiability can be decided in polynomial time. A satisfiable Horn formula has a single minimal model, whose representation as a set of atoms can be computed in polynomial time.

3.1 Model Intersection Steps, Horn and MIP Formulas

Horn formulas can be characterized not just syntactically, but also by a semantic property, based on the concept of “model intersection”, which can be rendered with the following second-order operator im (“intersect models”):

Definition 12 (Second-Order Operator im – Model Intersection Step).

For all interpretations I and formulas F define:

$$I \models \text{im}(F) \quad \text{iff}_{\text{def}} \quad \text{There exist interpretations } J, K \text{ such that} \\ J \models F, K \models F \text{ and } J \cap K \cap \text{POS} = I \cap \text{POS}.$$

The models of $\text{im}(F)$ are all the models that are obtained by “intersecting” two models of F , that is, all the models whose set of positive literals is the intersection of the sets of the positive literals of two models of F . The following property is immediate from the definition of im :

Proposition 13 (Entailment of Model Intersection Step). *For all formulas F it holds that $F \models \text{im}(F)$.*

The im operator can also be expressed in terms of predicate quantification, similarly as explained in Sect. 2.6. It is required that ATOMS can be partitioned not just into two, but into three corresponding partitions, ATOMS^0 , ATOMS^1 and ATOMS^2 . Let F be a formula in which only atoms from ATOMS^0 do occur, let F^1, F^2 denote F after replacing all atoms with their correspondents from ATOMS^1 and ATOMS^2 , respectively. Let LITS^0 denote the set of all literals whose atom is in ATOMS^0 . The following equivalence, which justifies the expression of im in terms of predicate quantification, then holds:

$$\text{im}(F) \equiv \text{project}_{\text{LITS}^0}(F^1 \wedge F^2 \wedge \bigwedge_{P \in \text{ATOMS}^0}(P \leftrightarrow P^1 \wedge P^2)). \quad (\text{v})$$

Based on im , we define the class of *MIP formulas* (“formulas with the model intersection property”):

Definition 14 (MIP Formula). A formula F is called a *MIP formula* if and only if

$$F \equiv \text{im}(F).$$

By Prop. 13, MIP formulas can equivalently be characterized as the formulas F for which it holds that $\text{im}(F) \models F$. MIP formulas provide a *semantic* characterization of Horn formulas: A formula is a MIP formula if and only if it is equivalent to a Horn formula [21, 5].

3.2 MIP Approximations

The *least Horn upper bound* [23] (also called *Horn approximation* [12]) of a given formula is the strongest Horn formula that is weaker than or equivalent to the given formula. It is equivalent to the conjunction of all prime implicates that are Horn [23]. A semantic characterization can be based on the model intersection property, but, as it seems, not straightforwardly in terms of the introduced second-order operators. We express it with an operator lmub (“least MIP upper bound”), defined as follows:

Definition 15 (Operator lmub – Least Upper MIP Bound). For all interpretations I and formulas F define $I \models \text{lmub}(F)$ if and only if I is in the smallest set \mathfrak{J} of interpretations such that $\mathfrak{J} \supseteq \{I \mid I \models F\}$ and for all $J, K \in \mathfrak{J}$ it holds that the interpretation I such that $I \cap \text{POS} = J \cap K \cap \text{POS}$ is also in \mathfrak{J} .

The set of models of $\text{lmub}(F)$ can also be characterized as the least fixed point of the function that maps the set of models of satisfiable formulas G to the set of models of $\text{im}(G)$ and maps $\{\}$ to the models of F . For all formulas F , the formula $\text{lmub}(F)$ is the strongest MIP formula that is weaker than or equivalent to formula F . The following proposition gathers properties of lmub :

Proposition 16 (Properties of lmub). *For all formulas F, G it holds that:*

- (i) $F \models \text{lmub}(F)$.
- (ii) $\text{lmub}(F)$ is a MIP formula.
- (iii) If $F \models G$ and G is a MIP formula, then $\text{lmub}(F) \models G$.
- (iv) F is a MIP formula if and only if $F \equiv \text{lmub}(F)$.
- (v) $\text{glb}(F) \models \text{lmub}(F)$.
- (vi) $\text{glb}(F) \equiv \text{glb}(\text{lmub}(F))$.
- (vii) $\text{glb}(F) \equiv \text{circ}_{\text{POS}}(\text{lmub}(F))$.
- (viii) $\text{project}_{\text{POS}}(\text{glb}(F)) \equiv \text{project}_{\text{POS}}(\text{lmub}(F))$.
- (ix) $\text{lmub}(F)$ is well-founded with respect to POS.

The least upper MIP bound of a given formula is entailed by the given formula (Prop. 16.i), is a MIP formula (Prop. 16.ii) and entails all MIP formulas that are entailed by the given formula (Prop. 16.iii). MIP formulas can not just be characterized as the formulas F such that $F \equiv \text{im}(F)$ (Def. 14), but also as the formulas F such that $F \equiv \text{lmub}(F)$ (Prop. 16.iv). This follows from Prop. 16.i, 16.ii and 16.iii. The greatest lower bound is contained in the set of models of the least upper MIP bound (Prop. 16.v), is also the greatest lower bound of that set (Prop. 16.vi) and is the minimal element of that set (Prop. 16.vii). All comparisons between interpretations are understood here with respect to \leq_{POS} . Under projection to POS, the greatest lower bound and the least upper MIP bound are equivalent, or, in other words: The set of all interpretations that are greater or equal to than the greatest lower bound is equal to the set of all interpretations that are greater than or equal to some model of the least upper MIP bound (Prop. 16.viii). Prop. 16.ix holds trivially since all propositional formulas are well-founded with respect to any scope. Nevertheless, it is explicitly stated here, since it can be derived in another way that also applies in a general setting with first-order logic as basis, where well-foundedness can not be taken as granted: $\text{lmub}(F) \models \text{project}_{\text{POS}}(\text{lmub}(F)) \equiv \text{project}_{\text{POS}}(\text{glb}(F)) \equiv \text{project}_{\text{POS}}(\text{circ}_{\text{POS}}(\text{lmub}(F)))$, which follows from Prop. 16.viii and 16.vii and since a projection of a formula is always entailed by the formula.

The operator fmub (“filled MIP upper bound”) defined in the following expresses another unique distinguished MIP formula that is weaker than or equivalent to its argument formula:

Definition 17 (Second-Order Operator fmub – “Filled” MIP Upper Bound). For all formulas F define:

$$\text{fmub}(F) \stackrel{\text{def}}{=} \text{project}_{\text{POS}}(\text{glb}(F)) \wedge \text{project}_{\text{NEG}}(F).$$

Its definiens in terms of second-order operators makes fmub easier to handle than lmub . The set of models of fmub , so-to-speak, completely “fills” the space of possibilities “between” the greatest lower bound and the maximal models.

More precisely, the models of $\text{fmub}(F)$ are *all* those interpretations that are greater than or equal to the greatest lower bound of the set of models of F and at the same time less than or equal to some model of F . The following properties follow from the definitions of the involved operators:

Proposition 18 (Properties of fmub). *For all formulas F it holds that:*

- (i) $F \models \text{fmub}(F)$.
- (ii) $\text{fmub}(F)$ is a MIP formula.
- (iii) $\text{glb}(F) \models \text{fmub}(F)$.
- (iv) $\text{glb}(F) \equiv \text{glb}(\text{fmub}(F))$.
- (v) $\text{glb}(F) \equiv \text{circ}_{\text{POS}}(\text{fmub}(F))$.
- (vi) $\text{project}_{\text{POS}}(\text{glb}(F)) \equiv \text{project}_{\text{POS}}(\text{fmub}(F))$.
- (vii) $\text{lmub}(F) \models \text{fmub}(F)$.
- (viii) $\text{fmub}(\text{project}_{\text{POS}}(F)) \equiv \text{lmub}(\text{project}_{\text{POS}}(F))$.
- (ix) $\text{project}_{\text{NEG}}(\text{fmub}(F)) \equiv \text{project}_{\text{NEG}}(\text{lmub}(F))$.

Propositions 18.i–18.vi are analogous to Prop. 16.i, 16.ii and 16.v–16.viii. The left sides of Prop. 18.iii–18.vi are identical to the left sides of Prop. 16.v–16.viii, respectively. For all formulas F , the least upper MIP bound of F entails $\text{fmub}(F)$ (Prop. 18.vii). If F is positive, then both are equivalent (Prop. 18.viii). For all formulas F it holds that under projection to NEG the formulas $\text{fmub}(F)$ and $\text{lmub}(F)$ are equivalent, or, in other words: The set of all interpretations that are less than or equal to some model of $\text{fmub}(F)$ is identical to the set of all interpretations that are less than or equal to some model of the least upper MIP bound of F (Prop. 18.ix). From Prop. 18.vi and 16.viii it follows that also under projection to POS the formula $\text{fmub}(F)$ and the least upper MIP bound of F are equivalent, that is, $\text{project}_{\text{POS}}(\text{fmub}(F)) \equiv \text{project}_{\text{POS}}(\text{lmub}(F))$.

4 Expressing Definientia in Given Classes

We consider here the class of MIP formulas as well as two further classes that are also closed under equivalence. For such a class \mathcal{C} , we call a definiens in \mathcal{C} a \mathcal{C} -definiens and term the property that a \mathcal{C} -definiens exists \mathcal{C} -definability. A formula that is a \mathcal{C} -definiens under the sole precondition of \mathcal{C} -definability is called a *representative \mathcal{C} -definiens*. The representative definientia presented in the following theorems are second-order formulas, which, when their arguments are instantiated, are equivalent to a propositional formula in the respective syntactic class that underlies the considered semantically characterized class.

4.1 Expressing MIP-Definientia

The following theorem gives a characterization of MIP-definability and a representative MIP-definiens:

Theorem 19 (MIP-Definability and Representative MIP-Definiens).

For all scopes S and formulas F, G it holds that:

- (i) G is MIP-definable in terms of S within F if and only if

$$\text{lmub}(\text{gsnc}_S(F, G)) \models \text{gwsc}_S(F, G).$$

- (ii) *If G is MIP-definable in terms of S within F , then the following formula is a MIP-definiens of G in terms of S within F :*

$$\text{lmub}(\text{gsnc}_S(F, G)).$$

Proof. (19.i) Left-to-right: Assume the left side of the theorem. Then, there exists a MIP formula H such that $\text{gsnc}_S(F, G) \models H \models \text{gWSC}_S(F, G)$. From Prop. 16.iii it follows that $\text{lmub}(\text{gsnc}_S(F, G)) \models H \models \text{gWSC}_S(F, G)$, which implies the right side. Right-to-left: Assume $\text{lmub}(\text{gsnc}_S(F, G)) \models \text{gWSC}_S(F, G)$. Then, by Prop. 16.i and Def. 5.i it follows that $\text{lmub}(\text{gsnc}_S(F, G))$ is a definiens of G in terms of S within F . By Prop. 16.ii it is also a MIP formula, hence a MIP-definiens. (19.ii) Follows from Thm. 19.i, Prop. 16.i, Prop. 16.ii and Def. 5.i. \square

The MIP-definiens according to Thm. 19.ii is the strongest MIP-definiens, that is, it entails all MIP-definientia, which follows from Def. 5.i and Prop. 16.iii:

Proposition 20 (The Representative MIP-Definiens is the Strongest). *Let S be a scope and let F, G be formulas such that G is MIP-definable in terms of S within F . Then for all MIP-definientia H of G in terms of S within F it holds that $\text{lmub}(\text{gsnc}_S(F, G)) \models H$.*

By Prop. 18.i, for all formulas F it holds that $F \models \text{fmub}(F)$. Thus, if

$$\text{fmub}(\text{gsnc}_S(F, G)) \models \text{gWSC}_S(F, G), \quad (\text{vi})$$

then it follows from Def. 5.i that $\text{fmub}(\text{gsnc}_S(F, G))$ is a MIP-definiens of G in terms of S within F . In contrast to the MIP-definiens according to Thm. 19.ii, that is, $\text{lmub}(\text{gsnc}_S(F, G))$, the formula $\text{fmub}(\text{gsnc}_S(F, G))$ only involves second-order operators that can be expressed by predicate quantification. However, MIP-definability does not imply that $\text{fmub}(\text{gsnc}_S(F, G))$ is a definiens – it might be too weak, as shown in the following simple example:

Example 21 (MIP-Definiens: lmub vs. fmub). Let $F = \top$, $G = (p \rightarrow q)$ and $S = \{p, q\}$. Then $\text{gsnc}_S(F, G) \equiv \text{lmub}(\text{gsnc}_S(F, G)) \equiv \text{gWSC}_S(F, G) \equiv G$. Thus there is a single definiens of G in terms of S within F , which is G itself. Moreover, this definiens is a MIP-definiens. However $\text{fmub}(\text{gsnc}_S(F, G)) \equiv \top \not\models \text{gWSC}_S(F, G) \equiv G$. Thus $\text{fmub}(\text{gsnc}_S(F, G))$ is too weak to be a definiens of G .

4.2 Expressing COA-Definientia

We now turn to a formula class for which definability and a representative definiens can be characterized purely with second-order operators that can be expressed in terms of predicate quantification:

Definition 22 (COA Formula). A formula F is called a *COA formula* if and only if F is MIP formula and it holds that $F \in \text{POS}$.

A formula is a COA (“conjunction of atoms”) formula if and only if it is equivalent to a positive Horn formula, or, in other words, equivalent to a conjunction of atoms. The following theorem gives a characterization of COA-definability and a representative COA-definiens, in terms of glb and fmub :

Theorem 23 (COA-Definability and Representative COA-Definiens). *For all scopes S and formulas F, G it holds that:*

- (i) G is COA-definable in terms of S within F if and only if

$$\text{glb}(\text{gsnc}_{S \cap \text{POS}}(F, G)) \models \text{gWSC}_{S \cap \text{POS}}(F, G).$$

(ii) *If G is COA-definable in terms of S within F , then the following formula is a COA-definiens of G in terms of S within F :*

$$\text{fmub}(\text{gsnc}_{S \cap \text{POS}}(F, G)).$$

Proof. (23.i) Consider the following equivalences: G is COA-definable in terms of S within F iff (by Def. 22) G is MIP-definable in terms of $S \cap \text{POS}$ within F iff (by Thm. 19.i) $\text{lmub}(\text{gsnc}_{S \cap \text{POS}}(F, G)) \models \text{gWSC}_{S \cap \text{POS}}(F, G)$ iff (by Prop. 4.ii, 16.ix, 8) $\text{circ}_{\text{POS}}(\text{lmub}(\text{gsnc}_{S \cap \text{POS}}(F, G))) \models \text{gWSC}_{S \cap \text{POS}}(F, G)$ iff (by Prop. 16.vii) $\text{glb}(\text{gsnc}_{S \cap \text{POS}}(F, G)) \models \text{gWSC}_{S \cap \text{POS}}(F, G)$. (23.ii) From Def. 22 and Thm. 19.ii it follows that $\text{lmub}(\text{gsnc}_{S \cap \text{POS}}(F, G))$ is a COA-definiens of G in terms of S within F . By Prop. 4.i and 18.viii the following equivalence holds: $\text{lmub}(\text{gsnc}_{S \cap \text{POS}}(F, G)) \equiv \text{fmub}(\text{gsnc}_{S \cap \text{POS}}(F, G))$. \square

The following example shows a case where the strongest and weakest definientia, characterized by the GSNC and the GWSC, are no COA formulas, but there exists a COA-definiens between these extremes:

Example 24 (COA-Definiens). Let F be the formula

$$(q \rightarrow r \vee s) \wedge (t \rightarrow q) \wedge ((r \vee s) \wedge u \rightarrow p) \wedge (p \rightarrow t \wedge u).$$

Consider the task of finding definientia of p within F , in terms of positive occurrences of the other symbols. Let $S = \{+q, +r, +s, +t, +u\}$. It holds that

$$\begin{aligned} \text{gsnc}_S(F, p) &\equiv q \wedge t \wedge u \wedge (r \vee s), \text{ and} \\ \text{gWSC}_S(F, p) &\equiv u \wedge (q \vee r \vee s \vee t). \end{aligned}$$

It is easy to see that $\text{gsnc}_S(F, p) \models \text{gWSC}_S(F, p)$. Thus, we know that p is definable in terms of S within F , and, moreover, that $\text{gsnc}_S(F, p)$ provides the strongest such definiens and $\text{gWSC}_S(F, p)$ the weakest one. Neither one of these two definientia is a COA formula. However, by applying Thm. 23.i, there must exist a COA-definiens, since, assuming that $\text{ATOMS} = \{p, q, r, s, t, u\}$, it holds that $\text{glb}(\text{gsnc}_S(F, p)) \equiv (q \wedge t \wedge u \wedge \neg p \wedge \neg r \wedge \neg s) \models \text{gWSC}_S(F, p)$. We can thus apply Thm. 23.ii to justify that the formula $\text{fmub}(\text{gsnc}_S(F, p))$ is a COA-definiens of p in terms of S within F . This formula is equivalent to $(q \wedge t \wedge u)$. It is indeed easy to verify that $\text{gsnc}_S(F, p) \models \text{fmub}(\text{gsnc}_S(F, p)) \models \text{gWSC}_S(F, p)$.

4.3 Expressing KRO-Definientia with a Meta-Level Representation

A seemingly straightforward idea to express syntactic constraints by means of vocabulary restrictions would be enriching the vocabulary by “meta-level” symbols for logic operators, and applying restrictions on these. However, this is typically not sufficient, since arbitrary combinations of disjunctions and negations of formulas would also meet such restrictions. For formula classes that are not closed under disjunction and negation this must be prevented. Negation can be excluded with literal projection. By excluding negation as well as disjunction, COA-definientia provide a means to encode other formula classes on the meta-level as vocabulary restrictions. This is now shown for a particular class of formulas: Recall that a Krom formula is a formula in clausal form, where each clause contains at most two literals. Like Horn formulas, Krom formulas can be decided in polynomial time. We define KRO formulas as semantic version of Krom formulas:

Definition 25 (KRO Formulas). A *KRO formula* is a formula that is equivalent to a Krom formula.

The following theorem gives a characterization of KRO-definability and a representative KRO-definiens:

Theorem 26 (KRO-Definability and Representative KRO-Definiens). Assume a fixed total order \leq on literals. For all scopes S let:

$$\begin{aligned} \text{KD}(S) &\stackrel{\text{def}}{=} (\text{empty} \leftrightarrow \perp) \wedge \bigwedge_{L, M \in S, L \leq M, L \neq \overline{M}} (\text{clause}(L, M) \leftrightarrow L \vee M). \\ \text{KS}(S) &\stackrel{\text{def}}{=} \{\text{empty}\} \cup \{\text{clause}(L, M) \mid L, M \in S, L \leq M, L \neq \overline{M}\}. \end{aligned}$$

For all scopes S and formulas F, G it holds that:

- (i) G is KRO-definable in terms of S within F if and only if G is COA-definable in terms of $\text{KS}(S)$ within $(F \wedge \text{KD}(S))$.
- (ii) If G is KRO-definable in terms of S within F , then the following formula is a KRO-definiens of G in terms of S within F :

$$\text{project}_S(\text{fmub}(\text{gsnc}_{\text{KS}(S)}(F \wedge \text{KD}(S), G)) \wedge \text{KD}(S)).$$

In the theorem statement, $\text{KD}(S)$ denotes the conjunction of the definitions of the auxiliary atoms **empty**, representing the empty clause, and of the form $\text{clause}(L, M)$, representing nonempty Krom clauses, where L, M are those literals from the original vocabulary that are in the scope S . Evidently, the size of $\text{KD}(S)$ is polynomially bounded by the cardinality of the original vocabulary. The set of the positive literals with the auxiliary atoms is denoted by $\text{KS}(S)$. KRO-definability is then expressed in Thm. 26.i as COA-definability with respect to $\text{KS}(S)$ within the original background formula F , conjoined with the definitions $\text{KD}(S)$. Justified by Thm. 23.ii, the inner formula of Thm. 26.ii, that is, $\text{fmub}(\text{gsnc}_{\text{KS}(S)}(F \wedge \text{KD}(S), G))$, then denotes the representative COA-definition of G in terms of $\text{KS}(S)$. The actual KRO-definiens is then obtained from this COA-definiens by conjoining it with the definition $\text{KD}(S)$ of the auxiliary atoms and then applying projection onto the original vocabulary. We note that, as for COA, also for KRO, definability and a representative definiens can be characterized purely with operators that can be expressed in terms of predicate quantification.

The following example shows a case where, analogously to Examp. 24, the strongest and weakest definiens, characterized by the GSNC and the GWSC, are no KRO formulas, but there exists a KRO-definiens between these extremes:

Example 27 (KRO-Definiens). Let F be the formula

$$((q \leftrightarrow r) \rightarrow s \vee t \vee u) \wedge (s \vee t \vee u \rightarrow p) \wedge (p \rightarrow (q \leftrightarrow r)).$$

Consider the task of finding definiens of p within F in terms of the other symbols. Let $S = \{+q, -q, +r, -r, +s, -s, +t, -t, +u, -u\}$. It holds that

$$\begin{aligned} \text{gsnc}_S(F, p) &\equiv (r \leftrightarrow q) \wedge (s \vee t \vee u), & \text{and} \\ \text{gWSC}_S(F, p) &\equiv (r \rightarrow q \vee s \vee t \vee u) \wedge (q \rightarrow r \vee s \vee t \vee u) \end{aligned}$$

Since $\text{gsnc}_S(F, p) \models \text{gWSC}_S(F, p)$, we know that p is definable in terms of S within F , and, moreover, that $\text{gsnc}_S(F, p)$ provides the strongest such definiens and $\text{gWSC}_S(F, p)$ the weakest one. Clearly, neither one of these two definiens is equivalent to a Krom formula. If the scope is restricted to $\{q, r\}$, there exists a

unique definiens, which is equivalent to a Krom formula: $\text{gsnc}_{\{q,r\}}(F, p) \equiv (q \leftrightarrow r) \equiv \text{gWSC}_{\{q,r\}}(F, p)$. However, for scope S as specified above, this is neither the weakest nor the strongest definiens. Also from Theorem. 26.i it follows that there must exist a Krom definiens, since, assuming that ATOMS is the union of $\{p, q, r, s, t, u\}$ and the set of atoms occurring in $\text{KS}(S)$ it can be verified that $\text{glb}(\text{gsnc}_{\text{KS}(S)}((F \wedge \text{KD}(S)), p)) \models \text{gWSC}_{\text{KS}(S)}((F \wedge \text{KD}(S)), p)$. We can thus apply Thm. 26.ii to justify that the following formula is a Krom definiens of p in terms of S within F : $\text{project}_S(\text{fmub}(\text{gsnc}_{\text{KS}(S)}((F \wedge \text{KD}(S)), p)) \wedge \text{KD}(S))$. This formula is equivalent to $\text{project}_S(\text{clause}(q, \neg r) \wedge \text{clause}(\neg q, r) \wedge \text{KD}(S))$, and thus equivalent to $(q \leftrightarrow r)$.

5 Conclusion

We have begun to investigate a formalized and mechanizable way of combining two different aspects: Expressibility in formula classes is viewed from the point of expressibility in restricted vocabularies, which can be formulated by predicate quantification. In particular, we considered whether definability and representative definitia with respect to given formula classes that are not just specified as vocabulary restrictions can be characterized in terms of second-order operators which ultimately can be expressed just by predicate quantification. We have seen that with propositional logic as basis, such characterizations are possible for the class of formulas that are equivalent to a conjunction of atoms and the class of formulas that are equivalent to a Krom formula. For the class of formulas that are equivalent to a Horn formula, a further operator is required, which seems not straightforwardly reducible to predicate quantification.

An inherent feature or limitation of the presented approach is that it applies only to formula classes that are *closed under equivalence*. Nevertheless, with respect to vocabulary restrictions, elimination methods usually produce outputs that do no longer contain the quantified predicates, thereby ensuring that results are also in the corresponding syntactic classes. It needs to be investigated, in which way elimination methods applied to the suggested second-order expressions for the considered formula classes yield results that are actually also in the corresponding syntactic classes.

The underlying framework of second-order operators straightforwardly extends from propositional to first-order logic (see, e.g., [24, 25, 28]). However, for the first-order case the correspondence of the semantic characterizations of formula classes to expressibility in syntactic classes such as Horn formulas, conjunctions of atoms and Krom formulas still needs to be examined.

In the paper, operators and properties have been formally defined in terms of each other in a way that fits mechanization. In fact, they have been defined similarly on top of the *ToyElim* system [27].¹ This is currently only suitable for small experiments and an advanced implementation of the suggested operators seems to be a major challenge on its own. At least in principle, the presented characterizations of definitia should be expressible also on top of other systems for second-order quantifier elimination and its variants, the computation of forgetting, projection and uniform interpolants.

¹ See <http://cs.christophwernhard.com/toyelim/>.

References

1. Biere, A.: Resolve and expand. In: Theory and Applications of Satisfiability Testing: 7th Int. Conf., SAT 2004. LNCS, vol. 3542, pp. 238–246. Springer (2004)
2. Calvanese, D., Giacomo, G.D., Lenzerini, M., Vardi, M.Y.: View-based query processing: On the relationship between rewriting, answering and losslessness. *Theor. Comput. Sci.* 371(3), 169–182 (2007)
3. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. *J. Artif. Intell. Res.* 31, 273–318 (2008)
4. Cuenca Grau, B., Motik, B.: Reasoning over ontologies with hidden content: The import-by-query approach. *J. Artif. Intell. Res.* 45, 197–255 (2012)
5. Dechter, R., Pearl, J.: Structure identification in relational data. *Artif. Intell.* 58, 237–270 (1992)
6. Doherty, P., Lukaszewicz, W., Szalas, A.: Computing strongest necessary and weakest sufficient conditions of first-order formulas. In: Proc. 17th Int. Joint Conf. on Artif. Intell., IJCAI-01. pp. 145–151. Morgan Kaufmann (2001)
7. Eén, N., Biere, A.: Effective preprocessing in SAT through variable and clause elimination. In: Theory and Applications of Satisfiability Testing, 8th Int. Conf., SAT 2005. LNCS, vol. 3569, pp. 61–75 (2005)
8. Franconi, E., Kerhet, V., Ngo, N.: Exact query reformulation over databases with first-order and description logics ontologies. *J. Artif. Intell. Res.* 48, 885–922 (2013)
9. Gabbay, D.M., Schmidt, R.A., Szalas, A.: Second-Order Quantifier Elimination: Foundations, Computational Aspects and Applications. College Publications, London (2008)
10. Ghilardi, S., Lutz, C., Wolter, F.: Did I damage my ontology? A case for conservative extensions in description logics. In: Proc. 10th Int. Conf. on Princ. of Knowledge Rep. and Reasoning, KR 2006. pp. 187–197. AAAI Press (2006)
11. Heule, M., Järvisalo, M., Biere, A.: Clause elimination procedures for CNF formulas. In: Logic for Prog., Artif. Intell. and Reasoning: 17th Int. Conf., LPAR-17. LNCS, vol. 6397, pp. 357–371. Springer (2010)
12. Kautz, H., Kearns, M., Selman, B.: Horn approximations of empirical data. *Artif. Intell.* 74, 129–145 (1995)
13. Konev, B., Walther, D., Wolter, F.: Forgetting and uniform interpolation in large-scale description logic terminologies. In: Proc. 21st Int. Joint Conf. on Artif. Intell., IJCAI-09. pp. 830–835. AAAI Press (2009)
14. Koopmann, P., Schmidt, R.A.: Uniform interpolation of \mathcal{ALC} -ontologies using fix-points. In: 9th Int. Symp. on Frontiers of Combining Systems, FroCoS 2013. LNCS (LNAI), vol. 8152, pp. 87–102. Springer (2013)
15. Lang, J., Liberatore, P., Marquis, P.: Propositional independence – formula-variable independence and forgetting. *J. Artif. Intell. Res.* 18, 391–443 (2003)
16. Lifschitz, V.: Circumscription. In: Gabbay, D.M., Hogger, C.J., Robinson, J.A. (eds.) *Handbook of Logic in Artificial Intelligence and Logic Programming*, vol. 3, pp. 298–352. Oxford University Press (1994)
17. Lin, F.: On strongest necessary and weakest sufficient conditions. *Artif. Intell.* 128, 143–159 (2001)
18. Lutz, C., Wolter, F.: Foundations for uniform interpolation and forgetting in expressive description logics. In: Proc. 22nd Int. Joint Conf. on Artif. Intell., IJCAI-11. pp. 989–995. AAAI Press (2011)
19. Manthey, N., Philipp, T., Wernhard, C.: Soundness of inprocessing in clause sharing SAT solvers. In: Theory and Applications of Satisfiability Testing, 16th Int. Conf., SAT 2013. LNCS, vol. 7962, pp. 22–39. Springer (2013)

20. Marx, M.: Queries determined by views: Pack your views. In: Proc. 25th ACM SIGMOD-SIGACT-SIGART Symp. on Princ. of Database Systems, PODS '07. pp. 23–30. ACM (2007)
21. McKinsey, J.C.C.: The decision problem for some classes of sentences without quantifiers. *J. Symb. Log.* 8, 61–76 (1943)
22. Nash, A., Segoufin, L., Vianu, V.: Views and queries: Determinacy and rewriting. *ACM Trans. Database Syst.* 35(3) (2010)
23. Selman, B., Kautz, H.A.: Knowledge compilation using Horn approximations. In: Proc. 9th Nat. Conf. on Artif. Intell., AAAI-91, Volume 2. pp. 904–909. AAAI Press (1991)
24. Wernhard, C.: Literal projection for first-order logic. In: Logics in Artif. Intell.: 11th European Conf., JELIA 08. LNCS (LNAI), vol. 5293, pp. 389–402. Springer (2008)
25. Wernhard, C.: Projection and scope-determined circumscription. *J. Symb. Comput.* 47(9), 1089–1108 (2012)
26. Wernhard, C.: Abduction in logic programming as second-order quantifier elimination. In: 9th Int. Symp. on Frontiers of Combining Systems, FroCoS 2013. LNCS (LNAI), vol. 8152, pp. 103–119. Springer (2013)
27. Wernhard, C.: Computing with logic as operator elimination: The ToyElim system. In: Applications of Declarative Programming and Knowledge Management, 19th Int. Conf. (INAP 2011) and 25th Workshop on Logic Prog. (WLP 2011), Revised Selected Papers. LNCS (LNAI), vol. 7773. Springer (2013)
28. Wernhard, C.: Expressing view-based query processing and related approaches with second-order operators. Tech. Rep. Knowledge Representation and Reasoning 14–02, Technische Universität Dresden (2014), <http://www.wv.inf.tu-dresden.de/Publications/2014/report-2014-02.pdf>