

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.

Tasks in knowledge processing such as view-based query rewriting with exact rewritings [1, 14, 16, 5, 21] 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 [7, 6, 9, 12, 18, 2, 19, 10], and in the pre-processing of propositional formulas [4, 8, 13]. 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 definientia 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 [3, 15]:

A propositional formula is equivalent to a Horn formula if and only if it has the *model intersection property*. Literal projection [11, 17] 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.

As elaborated in [22], such characterizations of definientia and definability, that is, the existence of definientia, can be developed in a framework on the basis of three second-order operators: For literal projection, for “raising” [18], which can be applied to express generalizations of circumscription, including model maximization, and for “scoped difference” [22], which allows, for example, to specify a formula whose models are exactly all the lower bounds of the set of models of a given formula (interpretations are compared there w.r.t. the subset relationship between the sets of the ground atoms that they satisfy). The characterizations of definientia and definability are then expressed by further second-order operators, which can be defined like macros in terms of these three basic ones. As also shown in [22], the three basic operators themselves can all be encoded as predicate quantification, such that characterizations of definability and definientia in terms of these operators can be translated to characterizations with predicate quantification as the only second-order operator.

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.

Although the envisaged underlying logic is classical first-order logic, the material in has been technically developed in [22] essentially for propositional logic, for simplicity of presentation and because the considered formula classes there have immediate correspondence to syntactic classes such as Horn and Krom formulas as well as conjunctions of atoms. Actually, the material transfers to a large extent easily to first-order logic, following the principles in [17, 18, 21]. Some minor particularities are indicated in [22]. What still need to be examined for the first-order case is the correspondence of the semantic characterizations of formula classes to expressibility in syntactic classes.

Operators and properties are 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 [20].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 definientia 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.

References

- [1] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y. Vardi. View-based query processing: On the relationship between rewriting, answering and losslessness. *TCS*, 371(3):169–182, 2007.
- [2] B. Cuenca Grau and B. Motik. Reasoning over ontologies with hidden content: The import-by-query approach. *JAIR*, 45, 2012.
- [3] Rina Dechter and Judea Pearl. Structure identification in relational data. *AI*, 58:237–270, 1992.
- [4] Niklas Eén and Armin Biere. Effective preprocessing in SAT through variable and clause elimination. In *SAT 2005*, volume 3569 of *LNCS (LNAI)*, pages 61–75, 2005.
- [5] Enrico Franconi, Volha Kerhet, and Nhung Ngo. Exact query reformulation over databases with first-order and description logics ontologies. *JAIR*, 48:885–922, 2013.
- [6] Dov M. Gabbay, Renate A., Schmidt, and Andrzej Szalas. *Second-Order Quantifier Elimination: Foundations, Computational Aspects and Applications*. College Publications, London, 2008.
- [7] Silvio Ghilardi, Carsten Lutz, and Frank Wolter. Did I damage my ontology? A case for conservative extensions in description logics. In *KR 2006*, pages 187–197. AAAI Press, 2006.
- [8] Marijn Heule, Matti Järvisalo, and Armin Biere. Clause elimination procedures for CNF formulas. In *LPAR-17*, volume 6397 of *LNCS*, pages 357–371. Springer, 2010.
- [9] Boris Konev, Dirk Walther, and Frank Wolter. Forgetting and uniform interpolation in large-scale description logic terminologies. In *IJCAI-09*, pages 830–835. AAAI Press, 2009.
- [10] Patrick Koopmann and Renate A. Schmidt. Uniform interpolation of *ALC*-ontologies using fixpoints. In *FroCoS 2013*, volume 8152 of *LNCS (LNAI)*, pages 87–102. Springer, 2013.
- [11] Jérôme Lang, Paolo Liberatore, and Pierre Marquis. Propositional independence – formula-variable independence and forgetting. *JAIR*, 18:391–443, 2003.
- [12] Carsten Lutz and Frank Wolter. Foundations for uniform interpolation and forgetting in expressive description logics. In *IJCAI-11*, pages 989–995. AAAI Press, 2011.
- [13] Norbert Manthey, Tobias Philipp, and Christoph Wernhard. Soundness of inprocessing in clause sharing SAT solvers. In *SAT 2013*, volume 7962 of *LNCS*, pages 22–39. Springer, 2013.
- [14] Maarten Marx. Queries determined by views: pack your views. In *PODS '07*, pages 23–30. ACM, 2007.
- [15] J. C. C. McKinsey. The decision problem for some classes of sentences without quantifiers. *JSL*, 8:61–76, 1943.
- [16] Alan Nash, Luc Segoufin, and Victor Vianu. Views and queries: Determinacy and rewriting. *TODS*, 35(3), 2010.
- [17] Christoph Wernhard. Literal projection for first-order logic. In *JELIA 08*, volume 5293 of *LNCS (LNAI)*, pages 389–402. Springer, 2008.
- [18] Christoph Wernhard. Projection and scope-determined circumscription. *JSC*, 47(9):1089–1108, 2012.
- [19] Christoph Wernhard. Abduction in logic programming as second-order quantifier elimination. In *FroCoS 2013*, volume 8152 of *LNCS (LNAI)*, pages 103–119. Springer, 2013.
- [20] Christoph Wernhard. Computing with logic as operator elimination: The ToyElim system. In *INAP 2011/WLP 2011*, volume 7773 of *LNCS (LNAI)*. Springer, 2013.
- [21] Christoph Wernhard. Expressing view-based query processing and related approaches with second-order operators. Technical Report KRR 14–02, TU Dresden, 2014. <http://www.wv.inf.tu-dresden.de/Publications/2014/report-2014-02.pdf>.
- [22] Christoph Wernhard. Second-order characterizations of definientia in formula classes. Technical Report KRR 14–03, TU Dresden, 2014. <http://www.wv.inf.tu-dresden.de/Publications/2014/report-2014-03.pdf>.

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