Application Patterns of Projection/Forgetting

Christoph Wernhard

Technische Universität Dresden

Projection and forgetting are tools that allow to express a variety of tasks in knowledge processing. Computing the *forgetting* about a set of predicates in a formula can be viewed as existentially quantifying the predicates and eliminating these existential second-order quantifiers, which results in a formula that does no longer contain the "forgotten" or quantified predicates, but is, with respect to the retained predicates, equivalent to the original formula. Dually, computing the *projection* onto a set of predicates, often also called *uniform interpolant* with respect to the set of predicates, is computing the forgetting about all predicates except of those in the given set. It is very useful to consider forgetting and projection as *second-order operators* with elimination as the associated computational processing. In this way, forgetting and projection can be nested and further second-order operators can be defined in terms of these primitive ones. Operator properties can be utilized to clarify semantic relationships and to justify computational techniques that apply to particular nested combinations. In the presentation, it is outlined how such a framework of classical logic extended by second-order operators, with elimination as associated computation, can be used to express the following properties and concepts, along with their associated computational tasks:

- Literal forgetting and literal projection, where the polarity of predicate occurrences is considered [13, 19]
- Theory approximations that are based on vocabulary restrictions: weakest sufficient and strongest necessary condition [14, 5, 21, 24]
- Definability and definientia of a given formula in terms of a given vocabulary within a given background formula [18, 16, 24]
- Perfect query rewriting [3, 24]
- Exact query rewriting [3, 16, 17, 7, 24]
- Greatest lower bound of a formula
- Alternate definiens in given formula classes (modulo equivalence): conjunctions of facts, Krom formulas [25]
- Conservative theory extension [9, 24]
- Least common subsumer with respect to a given vocabulary [1]
- Concept matching modulo equivalence [2]
- Predicate circumscription [21]
- Stable and 3-valued partial stable model semantics, generalized such that chosen predicates can be handled with open world semantics [20, 22]
- Abductive explanations w.r.t. classical semantics as well as stable and partial stable model semantics [22]

With the suggested second-order formulations, such knowledge processing techniques can not only be expressed and related to each other in a single semantic framework, but also – at least in principle – can be processed and combined within a single general first-order-based system that is able to perform second-order operator elimination. For small experiments, an implemented such system is available [23]. An advanced system seems to be a major challenge today. Relevant techniques include methods for second-order quantifier elimination based on first-order logic [4, 8], recent advances with respect to forgetting/uniform interpolation in the context of description logics, e.g., [9, 11, 12], and progress in SAT solving, where Boolean variable elimination now plays an important role in preprocessing [6, 10, 15].

References

[1] F. Baader and R. Küsters. Computing the least common subsumer and the most specific concept in the presence of cyclic $\mathcal{ALN}$-concept descriptions. In *KI-98*, volume 1504 of *LNCS*, pages 129–140. Springer, 1998.

[2] F. Baader, R. Küsters, A. Borgida, and D. McGuinness. Matching in description logics. *JLC*, 9(3):411–447, 1999.

[3] D. Calvanese, G. D. Giacomo, M. Lenzerini, and M. Y. Vardi. View-based query processing: On the relationship between rewriting, answering and losslessness. *TCS*, 371(3):169–182, 2007.

[4] P. Doherty, W. Łukaszewicz, and A. Szałas. Computing circumscription revisited: A reduction algorithm. *JAR*, 18(3):297–338, 1997.

[5] P. Doherty, W. Łukaszewicz, and A. Szałas. Computing strongest necessary and weakest sufficient conditions of first-order formulas. In *IJCAI-01*, pages 145–151. Morgan Kaufmann, 2001.

[6] N. Eén and A. Biere. Effective preprocessing in SAT through variable and clause elimination. In *SAT 2005*, volume 3569 of *LNCS*, pages 61–75, 2005.

[7] E. Franconi, V. Kerhet, and N. Ngo. Exact query reformulation over databases with first-order and description logics ontologies. *JAIR*, 48:885–922, 2013.

[8] D. Gabbay and H. J. Ohlbach. Quantifier elimination in second-order predicate logic. In *KR'92*, pages 425–435. Morgan Kaufmann, 1992.

[9] S. Ghilardi, C. Lutz, and F. Wolter. Did I damage my ontology? A case for conservative extensions in description logics. In *KR 2006*, pages 187–197. AAAI Press, 2006.

[10] M. Heule, M. Järvisalo, and A. Biere. Clause elimination procedures for CNF formulas. In *LPAR-17*, volume 6397 of *LNCS*, pages 357–371. Springer, 2010.

[11] B. Konev, D. Walther, and F. Wolter. Forgetting and uniform interpolation in large-scale description logic terminologies. In *IJCAI-09*, pages 830–835. AAAI Press, 2009.

[12] P. Koopmann and R. A. Schmidt. Uniform interpolation of $\mathcal{ALC}$-ontologies using fixpoints. In *FroCoS 2013*, volume 8152 of *LNCS (LNAI)*, pages 87–102. Springer, 2013.

[13] J. Lang, P. Liberatore, and P. Marquis. Propositional independence – formula-variable independence and forgetting. *JAIR*, 18:391–443, 2003.

[14] F. Lin. On strongest necessary and weakest sufficient conditions. *AI*, 128(1–2):143–159, 2001.

[15] N. Manthey, T. Philipp, and C. Wernhard. Soundness of inprocessing in clause sharing SAT solvers. In *SAT 2013*, volume 7962 of *LNCS*, pages 22–39. Springer, 2013.

[16] M. Marx. Queries determined by views: pack your views. In *PODS '07*, pages 23–30. ACM, 2007.

[17] A. Nash, L. Segoufin, and V. Vianu. Views and queries: Determinacy and rewriting. *TODS*, 35(3), 2010.

[18] A. Tarski. Einige methologische Untersuchungen zur Definierbarkeit der Begriffe. *Erkenntnis*, 5:80–100, 1935.

[19] C. Wernhard. Literal projection for first-order logic. In *JELIA 08*, volume 5293 of *LNCS (LNAI)*, pages 389–402. Springer, 2008.

[20] C. Wernhard. Circumscription and projection as primitives of logic programming. In *Tech. Comm. ICLP'10*, volume 7 of *LIPIcs*, pages 202–211, 2010.

[21] C. Wernhard. Projection and scope-determined circumscription. *JSC*, 47(9):1089–1108, 2012.

[22] C. Wernhard. Abduction in logic programming as second-order quantifier elimination. In *FroCoS 2013*, volume 8152 of *LNCS (LNAI)*, pages 103–119. Springer, 2013.

[23] C. Wernhard. Computing with logic as operator elimination: The ToyElim system. In *INAP 2011/WLP 2011*, volume 7773 of *LNCS (LNAI)*. Springer, 2013.

[24] C. 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`.

[25] C. 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`.