Reasoning with constraints in database models*

Domenico Beneventano[1], Sonia Bergamaschi[1], Stefano Lodi[2] and Claudio Sartori[2]

[1]Department of Engineering "Enzo Ferrari" University of Modena and Reggio Emilia, Via P.Vivarelli n.10, Modena
[2] Department of Computer Science and Engineering University of Bologna, Viale del Risorgimento n.2, Bologna

Abstract
Integrity constraints are rules which should guarantee the integrity of a database. Provided that an adequate mechanism to express them is available, the following question arises: is there any way to populate a database which satisfies the constraints supplied by a designer? Constraints are expressed in various fashions, depending on the data model: in logic environment they are usually some subset of first order logic formulas, SQL2 allows general forms of inclusion dependencies and predicates to be evaluated on row values, while in OODBMSs most of the stuff is often left to the writing of methods. The contribution of the present work is twofold: first, to extend the expressiveness of complex object data models to capture the semantics of a relevant set of state constraints; second, to provide a specialized reasoner based on the tableaux calculus which is able to check the coherence of an object database schema with constraints.

The inspiration for this scientific article, and more generally for the research line, stems from the collaboration with the German Research Center for Artificial Intelligence (*DFKI*), which introduced us to *Description Logics*, in a context where the majority of the database scientific community was using Datalog. In particular, the first and fundamental scientific article in which techniques based on *Description Logics*, developed in the field of Artificial Intelligence, were applied to the context of Databases, was written during the period when Prof. Bernhard Nebel was a visiting professor here in Italy [1].

The ultimate and refined research outcome on reasoning techniques with constraints in database models has been published on the IEEE Transactions on Knowledge and Data Engineering [2]. In this work we proposed: 1) two alternative formalisms, able to express a relevant set of state integrity constraints with a declarative style; 2) two specialized reasoners, based on the tableaux calculus, able to check the consistency of complex objects database schemata expressed with the two formalisms. The proposed formalisms share a common kernel, which supports complex objects and object identifiers, and which allows the expression of acyclic descriptions of classes, nested relations and views, built up by means of the recursive use of record, quantified set, and object type constructors and by the intersection, union, and complement operators. Furthermore, the kernel formalism allows the declarative formulation of

SEBD 2023: 31st Symposium on Advanced Database System, July 02–05, 2023, Galzignano Terme, Padua, Italy
* This research has been partially funded by the project Sistemi informatici e Calcolo Parallelo, sottoprogetto 5, obiettivo LOGIDATA+ of the National Research Council (CNR).
domenico.beneventano@unimore.it (D. Beneventano); sonia.bergamaschi@unimore.it (S. Bergamaschi); stefano.lodi@unibo.it (S. Lodi); claudio.sartori@unibo.it (C. Sartori)
www.dbgroup.unimore.it (D. Beneventano); www.dbgroup.unimore.it (S. Bergamaschi); www.unibo.it/sitoweb/stefano.lodi (S. Lodi); www.unibo.it/sitoweb/claudio.sartori (C. Sartori)
0000-0001-6616-1753 (D. Beneventano); 0000-0001-8087-6587 (S. Bergamaschi); 0000-0001-8861-6341 (S. Lodi); 0000-0003-4535-1026 (C. Sartori)

typing constraints and integrity rules. In order to improve the expressiveness and maintain the decidability of the reasoning activities, we extended the kernel formalism into two alternative directions. The first formalism, $\mathcal{OLCP}$, introduces the capability of expressing path relations. Because cyclic schemas are extremely useful, we introduce a second formalism, $\mathcal{OLD}$, with the capability of expressing cyclic descriptions but disallowing the expression of path relations. In fact, we showed that the reasoning activity in $\mathcal{OLCDP}$ (i.e., $\mathcal{OLCP}$ with cycles) is undecidable.

We also demonstrated how techniques based on *Description Logics* are highly valuable in another crucial activity of a DBMS: query optimization; the related work "Description logics for semantic query optimization in object-oriented database systems" has been published on the ACM Transactions on Database Systems [3].

References

[1] S. Bergamaschi, B. Nebel, Acquisition and validation of complex object database schemata supporting multiple inheritance, Appl. Intell. 4 (1994) 185–203. URL: https://doi.org/10.1007/BF00872108. doi:10.1007/BF00872108.

[2] D. Beneventano, S. Bergamaschi, S. Lodi, C. Sartori, Consistency checking in complex object database schemata with integrity constraints, IEEE Trans. Knowl. Data Eng. 10 (1998) 576–598. URL: https://doi.org/10.1109/69.706058. doi:10.1109/69.706058.

[3] D. Beneventano, S. Bergamaschi, C. Sartori, Description logics for semantic query optimization in object-oriented database systems, ACM Trans. Database Syst. 28 (2003) 1–50. URL: https://doi.org/10.1145/762471.762472. doi:10.1145/762471.762472.