On the Ordering of Rewrite Rules (Extended Abstract)

Joachim Kröger[1], Stefan Paul[2], and Andreas Heuer[1]

[1] University of Rostock, Computer Science Department, DB Research Group, D–18051 Rostock, Germany
{jo, heuer}@informatik.uni-rostock.de
http://wwwdb.informatik.uni-rostock.de/Research/CROQUE.engl.html
[2] c/o SAG Systemhaus GmbH, Brandstücken 18, D–22549 Hamburg, Germany
stp@software-ag.de

Abstract. The conceptual development of the rule–based component of the CROQUE query rewrite and optimization system led to the derivation of an ordering of the rules present in the rule base according to their "optimization potential" in order to increase the efficiency of the logical term rewriting [5]. This heuristic may indeed be used for any other rule–based optimizer, too.
The major contribution of our approach is a combination of three ideas: (1) limit the search space of query optimization by grouping and ordering rules (for rule–based optimizers), (2) use of "offline" pre–optimization ordering instead of dynamic ordering during the optimization process, (3) taking into consideration more than one ("n–best") alternatives for further evaluation.

1 Motivation

The CROQUE project[1] [4, 2] is concerned with different aspects of optimization and evaluation of object–oriented queries. Starting point of all our considerations are queries in ODMG's OQL ([1], formalized in [6]), that are first represented internally employing a hybrid approach of calculus and algebra [3] before being transformed using a rule– and cost–based optimizer.

In general, equivalence rules may be applied in both directions. Since in most cases a heuristically preferred direction of application can easily be distinguished, we only consider *directed* rules (transformation rules) in the framework of the CROQUE project. Certain normalization steps are separately realized first, splitting the rule set in smaller subsets. Thus, not all defined rules are part of the rewriting according to the described concept.

An "exhaustive matching" of *all* rules against *every* operator node does not make sense since most of the rules may not match successfully against most of

[1] CROQUE (*C*ost– and *R*ulebased *O*ptimization of object–oriented *QUE*ries) is a joint research project of the universities of Rostock and Konstanz funded by the German Research Association (DFG) under contracts He 1768/5–2 and Scho 554/1–2.

the considered nodes. Efficiency is increased by dividing the rule set up into rule classes using the root node of the left hand side of the rule for classification. In this way, the set of rules to be considered in a (sub–)query tree rewriting may be reduced to exactly one rule class by only one single function call. The overhead needed for pattern matching will hereby be cut down considerably.

2 Ordering Rewrite Rules

Realizing the real need of a heuristic search space pruning we developed a heuristic that is the present focus of our interest: per (sub–)query tree rewriting no more than the n *best* rules shall be applied. Therefore, we have to order the rules. Additionally, this strategy also succeeds in minimizing the number of pattern matcher calls. The ordering of rules is not a new idea but in other approaches the ordering has to be programmed in general or estimation of the rule ordering is done online. A somewhat more detailed discussion of other approaches comprising the systems COKO–KOLA, Volcano, EXODUS, Starburst, GOM, and Gral may be found in the full paper [5].

The realization of the rewriting in CROQUE is as follows. A given logical expression is rewritten by the rewrite engine according to a given strategy thus spanning a logical search space. Therefore, each call to the pattern matcher supplies a (sub–)query tree t and an ordered list r of rules. The elements of r are matched against t in order. User queries, successfully applied rules, and associated rewriting results (which altogether define the system's query load) are passed to the adaptation component which may then possibly influence the rule ordering.

In order to minimize the overhead we avoid an online ordering during the query evaluation, i.e. the rules are applied according to their order in the considered rule classes' list. An initial ordering of the rules is done only once in the beginning. Afterwards, the ordering is modified during distinguished adaptation phases. These steps may be done according to any policy, the only request we demand for is that the adaptation is done offline. By means of the adaptation, possible mistakes or inaccuracies of the initial ordering shall be remedied and additionally the ordering shall be adapted to cover the characteristics of the considered database, e.g. the used schema, present instances, and the observed query load.

2.1 Initial Rule Ordering

The initial ordering of the rules is realized according to a very simple method: two rules are comparable, i.e. may be ordered, if there exist a smallest common pattern that is contained in both rules. Applying both rules to this pattern will lead to two expressions that may easily be ranked "by a human query optimization expert" (e.g., the DBA). Moreover, the ordering is assumed to be transitive. Incomparable rules are ordered randomly in the initial ordering phase. The isolated rule orderings obtained in this way are integrated for every rule class by determining their topological order.

2.2 Rule Order Adaptation

The adaptation of the initial ordering is done by the help of selected statistics. Furthermore, a combination of benchmarks and specific user queries is utilized for an offline rule assessment. Statistics are mainly intended as a means to help ordering incomparable rules since a random ordering is not really satisfying. Rules that may be used very often with a noticeable success are ranked higher than rarely useable rules that only result in small cost reductions. So–called benchmarks are used to correct the ranking done by hand in the initial rule ordering phase.

3 Overall Optimization Process

Optimization in CROQUE is not restricted to algebraic rewriting. The rewriting of expressions is done in a hybrid approach consisting of a calculus and an algebra notation [3]. Rewriting calculus expressions is done in the same way as rewriting algebraic expressions by means of rules and pattern matching. Thus, the results are transferable. The use of heuristics therefore integrates well into the related work of the CROQUE project.

Evaluation of all our concepts will be done on the basis of our query optimizer prototype. Most parts of its implementation are already complete so that the optimizer is nearly fully operational. There will be further work in improving our heuristics. Above all, the (automatic) adaptation of the rule ordering will be a topic of our ongoing investigations as well as investigations about further opportunities for adaptation on other optimizer layers.

References

[1] R.G.G. Cattell, editor. *The Object Database Standard: ODMG–93, Release 1.2.* Morgan–Kaufmann, San Mateo, CA, 1996.

[2] D. Gluche, T. Grust, C. Mainberger, and M.H. Scholl. Incremental Updates for Materialized Views with User-Defined Functions. In *Proc. of the Fifth Int. Conference on Deductive and Object-Oriented Databases (DOOD'97), Montreux, Switzerland*, December 1997.

[3] T. Grust, J. Kröger, D. Gluche, A. Heuer, and M.H. Scholl. Query Evaluation in CROQUE — Calculus and Algebra Coincide. In *Proc. of the 15th British National Conference on Databases (BNCOD 15), London, UK, LNCS 1271, Springer*, pages 84–100, July 1997.

[4] A. Heuer and J. Kröger. Query Optimization in the CROQUE Project. In *Proc. of the 7th Int. Conference on Database and Expert Systems Applications (DEXA '96), Zurich, Switzerland, LNCS 1134, Springer*, pages 489–499, September 1996.

[5] J. Kröger, S. Paul, and A. Heuer. Query Optimization: On the Ordering of Rules. Preprint CS–09–98, CS Dept., University of Rostock, 1998.
Available at http://wwwdb.informatik.uni-rostock.de/~jo/CS–09–98.html

[6] H. Riedel and M.H. Scholl. A Formalization of ODMG Queries. In *Proc. of the 7th Int. Conference on Database Semantics (DS–7), Leysin, Switzerland*, October 1997.