

Query Optimization Using Multiple SIP Strategies

Andreas Behrend

Universität Bonn, Institut für Informatik III

Römerstr. 164, 53117 Bonn

behrend@cs.uni-bonn.de

1 Introduction

The magic sets rewriting technique (cf. [Ull89]) seems to be the most promising approach to evaluating queries bottom-up for database systems with a powerful view concept. This is in particular the case for systems which will implement the new SQL3 standard and hence will allow the definition of recursive views. It has been shown that bottom-up query evaluation via magic sets is basically equivalent to top-down approaches and for readability reasons we use Datalog expressions throughout the paper. The attractiveness of magic sets lies in its generality and efficiency. Several approaches to improve the magic set method have been proposed which are often applicable in special cases only, e.g. [SSES90] and [Sagiv90]. Another approach called *envelopes* [Sagiv90] is as general as the magic set method and can be better than magic sets in many cases. However, given a *query execution plan* obtained by a single magic set transformation, this plan is normally not reoptimized dynamically, i.e. the actual number of facts generated during query processing cannot be considered.

A magic set transformation is done with respect to a chosen sideways information passing strategy (SIP strategy) which indicates how bindings in the head of a rule are passed to the rule's body and in which order the body literals have to be evaluated. Applying the magic set transformation during query processing once again allows the use of a different SIP strategy which takes into account dynamic criteria as well, e.g. the size of derived relations or the number of accesses to relations. Since a chosen SIP strategy represents a special way to 'prove' a given query it is natural to ask whether it is possible to combine the effects of different SIP strategies leading to a more efficient 'proof process'. Consider for example the following Datalog rules for computing the transitive closure of a relation b :

$$\begin{aligned} p(X, Y) &\leftarrow b(X, Y) \\ p(X, Y) &\leftarrow b(X, Z), p(Z, Y) \end{aligned}$$

As stated in [Sagiv90], given a query $?-p(a, b)$ it is sometimes better to evaluate the more general query $?-p(a, Y)$ and to check whether b is in the answer as compared to evaluating $?-p(a, b)$ directly. For this example there are two possible full SIP strategies which evaluate the body literals from 'left-to-right' or from 'right-to-left'. But the magic set method with respect to these strategies leads already to corresponding proof processes like $(?-p(a, Y)$ and check b in $Y)$ and $(?-p(X, b)$ and check a in $X)$, that is, both do not really use the other binding respectively. Combining the two strategies, however, leads to a proof process which evaluates $?-p(a, Z)$ and $?-p(Z, b)$ until all solutions are found for one side. Thus, changing the SIP strategy during the materialization process corresponds to a bidirectional search. Reoptimization to adapt query

plans at runtime can then be seen as a weighted multidirectional search. This can reduce the overall number of generated facts and reach optimization effects which cannot be achieved by magic sets based on a single choice of SIP strategy. Of course, counting the number of facts is not a complete cost measure, but can be a good indication of the relative efficiency of a method.

2 A motivating example

Consider the following Datalog rules:

R:

$$q(X, Y) \leftarrow p(X, Y), r(Y, X)$$

$$\begin{array}{ll} p(X, Y) \leftarrow b1(X, Y) & r(Y, X) \leftarrow b2(Y, X) \\ p(X, Y) \leftarrow b1(X, Z), p(Z, Y) & r(Y, X) \leftarrow b2(Y, Z), r(Z, X) \end{array}$$

Relations b1 and b2 are base relations, r and p are derived and compute the transitive closure of b2 and b1, respectively. The relation q is the intersection of the transitive closures. Given the query ?-q(a,Y), one has to decide which SIP strategy would be the best to compute all relevant answers via the magic set method. There are $2^3 = 8$ rule sets with different orders of body literals for this example. Thus, any choice of a full SIP strategy leads to one of these eight possible rule sets. Consider for example the 'left-to-right' SIP strategy where the binding of X in the query is passed to p. Evaluation of the p-literal leads to a binding for the argument Y of r. This SIP strategy leads to the following magic set transformed rules (Note that the adornment b denotes a bound argument position, whereas f denotes a free argument position):

R¹:

$$q^{bf}(X, Y) \leftarrow m_q^{bf}(X), p^{bf}(X, Y), r^{bb}(Y, X)$$

$$\begin{array}{ll} p^{bf}(X, Y) \leftarrow m_p^{bf}(X), b1(X, Y) & p^{bf}(X, Y) \leftarrow m_p^{bf}(X), b1(X, Z), p^{bf}(Z, Y) \\ r^{bb}(Y, X) \leftarrow m_r^{bb}(Y, X), b2(Y, X) & r^{bb}(Y, X) \leftarrow m_r^{bb}(Y, X), b2(Y, Z), r^{bb}(Z, X) \end{array}$$

$$\begin{array}{ll} m_p^{bf}(X) \leftarrow m_q^{bf}(X) & m_r^{bb}(Y, X) \leftarrow m_q^{bf}(X), p^{bf}(X, Y) \\ m_p^{bf}(Z) \leftarrow m_p^{bf}(X), b1(X, Z) & m_r^{bb}(Z, X) \leftarrow m_r^{bb}(Y, X), b2(Y, Z) \end{array}$$

The query ?-q(a,Y) is represented by the 'seed' fact $m_q^{bf}(a)$. Suppose the base relations consist of the following tuples:

F:

$$\begin{array}{llllll} b1(a, b) & b1(a, c) & b1(c, c') & b1(c, c'') & b1(c', d') & b1(c', d'') \\ b2(e, f) & b2(f, a) & b2(c'', a) & & & \end{array}$$

Obviously, there is only one way how to connect the two transitive closures with the starting and ending value $X=a$, namely by means of $Y=c''$. While computing the implicit state S_D of the deductive database $D = \langle F, R^1 \rangle$ bottom-up, the entire transitive closure from the starting point a for the relation p is computed as well as seven m_r^{bb} -facts representing all resulting subqueries to the r relation for the solutions found for p. The entire number of generated facts is 28, produced in order to find the only solution $q^{bf}(a, c'')$. Since the transitive closure of b2 is much smaller than p it is natural to ask whether another choice of SIP strategy might lead to

fewer facts in order to answer the query. Consider for example a SIP strategy where the relevant part of r is computed before the relevant part of p leading to the following magic rules:

R^2 :

$$q^{bf}(X, Y) \leftarrow m_q^{bf}(X), r^{fb}(Y, X), p^{bb}(X, Y)$$

$$p^{bb}(X, Y) \leftarrow m_p^{bb}(X, Y), b1(X, Y)$$

$$r^{fb}(Y, X) \leftarrow m_r^{fb}(X), b2(Y, X)$$

$$r^{bb}(Y, X) \leftarrow m_r^{bb}(Y, X), b2(Y, X)$$

$$p^{bb}(X, Y) \leftarrow m_p^{bb}(X, Y), b1(X, Z), p^{bb}(Z, Y)$$

$$r^{fb}(Y, X) \leftarrow m_r^{fb}(X), b2(Y, Z), r^{bb}(Z, X)$$

$$r^{bb}(Y, X) \leftarrow m_r^{bb}(Y, X), b2(Y, Z), r^{bb}(Z, X)$$

$$m_p^{bb}(X, Y) \leftarrow m_q^{bf}(X), r^{fb}(Y, X)$$

$$m_p^{bb}(Z, Y) \leftarrow m_p^{bb}(X, Y), b1(X, Z)$$

$$m_r^{fb}(X) \leftarrow m_q^{bf}(X)$$

$$m_r^{bb}(Z, X) \leftarrow m_r^{fb}(X), b2(Y, Z)$$

$$m_r^{bb}(Z, X) \leftarrow m_r^{bb}(X, Y), b2(Y, Z)$$

The seed is again $m_q^{bf}(a)$. The implicit state S_D of the deductive database $D = \langle F, R^2 \rangle$ consists now of 31 generated facts. Obviously, the intuitive change of the SIP strategy does not provide a better way to compute the answers to the query $?-q(a, Y)$, although a smaller overall number of 7 answer facts will be generated by a fixpoint iteration process. The problem lies in the generation of many magic facts representing subqueries which cannot be in the answer set of the p relation since the bound arguments are not in the domain and hence not in the transitive closure of $b1$.

The 'best' SIP strategy for this example is to evaluate all body literals from 'right-to-left' leading to the following rules:

R^3 :

$$q^{bf}(X, Y) \leftarrow m_q^{bf}(X), r^{fb}(Y, X), p^{bb}(X, Y)$$

$$p^{bb}(X, Y) \leftarrow m_p^{bb}(X, Y), b1(X, Y)$$

$$r^{fb}(Y, X) \leftarrow m_r^{fb}(X), b2(Y, X)$$

$$p^{fb}(X, Y) \leftarrow m_p^{fb}(Y), b1(X, Y)$$

$$p^{bb}(X, Y) \leftarrow m_p^{bb}(X, Y), p^{fb}(Z, Y), b1(X, Z)$$

$$r^{fb}(Y, X) \leftarrow m_r^{fb}(X), r^{fb}(Z, X), b2(Y, Z)$$

$$p^{fb}(X, Y) \leftarrow m_p^{fb}(Y), p^{fb}(Z, Y), b1(X, Z)$$

$$m_p^{fb}(Y) \leftarrow m_p^{bb}(X, Y)$$

$$m_p^{bb}(X, Y) \leftarrow m_q^{bf}(X), r^{fb}(Y, X)$$

$$m_r^{fb}(X) \leftarrow m_q^{bf}(X)$$

Computing the implicit state S_D with $D = \langle F, R^3 \rangle$ would generate only 14 facts. Note that even with this choice of SIP strategy subqueries for the p relation are generated which cannot lead to any answers since the bound arguments are not in the domain of the p relation¹. However, the cross product obtained by generating all possible subqueries for p using the rule set R^2 is avoided. It is easy to show that for this example the magic set rewriting does not really provide a better way to find all the answers to the given query, since computing the two transitive closures and then the intersection of both will also generate 15 facts only. Moreover, for this example it is better not to apply a 'full SIP strategy', e.g. by using only r^{fb} and p^{fb} , which would lead to 8 facts only during the materialization process.

The three examples above showed that a chosen SIP strategy has a considerable effect on the overall number of generated facts, and it is not easy to decide whether a certain SIP strategy

¹It is quite simple to integrate a domain check into the rules but this kind of check is very limited and will be a side effect of the approach presented later.

is better than another one without evaluating both. The first strategy follows basically the original definition given by the schema developer. Although the number of rules after the magic set transformation has been applied is the smallest of the three examples, one has not considered the different sizes of the derived relations. In the second example we have tried to estimate the relation sizes by looking at the base relations. Since b_2 is much smaller than b_1 we used a SIP strategy which evaluates r before p . In spite of the very small number of answers for p and r , the computation of the complete transitive closure of b_1 within the subqueries led to a very expensive query evaluation process for this example. The 'best' SIP strategy in the third example, however, does not evaluate the body literals with the maximum number of bindings possible and yet leads to the smallest number of generated facts.

3 Partial materialization

The question arises how to avoid that a bad SIP strategy is chosen for query evaluation without knowing the best one. In general it is not possible to know in advance the number of facts that will be generated for evaluating a given query, and hence it is not possible to choose the 'best' SIP strategy statically. However, one can get additional information about the number of relevant facts during a query bottom-up or top-down query evaluation process. It seems promising to reoptimize queries when relation parameters change. If one wants to change a chosen SIP strategy at run-time, there are several aspects which ought to be considered:

- It is only necessary to adorn magic literals. The adorned answer relations can lead to a redundant storage of similar facts, e.g. $p^{bf}(a, b)$ and $p^{bb}(a, b)$, which moreover cannot be shared by several rule sets resulting from different SIP strategies.
- More general subqueries subsume subqueries which are less general, e.g. $m_{-p^{bf}}(a)$ subsumes $m_{-p^{bb}}(a, k)$ completely. These subqueries do not have to be evaluated.
- The use of different SIP strategies can only be valuable if the strategies correspond to different proof processes. This criterion is important to keep the number of SIP strategies to be considered small. Moreover, using different SIP strategies makes the sharing of subqueries and subanswers very important.
- The original rules should only be reoptimized after a considerable number of facts have been generated [Derr93]. Thus, we need a *selection function* to determine at what time a change to a another SIP strategy could be worthwhile.

We will now give an intuitive idea how a query evaluation process could work while considering the actual sizes of derived relations during a bottom-up fixpoint computation. The chosen selection function is very simple and determines the order of body literals according to the exact relation sizes. Consider again our sample rules for the query $?-q(a, Y)$. At first, none of the derived relations contains answers yet. Thus, we can choose p or r arbitrarily. Suppose we have chosen p to compute the first relevant answers. After applying the magic set transformation the fact $m_{-p^{bf}}(a)$ is generated in the first iteration round. Now the selection function changes the order of derived relations to 'r before p' since p contains one fact already. Note that we count the answer facts as well as the generated subqueries for a derived relation to get a better cost measure. In the following iteration round the fact $m_{-r^{fb}}(a)$ is generated. Again, the two derived relations have the same number of facts and we can choose any of them. Following the original

order of body literals we choose p one more. But a subquery for p has already been generated and hence, the selection function has to decide in which order the body literals of the p relation has to be evaluated. Consider again the corresponding transformed rules for p:

$$\begin{aligned} p(X, Y) &\leftarrow m_{\text{-}p^{bf}}(X), b1(X, Y) && m_{\text{-}p^{bf}}(a) \\ p(X, Y) &\leftarrow m_{\text{-}p^{bf}}(X), b1(X, Z), p(Z, Y). \end{aligned}$$

Since p is still smaller than b1, our selection function should place the p-literal before the b1-literal. However, given a subquery $m_{\text{-}p^{bf}}(_)$ the possible SIP strategy which evaluates p before b1 leading to the most general subquery $m_{\text{-}p^{ff}}$ subsumes the strategy where b1 is evaluated before p and hence can only be worse. Therefore, we keep the original order and in the next iteration round the following facts are generated:

$$m_{\text{-}p^{bf}}(b) \quad m_{\text{-}p^{bf}}(c) \quad p(a, b) \quad p(a, c)$$

Up till now, 5 facts have been generated for p and only one fact for r. Thus, our selection function chooses r and has now to decide which SIP strategy would be the best for evaluating the rules defining r. Since r is still smaller than b2 we change the original order of body literals and evaluate r before b2. During the next iteration round the following facts are generated:

$$m_{\text{-}r^{bb}}(b, a) \quad m_{\text{-}r^{bb}}(c, a) \quad r(c'', a) \quad r(f, a)$$

Note that the two answers are generated for the given subquery $m_{\text{-}r^{fb}}(a)$ whereas the two further subqueries result from applying the first two answers of p. Both derived relations consist now of 5 facts, and we choose again p for generating new facts. The next iteration will yield

$$m_{\text{-}p^{bf}}(c'') \quad m_{\text{-}p^{bf}}(c') \quad p(c, c'') \quad p(c, c').$$

Choosing r, will lead to $r(e, a)$ in the next iteration round. Since r is still smaller than p the selection function chooses r again for the next iteration. This time, however, no other facts can be derived for r. With the generated answers we can now obtain the single solution for q. Note that we have not evaluated all generated subqueries since the complete evaluation of the subquery $m_{\text{-}r^{fb}}(a)$ generated all possible values for Y in $?-q(a, Y)$ have been. The overall number of generated facts is 16. Hence for this example the method presented is almost as good as the best full SIP strategy. Moreover, we have presented a method which is as general as the magic set method and in addition a method which provides a good heuristic when the 'best' SIP strategy is not known.

References

- [Derr93] DERR, MARCIA A.: *Adaptive Query Optimization in a Deductive Database System*. CIKM 1993, Washington, DC, USA.
- [Sagiv90] SAGIV, Y.: *Is There Anything Better than Magic?* NACLPL, 1990: 235-254.
- [SSES90] SIPPY, S., SOISALON-SOININEN, E.: *Multiple SIP Strategies and Bottom-Up Adorning in Logic Query Optimization*. ICDT, 1990: 485-498.
- [Ull89] ULLMAN, J. D.: *Principles of Database and Knowledge-Base Systems*. Volume I and II, Computer Science Press, 1990.