Soft Stratification for Magic Set Based Query Evaluation in Deductive Databases

Andreas Behrend
Institute of Computer Science III, University of Bonn
Roemerstr. 164, D-53117 Bonn, Germany
behrend@cs.uni-bonn.de

ABSTRACT

In this paper we propose a new bottom-up query evaluation method for stratified deductive databases based on the Magic Set approach. As the Magic Sets rewriting may lead to unstratifiable rules, we propose to use Kerisit's weak consequence operator to compute the well-founded model of magic rules (guaranteed to be two-valued). We show that its application in combination with the concept weak stratification, however, may lead to a set of answers which is neither sound nor complete with respect to the well-founded model. This problem is cured by introducing the new concept soft stratification instead.

1. INTRODUCTION

The Magic Sets rewriting [3, 5] technique for query evaluation seems to be the most promising approach to evaluating database queries for database systems with a powerful view concept. This is in particular the case for systems which will implement the new SQL:1999 standard, and hence will allow the definition of recursive views. The attractiveness of this method lies in its generality and efficiency. Additionally, in [8, 12] it has been shown that the Magic Set method can improve the performance of nonrecursive queries as well. Thus, it seems worthwhile to implement a Magic Set transformation and a fixpoint based evaluation mechanism on top of a (non)recursive relational database system. In this paper we propose such an evaluation mechanism for stratifiable deductive databases based on the well-known and simple fixpoint operator by Tarski.

We focus on the use of Magic Set transformations on (function-free) stratifiable deductive databases consisting of $Datalog^{\neg}$ rules which are mainly used because of their syntactic simplicity. Although Magic Sets is sound and complete for stratifiable databases [10] its application may result in unstratifiable databases, such that a more general approach than iterated fixpoint computation is needed for determining the corresponding well-founded model [19]. The alternating fixpoint computation by Van Gelder [17, 18] or the weak stratification approach by Kerisit and Pugin [11], however, are not really efficient as they may compute many irrelevant facts during the course of a fixpoint computation. In addition, we show that the latter method may even lead to the computation of erroneous answer facts with respect to a given query and the corresponding well-founded model.

The structured bottom-up method by Balbin et al. in [1, 2], however, is complete and sound but because of its complexity difficult to implement. In addition, an implementation of this fixpoint approach may be not efficient as well, because it poses problems to relational optimizers in real database systems because of the entwined evaluation process. Therefore, we introduce the new concepts *soft stratification* and *soft consequence operator*. The latter is a variant of Kerisit's weak consequence operator. Together they provide a sound and complete evaluation method for determining the well-founded semantics of a Magic Set transformed database. This simple approach is easy to implement on top of an existing relational database system and allows further refinement during the subsequent relational optimization process. In addition, the approach represents an efficient evaluation procedure which can be applied to other transformation based approaches in deductive databases as well, e.g., update propagation methods [9], methods for view updating or transformation based methods for computing the well-founded model of general logic programs [6].

This paper is organized as follows: After introducing basic concepts and the Magic Sets method in the following two sections, we recall in section 4 the concepts weak stratification and weak consequence operator for query evaluation which are part of the Alexander method [16]. In section 5 we show by means of a counter example the erroneous derivations of this approach and introduce the concepts soft stratification and soft consequence operator instead. After proving the correctness of our approach, we present a comparison to other methods in section 6 and conclude the paper in section 7.

2. BASIC CONCEPTS

We consider a first order language with a universe of constants $U = \{a, b, c, \ldots\}$, a set of variables $\{X, Y, Z, \ldots\}$ and a set of predicate symbols $\{p, q, r \ldots\}$. A *term* is a variable or a constant (i.e., we restrict ourselves to function-free terms). Let p be an n-ary predicate symbol and t_i ($i = 1, \ldots, n$ and $n \geq 0$) terms then $p(t_1, \ldots, t_n)$ (or simply $p(\vec{t}\,)$) is denoted

PODS 2003, June 9-12, 2003, San Diego, CA.

atom. An atom is *ground* if every t_i is a constant. If A is an atom, we use $\texttt{pred}(A)$ to refer to the predicate symbol of A. A *fact* is a clause of the form $p(t_1, \ldots, t_n) \leftarrow true$ where $p(t_1, \ldots, t_n)$ is a ground atom. A *literal* is either an atom or a negated atom. A *rule* is a clause of the form $p(t_1, \ldots, t_n) \leftarrow L_1 \wedge \cdots \wedge L_m$ with $n \geq 0$ and $m \geq 1$ where $p(t_1, \ldots, t_n)$ is an atom denoting the rule's head, and $L_1, \ldots, L_m$ are literals representing its body. We assume all deductive rules to be *safe* (allowed or range-restricted, respectively); that is, all variables occurring in the head or in any negated literal of a rule must be present in some positive literal in its body as well. If A is the head of a given rule R, we use $\texttt{pred}(R)$ to refer to the predicate symbol of A. For a set of rules $\mathcal{R}$, $\texttt{pred}(\mathcal{R})$ is defined as $\cup_{r \in \mathcal{R}} \texttt{pred}(r)$.

Definition 1. *A deductive database $\mathcal{D}$ is a tuple $\langle \mathcal{F}, \mathcal{R} \rangle$ where $\mathcal{F}$ is a finite set of facts and $\mathcal{R}$ a finite set of rules such that $\texttt{pred}(\mathcal{F}) \cap \texttt{pred}(\mathcal{R}) = \emptyset$. Within a deductive database $\mathcal{D} = \langle \mathcal{F}, \mathcal{R} \rangle$, a predicate symbol p is called derived (view predicate), if $p \in \texttt{pred}(\mathcal{R})$. The predicate p is called extensional (or base predicate), if $p \in \texttt{pred}(\mathcal{F})$.*

For simplicity of exposition, and without loss of generality, we assume that a predicate is either base or derived, but not both, and that constants do neither occur in rule heads nor in body literals referring to a derived relation. Both conditions can be easily achieved by rewriting a given database. Before defining the semantics of a deductive database, we introduce the notion *stratification* using the so-called *predicate dependency graph* which describes dependencies between predicates in $\mathcal{R}$.

Definition 2 (Predicate Dependency Graph). *Let $\mathcal{D}$ be a deductive database and $Rel_{\mathcal{D}} = \texttt{pred}(\mathcal{F}) \uplus \texttt{pred}(\mathcal{R})$ the set of all predicate symbols in $\mathcal{D}$. The predicate dependency graph of $\mathcal{D}$ is the labelled directed graph $G_{\mathcal{D}} = \langle V, E \rangle$ where $V = Rel_{\mathcal{D}}$ and E is a set of labelled edges. With $p, q \in Rel_{\mathcal{D}}$, $R^+_{p,q} = \{A \leftarrow W \in \mathcal{R} \mid \texttt{pred}(A) = p$ and W contains a positive literal L with $\texttt{pred}(L) = q\}$ and $R^-_{p,q} = \{A \leftarrow W \in \mathcal{R} \mid \texttt{pred}(A) = p$ and W contains a negative literal L with $\texttt{pred}(L) = q\}$, E contains a negative edge (q, p, neg) iff $R^-_{p,q} \neq \emptyset$ and, E contains a positive edge (q, p, pos) iff $R^+_{p,q} \neq \emptyset$ and $(q, p, neg) \notin E$.*

We say that a predicate p depends positively on a predicate q iff the predicate dependency graph contains a path from q to p that includes no negative edge. A predicate p depends negatively on q iff the path includes at least one negative edge. We say that p and q are mutually dependent ($p \approx q$) iff p depends on q and q depends on p. With $p \in Rel_{\mathcal{D}}$ and $\texttt{dep}_{\mathcal{R}}(p) = \{q \in Rel_{\mathcal{D}} \mid p$ depends positively or negatively on $q\}$, we denote the set of rules $\texttt{def}_{\mathcal{R}}(p)$ by which relation p is defined as $\texttt{def}_{\mathcal{R}}(p) = \{R' \in \mathcal{R} \mid \texttt{pred}(R') \in \texttt{dep}_{\mathcal{R}}(p) \cup \{p\}\ \}$.

Definition 3 (Stratification). *Let $\mathcal{D}$ be a deductive database. A stratification λ on $\mathcal{D}$ is a mapping from the set of all predicate symbols $Rel_{\mathcal{D}}$ in $\mathcal{D}$ to the set of positive integers $\mathbb{N}$, such that for all predicate symbols $p, q \in Rel_D$:*

$$\begin{array}{lcl} p \text{ depends positively on } q & \Longrightarrow & \lambda(p) \geq \lambda(q) \\ p \text{ depends negatively on } q & \Longrightarrow & \lambda(p) > \lambda(q) \end{array}$$

The rule set $\mathcal{R}$ is then called stratified with respect to λ and all predicates p in $\mathcal{R}$ with the same value $\lambda(p)$ form a corresponding stratum. A database is called stratifiable iff at least one stratification exists.

Stratifiable rules do not allow recursion through negative predicate occurrences. A stratification partitions a given rule set such that all positive derivations of relations can be determined before a negative literal with respect to one of those relations is evaluated. Given a deductive database $\mathcal{D}$, the Herbrand base $\mathcal{H}_{\mathcal{D}}$ of $\mathcal{D}$ is the set of all ground atoms that can be constructed from the predicate symbols and constants occurring in $\mathcal{D}$. Any subset I of $\mathcal{H}_{\mathcal{D}}$ is a Herbrand interpretation of $\mathcal{D}$. Based on these notions we will now define the semantics of a deductive database. First, we recall the immediate consequence operator introduced by van Emden and Kowalski that will serve as the basic operator for determining the semantics of different classes of deductive databases.

Definition 4 (Immediate Consequence Operator). *Let $\mathcal{D} = \langle \mathcal{F}, \mathcal{R} \rangle$ be a deductive database. The immediate consequence operator $T_{\mathcal{R}}$ is a mapping on sets of ground atoms and is defined for $\mathcal{I} \subseteq \mathcal{H}_{\mathcal{D}}$ as follows:*

$$\begin{array}{ll} T_{\mathcal{R}}(\mathcal{I}) = \{A \mid & A \in \mathcal{I} \text{ or} \\ & \text{there exists a rule } A \leftarrow L_1 \wedge \ldots \wedge L_n \in [[\mathcal{R}]] \\ & \text{such that } L_i \in \mathcal{I} \text{ for all positive literals } L_i \\ & \text{and } L \notin \mathcal{I} \text{ for all negative literals } L_j \equiv \neg L\} \end{array}$$

where $[[\mathcal{R}]]$ denotes the set of all ground instances of rules in $\mathcal{R}$.

As the immediate consequence operator $T_{\mathcal{R}}$ is monotonic for semi-positive databases, i.e., databases in which negative literals reference base relations only, its least fixpoint lfp $(T_{\mathcal{R}}, \mathcal{F})$ exists, where lfp $(T_{\mathcal{R}}, \mathcal{F})$ denotes the least fixpoint of operator $T_{\mathcal{R}}$ containing $\mathcal{F}$, and coincides with the least Herbrand model $\mathcal{S}_{\mathcal{D}}$ of $\mathcal{D}$.

For stratifiable databases, however, the semantics is defined as the iterated fixpoint model $\mathcal{M}_{\mathcal{D}}$ which can be constructed as follows: Let $\mathcal{D} = \langle \mathcal{F}, \mathcal{R} \rangle$ be a stratifiable database and λ a stratification on $\mathcal{D}$. The partition $P_1 \uplus \ldots \uplus P_n$ of $\mathcal{R}$ defined by λ induces a sequence of least Herbrand models $M_1, \ldots M_n$:

$$\begin{array}{l} M_1 := \text{lfp } (T_{P_1}, \mathcal{F}), \\ M_2 := \text{lfp } (T_{P_2}, M_1), \\ \ldots \\ M_n := \text{lfp } (T_{P_n}, M_{n-1}) =: \mathcal{M}_{\mathcal{D}}. \end{array}$$

Given an arbitrary deductive database $\mathcal{D}$, its well-founded model $\mathcal{W}_{\mathcal{D}}$ then coincides with $\mathcal{S}_{\mathcal{D}}$ or $\mathcal{M}_{\mathcal{D}}$ if $\mathcal{D}$ is a semi-positive or stratifiable database, respectively. For illustrating the notations introduced above, consider the following example of a stratifiable deductive database $\mathcal{D} = \langle \mathcal{F}, \mathcal{R} \rangle$:

$$\begin{array}{llll} \underline{\mathcal{R}}: & h(X,Y) \leftarrow \neg p(Y,X) \wedge p(X,Y) & \underline{\mathcal{F}:} & e(1,2) \\ & p(X,Y) \leftarrow e(X,Y) & & e(2,1) \\ & p(X,Y) \leftarrow e(X,Z) \wedge p(Z,Y) & & e(2,3) \end{array}$$

Relation p represents the transitive closure of relation e while relation h selects all $p(X, Y)$-facts where Y is reachable from X but not vice versa. A stratification postpones the evaluation of h until all p tuples have been derived. The well-founded model is then given by $\mathcal{W}_{\mathcal{D}} = \mathcal{F} \cup \{p(1,1), p(1,2), p(1,3), p(2,1), p(2,2), p(2,3)\} \cup \{h(1,3), h(2,3)\}$.

3. MAGIC SETS

Various methods for efficient bottom-up evaluation of queries against the intensional part of a database have been proposed, e.g. Magic Sets [4], Counting [5], Alexander method [16]). All these approaches are rewriting techniques for deductive rules with respect to a given query such that bottom-up materialization is performed in a goal-directed manner cutting down the number of irrelevant facts generated. In the following we will focus on Magic Sets as this approach has been accepted as a kind of standard in the field. We refrain from presenting the Magic Sets approach in detail but rather present a simplified version of the Magic Templates algorithm [15].

Magic Sets rewriting is a two-step transformation in which the first phase consists of constructing an adorned rule set, and where the second phase is the actual Magic Sets rewriting. The adorned rule set is derived from the original database with respect to the binding pattern of the query and a choice of *sideways information passing strategy (sip)* strategies [15]. A SIP strategy determines for each rule the order in which the body literals are to be evaluated. As an example consider the following rule set

$$p(X,Y) \leftarrow e(X,Y) \qquad e(X,Y) \leftarrow b(X,Y)$$
$$p(X,Y) \leftarrow e(X,Z) \wedge p(Z,Y) \qquad e(X,Y) \leftarrow c(X,Y)$$

and the query $?-p(1,Y)$ asking for all nodes reachable from node 1. Within an adorned rule set each derived predicate is associated with an adornment, which is a string consisting of the symbols 'b' and 'f' representing bound and free argument positions when the predicate is evaluated. The adorned version of the deductive rules is constructed with respect to the adorned query $p_{bf}(1,Y)$ and the selected sip strategy. Assuming a left-to-right sip strategy for all rules, the adorned rule set of the example looks as follows:

$$p_{bf}(X,Y) \leftarrow e_{bf}(X,Y) \qquad e_{bf}(X,Y) \leftarrow b(X,Y)$$
$$p_{bf}(X,Y) \leftarrow e_{bf}(X,Z) \wedge p_{bf}(Z,Y) \quad e_{bf}(X,Y) \leftarrow c(X,Y)$$

In the course of a top-down evaluation of the query $p_{bf}(1,Y)$ each derived predicate would be called with the unique binding pattern encoded in its adornment when rule bodies are evaluated from left-to-right.

During the second phase of Magic Sets the adorned rules are rewritten such that bottom-up materialization of the resulting database simulates a top-down evaluation of the original query on the original database. For this purpose, each adorned rule is extended with a magic literal restricting the evaluation of the rule to the given binding in the adornment of the rule's head. The magic predicates are defined by rules computing all values that would be passed in the sequence of body literals according to the sip strategy. The initial values corresponding to the query are given by the so-called magic seed. Before we present the Magic Sets rewriting more precisely, the next definitions specify how magic literals are constructed and how the seed is derived from the query.

Definition 5 (Magic Predicates). *Let $A \equiv p_{ad}(\vec{x})$ be a positive literal with adornment* ad *and* $\texttt{bd}(\vec{x})$ *the sequence of variables within $\vec{x}$ indicated as bound in the adornment* ad*. Then the magic predicate of A is defined as*

$$\texttt{magic}(A) := m\_p_{ad}(\texttt{bd}(\vec{x})).$$

If $A \equiv \neg p_{ad}(\vec{x})$ is a negative literal, then the magic predicate of A is defined as $\texttt{magic}(A) := \neg\ m\_p_{ad}(\texttt{bd}(\vec{x}))$.

Definition 6 (Seed/Seed Rule). *Let $Q \equiv p_{ad}(\vec{c})$ be a query with adornment* ad *and $\texttt{bd}(\vec{c})$ the sequence of constants within $\vec{c}$ indicated as bound in the adornment* ad*. Then the seed of Q is defined as*

$$\texttt{seed}(Q) := m\_s\_p_{ad}(\texttt{bd}(\vec{c}))$$

and the corresponding seed rule is defined as

$$\texttt{seed\_rule}(Q) := m\_p_{ad}(\vec{x}) \leftarrow m\_s\_p_{ad}(\vec{x})$$

where $\vec{x}$ is a vector of distinct variables $x_1, \ldots, x_n$ and n is the length of the sequence $\texttt{bd}(\vec{x})$.

In order to simplify the definition of the Magic Sets transformation we assume that the body literals are already ordered from left to right according to the selected sip strategy.

Definition 7 (Magic Rules). *Let $\mathcal{R}$ be a stratifiable deductive rule set, $Q \equiv p_{ad}(\vec{c})$ an adorned query with $p \in \texttt{pred}(\mathcal{R})$, and $\mathcal{R}^Q$ the adorned rule set of $\mathcal{R}$ with respect to the query Q. The Magic Sets rewriting of $\mathcal{R}^Q$ yields the magic rules $\texttt{ms}(\mathcal{R}^Q)$ defined as the smallest set satisfying the following conditions:*

1. *For each deductive rule $A \leftarrow L_1 \wedge \ldots \wedge L_n \in \mathcal{R}^Q$ an answer rule of the form*

$$A \leftarrow \texttt{magic}(A) \wedge L_1 \wedge \ldots \wedge L_n$$

is in $\texttt{ms}(\mathcal{R}^Q)$.

2. *For each deductive rule $A \leftarrow L_1 \wedge \ldots \wedge L_n \in \mathcal{R}^Q$ and each derived body literal L_i $(1 \leq i \leq n)$ a sub-query rule of the form*

$$\texttt{magic}(L_i) \leftarrow \texttt{magic}(A) \wedge L_1 \wedge \ldots \wedge L_{i-1}$$

is in $\texttt{ms}(\mathcal{R}^Q)$.

Note that the definition of Magic Rules solely depends on the predicate p and adornment ad of a given query $p_{ad}(\vec{c})$ but not on the constants within $\vec{c}$.

Definition 8 (MAGIC DB TRANSFORMATION). *Let $\mathcal{D} = \langle \mathcal{F}, \mathcal{R} \rangle$ be a stratifiable deductive database, $Q \equiv p_{ad}(\vec{c})$ an adorned query with $p \in \texttt{pred}(\mathcal{R})$, and $\texttt{ms}(\mathcal{R}^Q)$ the magic rule set of $\mathcal{R}$ with respect to the query Q. The Magic DB transformation of $\mathcal{D}$ with respect to Q then yields the deductive database $\mathcal{D}^m = \langle \mathcal{F} \cup \{\texttt{seed}(Q)\}, \texttt{ms}(\mathcal{R}^Q) \cup \{\texttt{seed\_rule}(Q)\} \rangle$.*

Note that this definition of Magic Sets slightly differs from the one in [15] as we add the additional magic seed rule to $\texttt{ms}(\mathcal{R}^Q)$ in order to keep the condition $\texttt{pred}(\mathcal{F}) \cap \texttt{pred}(\mathcal{R}) = \emptyset$ in Definition 1 of deductive databases satisfied. For our example above, the Magic DB transformation then yields the deductive rule set

$$p_{bf}(X,Y) \leftarrow m\_p_{bf}(X) \wedge e(X,Y)$$
$$p_{bf}(X,Y) \leftarrow m\_p_{bf}(X) \wedge e(X,Z) \wedge p_{bf}(Z,Y)$$
$$e_{bf}(X,Y) \leftarrow m\_e_{bf}(X) \wedge b(X,Y)$$
$$e_{bf}(X,Y) \leftarrow m\_e_{bf}(X) \wedge c(X,Y)$$

$$m\_p_{bf}(Z) \leftarrow m\_p_{bf}(X) \wedge e(X,Z)$$
$$m\_p_{bf}(X) \leftarrow m\_s\_p_{bf}(X)$$
$$m\_e_{bf}(X) \leftarrow m\_p_{bf}(X)$$

as well as the magic seed fact $m\_s\_p_{bf}(1)$.

THEOREM 1. *Let $\mathcal{D}$ be a stratifiable database, Q a query to $\mathcal{D}$, $\mathcal{D}^m$ the database resulting from Magic DB transformation applied to $\mathcal{D}$ with respect to Q, and $\texttt{ans}(\mathcal{W}_\mathcal{D}, Q)$ the answer set of Q defined as $\texttt{ans}(\mathcal{W}_\mathcal{D}, Q) := \{L \mid L \equiv Q\sigma$, σ is a ground substitution for all variables in Q and $L \in \mathcal{W}_\mathcal{D}\}$. Then the answer set of Q with respect to $\mathcal{D}$ is equivalent to the answer set of the adorned query with respect to the rewritten database. Hence, if $Q \equiv p(\vec{c})$, then*

$$p(\vec{c})\sigma \in \texttt{ans}(\mathcal{W}_\mathcal{D}, Q) \Longleftrightarrow p_{ad}(\vec{c})\sigma \in \texttt{ans}(\mathcal{W}_{\mathcal{D}^m}, Q^a)$$

where σ is a ground substitution for the variables in Q and $Q^a \equiv p_{ad}(\vec{c})$ is the adorned query.

Proof See [10, 15].

In [10] it has been shown that the Magic Sets transformation is sound for stratifiable databases. However, the resulting rule set may be no more stratifiable and more general approaches than iterated fixpoint computation are needed. For determining the well-founded model [19] of general logic programs, the alternating fixpoint computation by Van Gelder [17, 18] or the conditional fixpoint by Bry [7] could be used. The application of these methods, however, is not really efficient because the specific reason for the unstratifiability of the transformed rule sets is not taken into account.

Therefore, other methods have been proposed in order to compute the semantics of unstratifiable databases resulting from a Magic Sets transformation explicitly. The structured bottom-up method proposed by Balbin et al. in [1, 2] realizes a bottom-up materialization process for the rewritten database which is suspended each time a negative literal $\neg A$ is queried with respect to a set of particular bindings. Then the query $?-A$ is evaluated by invoking an appropriate function call which actually performs an intermediate magic sets process initiated by corresponding magic seeds derived from the given bindings. Note that this function has to be recursive as the evaluation of the query $?-A$ itself may depend on the evaluation of other negative literals in deeper layers. Afterwards, the global process is continued and the answers for $?-A$ are used to evaluate the negative literal $\neg A$. The structured bottom-up method is complete and sound but because of its complexity difficult to implement.

The Alexander method proposed by Kerisit and Pugin in [11] represents a much simpler approach for computing the semantics of unstratifiable databases resulting from a Magic Sets transformation. This approach uses a modified consequence operator and a more general stratification concept, the so-called weak stratification, in order to evaluate negative literals correctly. In the sequel, we will concentrate on this approach because of its efficiency and simplicity.

4. QUERY EVALUATION USING WEAK STRATIFICATION

The definition of stratification requires two conditions with respect to positive and negative dependencies between predicates to be satisfied. The concept of weak stratification [11] relaxes these conditions by considering negative dependencies between predicates only.

DEFINITION 9 (WEAK STRATIFICATION). *Let $\mathcal{D}$ be a deductive database. A weak stratification λ^ω on $\mathcal{D}$ is a mapping from the set of all predicate symbols $Rel_\mathcal{D}$ in $\mathcal{D}$ to the set of positive integers $I\!N$, such that for all predicate symbols $p, q \in Rel_D$:*

$$p \text{ depends negatively on } q \quad \Longrightarrow \quad \lambda^\omega(p) > \lambda^\omega(q)$$

A weak stratification then induces a weak partition $\mathcal{P} = Pos \uplus N_1 \uplus \ldots \uplus N_n$ of $\mathcal{R}$ such that

1. *If $A \leftarrow W \subseteq \mathcal{R}$ is a positive rule (i.e., a rule with no negative body literals), then the rule $A \leftarrow W$ is in the set Pos.*
2. *If $A \leftarrow W \subseteq \mathcal{R}$ is a negative rule (i.e., a rule with at least one negative body literal) and $\lambda^\omega(A) = i$, then the rule $A \leftarrow W$ is in the set N_i.*

In [11] it has been shown that every rule set resulting from the Magic Sets transformation of a stratifiable rule set can be weakly stratified. For materializing weakly stratified databases, the authors propose a modified immediate consequence operator which we call weak consequence operator in the following.

DEFINITION 10 (WEAK CONSEQUENCE OPERATOR). *Let $\mathcal{D} = \langle \mathcal{F}, \mathcal{R} \rangle$ be a deductive database and λ^ω a weak stratification of $\mathcal{R}$ inducing the weak partition $\mathcal{P} = P_0 \uplus \ldots \uplus P_n$ of $\mathcal{R}$ with $P_0 = Pos$ and $P_i = N_i$ for $1 \leq i \leq n$. The weak consequence operator $T^\omega_\mathcal{P}$ is a mapping on sets of ground atoms and is defined for $\mathcal{I} \subseteq \mathcal{H}_\mathcal{D}$ as follows:*

$$T^\omega_\mathcal{P}(\mathcal{I}) := \begin{cases} \mathcal{I} & \text{if } \{j \mid T_{P_j}(\mathcal{I}) - \mathcal{I} \neq \emptyset\} = \emptyset \\ T_{P_i}(\mathcal{I}) & \text{with } i = \texttt{min}\{j \mid T_{P_j}(\mathcal{I}) - \mathcal{I} \neq \emptyset\}, \\ & \text{otherwise.} \end{cases}$$

As the weak consequence operator is monotonic, its least fixpoint exists and is given by lfp $(T^\omega_\mathcal{P}, \mathcal{F})$. It is obvious that the application of $T^\omega_\mathcal{P}$ can lead to more positive conclusions than there are within the set of positive elements of the corresponding well-founded model. In [11] the authors claim, however, that at least the answer relation with respect to a given query is correctly determined by means of lfp $(T^\omega_\mathcal{P}, \mathcal{F})$. We will show in the following section that this is not true and present a refined version of the concept weak stratification in order to determine the complete well-founded model correctly using the weak consequence operator.

5. SOFT STRATIFICATION

In general it is possible to find several distinct weak stratifications for a given rule set. However, not every chosen weak stratification may lead to correct derivations of facts with respect to the well-founded model if the weak consequence operator is applied. For illustrating this problem consider the following example of a stratifiable deductive database $\mathcal{D} = \langle \mathcal{F}, \mathcal{R} \rangle$

$\underline{\mathcal{R}}$: $p(X) \leftarrow b(X,Y,Z) \wedge \neg q(X) \wedge \neg q(Y) \wedge \neg q(Z)$
$q(X) \leftarrow d(X)$

$\underline{\mathcal{F}}$: $b(1,2,3)$ $d(2)$ $d(3)$

and the query $Q \equiv p(1)$. A weak partition $\mathcal{P} = Pos \uplus N_1$ of the Magic Sets transformed rule set $\mathtt{ms}(\mathcal{R}^Q) \uplus \{\mathtt{rule\_seed}(Q)\}$ could be as follows:

$\underline{Pos}$:

$q_b(X) \leftarrow m\_q_b(X) \wedge d(X)$
$m\_p_b(X) \leftarrow m\_s\_p_b(X)$
$m\_q_b(X) \leftarrow m\_p_b(X) \wedge b(X,Y,Z)$

$\underline{N_1}$:

$p_b(X) \leftarrow m\_p_b(X) \wedge b(X,Y,Z) \wedge \neg q_b(X) \wedge \neg q_b(Y) \wedge \neg q_b(Z)$
$m\_q_b(Y) \leftarrow m\_p_b(X) \wedge b(X,Y,Z) \wedge \neg q_b(X)$
$m\_q_b(Z) \leftarrow m\_p_b(X) \wedge b(X,Y,Z) \wedge \neg q_b(X) \wedge \neg q_b(Y)$

The magic seed is given by $m\_s\_p_b(1)$. Evaluating these rules using $T^{\omega}_{\mathcal{P}}$ would yield $\{m\_p_b(1)\}$, $\{m\_q_b(1)\}$, $\{p_b(1)$, $m\_q_b(2)$, $m\_q_b(3)\}$ and $\{q_b(2), q_b(3)\}$. With respect to the corresponding well-founded model $\mathcal{W}_{\mathcal{D}^m} := \{m\_p_b(1)$, $m\_q_b(1)$, $m\_q_b(2)$, $q(2)\} \cup \mathcal{F} \cup \{m\_s\_p_b(1)\}$ of $\mathcal{D}^m$, the facts $m\_q_b(3)$ and $q_b(3)$ are erroneous derivations. Additionally, the incorrect answer fact $p_b(1)$ is derived which is clearly wrong as relation p is empty in the iterated fixpoint model of the original database. The erroneous derivations are due to the fact that only negative dependencies are considered in weak partitions but no positive ones. It is necessary, however, to consider also those positive dependencies which ensure that all necessary derivations of query and answer facts have been made before a rule with a corresponding negative literal is evaluated.

A possible solution to this problem is to choose a weak partition in such a way that all rules on which a negative literal positively or negatively depends lie in deeper layers. Consider, for instance, the negative literal $\neg q_b(Y)$ in the rule for defining relation p. This literal also appears in the rule for defining $m\_q_b(Z)$ which ought to be applied after the rules

$$Pos \cup \{m\_q_b(Y) \leftarrow m\_p_b(X) \wedge b(X,Y,Z) \wedge \neg q_b(X)\}$$

have been considered in deeper layers by $T^{\omega}_{\mathcal{P}}$ in order to provide all necessary answer and sub-query facts. Additionally, for evaluating the negative literal $\neg q_b(Z)$ in the rule defining p, the rule

$$m\_q_b(Z) \leftarrow m\_p_b(X) \wedge b(X,Y,Z) \wedge \neg q_b(X) \wedge \neg q_b(Y)$$

must have been considered already in deeper layers. The following definition formalizes the dependency between literals and rules in a Magic Sets transformed rule set.

Definition 11 (Required Rules). *Let $\mathcal{R}$ be a set of stratifiable deductive rules, Q an adorned query, $\mathcal{R}^Q$ the adorned rule set of $\mathcal{R}$ with respect to Q and $\mathtt{ms}(\mathcal{R}^Q)$ the corresponding magic set transformed rules. For each rule $R_i \equiv A \leftarrow \mathtt{magic}(B), L_{i,1}, \ldots L_{i,l_i} \in \mathtt{ms}(\mathcal{R}^Q)$ $(i = 1, ..., |\mathtt{ms}(\mathcal{R}^Q)|)$ and each derived body literal $L_{i,j}$ $(j \in \{1, \ldots, l_i\})$ the set of required rules $req(L_{i,j})$ is defined as the smallest set satisfying the following conditions:*

1. *For each derived body literal $L_{i,k}$ $(1 \leq k \leq j)$ a sub-query rule of the form*

$$\mathtt{magic}(L_{i,k}) \leftarrow \mathtt{magic}(B) \wedge L_{i,1} \wedge \ldots L_{i,k-1}$$

   is in $req(L_{i,j})$.

2. *For each derived body literal $L_{i,k}$ $(1 \leq k \leq j)$ the magic rules $\mathtt{ms}(\mathtt{def}_{\mathcal{R}^Q}(\mathtt{pred}(L_{i,k}))$ are in $req(L_{i,j})$.*

As an example consider again the deductive database above and its adorned rule set $\mathcal{R}^Q$

$p_b(X) \leftarrow b(X,Y,Z) \wedge \neg q_b(X) \wedge \neg q_b(Y) \wedge \neg q_b(Z)$
$q_b(X) \leftarrow d(X)$

with respect to the query $Q \equiv p(1)$. Suppose the resulting magic sub-query rule

$$m\_q_b(Z) \leftarrow m\_p_b(X) \wedge b(X,Y,Z) \wedge \neg q_b(X) \wedge \neg q_b(Y)$$

for defining $m\_q_b(Z)$ has been numbered R_4. The set of required rules for the body literal $L_{4,3} \equiv \neg q_b(Y)$ within this rule then is given by

$$\begin{aligned} req(L_{4,3}) := \{\ & m\_q_b(X) \leftarrow m\_p_b(X) \wedge b(X,Y,Z), \\ & m\_q_b(Y) \leftarrow m\_p_b(X) \wedge b(X,Y,Z) \wedge \neg q_b(X), \\ & q_b(X) \leftarrow m\_q_b(X) \wedge d(X)\ \} \end{aligned}$$

while the latter rule results from the magic set transformation $\mathtt{ms}(\mathtt{def}_{\mathcal{R}^Q}(\mathtt{pred}(L_{4,2}))$ of literal $L_{4,2} \equiv \neg q_b(X)$ and from the magic set transformation $\mathtt{ms}(\mathtt{def}_{\mathcal{R}^Q}(\mathtt{pred}(L_{4,3}))$, respectively.

We will now introduce the notion soft stratification to denote a weak stratification which also takes the sets of required rules for negative literals into account.

Definition 12 (Soft Stratification). *Let $\mathcal{R}$ be a stratifiable deductive rule set and $\mathtt{ms}(\mathcal{R}^Q)$ the corresponding set of Magic Set transformed rules with respect to a given query Q. A soft stratification λ^s on $\mathtt{ms}(\mathcal{R}^Q)$ is a mapping from the set of rules $\mathtt{ms}(\mathcal{R}^Q)$ to the set of positive integers $I\!N$, such that for all negative rules $R_{neg} \in \mathtt{ms}(\mathcal{R}^Q)$ and all negative literals L of R_{neg}:*

$$R' \in \mathtt{ms}(\mathcal{R}^Q) \text{ and } R' \in req(L) \implies \lambda^s(R_{neg}) > \lambda^s(R')$$

In a soft stratification, positive as well as negative dependencies are considered, leading to a stronger condition in comparison to weak stratification on the subsequent rule partitioning. Stratification problems introduced by the Magic Sets transformation, however, are avoided because dependencies between rules and not between predicates (as in the original condition of stratification) are considered. In addition, only necessary dependencies between rules are taken into account in order to be most flexible in the relational reoptimisation phase. For materializing softly stratified databases, we use a slightly modified weak consequence operator which we call soft consequence operator in the following.

Definition 13 (Soft Consequence Operator). *Let $\mathcal{D} = \langle \mathcal{F}, \mathcal{R} \rangle$ be a deductive database, Q a query, $\mathcal{D}^m$ the database resulting from Magic DB transformation applied to $\mathcal{D}$ with respect to Q, and λ^s a soft stratification of $\mathtt{ms}(R^Q) \uplus \{\mathtt{seed\_rule}(Q)\}$ inducing the soft partition $\mathcal{P} = P_1 \uplus \ldots \uplus P_n$. The soft consequence operator $T^s_{\mathcal{P}}$ is a mapping on sets of* *ground atoms and is defined for* $\mathcal{I} \subseteq \mathcal{H}_{\mathcal{D}^m}$ *as follows:*

$$T^s_{\mathcal{P}}(\mathcal{I}) := \begin{cases} \mathcal{I} & \textit{if } \{j \mid T_{P_j}(\mathcal{I}) - \mathcal{I} \neq \emptyset\} = \emptyset \\ T_{P_i}(\mathcal{I}) & \textit{with } i = \mathtt{min}\{j \mid T_{P_j}(\mathcal{I}) - \mathcal{I} \neq \emptyset\} \\ & \textit{otherwise.} \end{cases}$$

In contrast to the weak consequence operator introduced before, this operator is defined for magic set transformed rule sets only. In addition, there is no distinction between positive and negative rule sets anymore.

As an example consider the following partition $\mathcal{P} = P_1 \uplus P_2 \uplus P_3 \uplus P_4$ of the Magic Sets transformed rule set $\mathtt{ms}(\mathcal{R}^Q) \uplus \{\mathtt{rule\_seed}(Q)\}$

P_1:

$m\_p_b(X) \leftarrow m\_s\_p_b(X)$
$m\_q_b(X) \leftarrow m\_p_b(X) \wedge d(X)$
$q_b(X) \leftarrow m\_q_b(X) \wedge d(X)$

P_2:

$m\_q_b(Y) \leftarrow m\_p_b(X) \wedge b(X,Y,Z) \wedge \neg q_b(X)$

P_3:

$m\_q_b(Z) \leftarrow m\_p_b(X) \wedge b(X,Y,Z) \wedge \neg q_b(X) \wedge \neg q_b(Y)$

P_4:

$p_b(X) \leftarrow m\_p_b(X) \wedge b(X,Y,Z) \wedge \neg q_b(X) \wedge \neg q_b(Y) \wedge \neg q_b(Z)$

which satisfies the condition of soft stratification. The determination of lfp $(T^s_{\mathcal{P}}, \mathcal{F})$ with $F = \{b(1,2,3), d(2), d(3)\}$ using the given soft partition then yields the correct well-founded model $\mathcal{W}_{\mathcal{D}^m} := \{m\_p_b(1), m\_q_b(1)\} \cup \{m\_q_b(2)\} \cup \{q(2)\} \cup \mathcal{F}$ of $\mathcal{D}^m$.

PROPOSITION 1. *Let* $\mathcal{D} = \langle \mathcal{F}, \mathcal{R} \rangle$ *be a stratifiable deductive database and* $\mathtt{ms}(\mathcal{R}^Q)$ *the corresponding set of Magic Sets transformed rules with respect to a given query* Q*. Then a soft stratification of* $\mathtt{ms}(\mathcal{R}^Q)$ *exists.*

PROOF: Suppose there is no soft stratification of the magic set transformed rules $\mathtt{ms}(\mathcal{R}^Q)$. Then there must be two rules $R_1 \in \mathtt{ms}(\mathcal{R}^Q)$ with a negative body literal L_1 and $R_2 \in \mathtt{ms}(\mathcal{R}^Q)$ with a negative body literal L_2 such that $R_2 \in req(L_1)$ and $R_1 \in req(L_2)$. Without loss of generality we can assume R_1 and R_2 to be answer rules because for every sub-query rule $R' \in \mathtt{ms}(\mathcal{R}^Q)$ with a derived body literal L' there exists a corresponding answer rule $R'' \in \mathtt{ms}(\mathcal{R}^Q)$ with a derived body literal L'' such that $req(L'') = req(L')$. Therefore, R_2 must be in the set $\mathtt{ms}(\mathtt{def}_{\mathcal{R}^Q}(\mathtt{pred}(L_1))$ and its corresponding adorned rule R_2^Q must be in $\mathtt{def}_{\mathcal{R}^Q}(\mathtt{pred}(L_1)$. As L_1 is a negative literal in R_1, R_2^Q then must depend negatively on R_1^Q. Analogously, you can show that the adorned rule R_1^Q must depend negatively on R_2^Q. Thus, the adorned rule set $\mathcal{R}^Q$ must be unstratifiable and subsequently $\mathcal{R}$, which contradicts the prerequisites of the proposition.

□

THEOREM 2. *Let* $\mathcal{D} = \langle \mathcal{F}, \mathcal{R} \rangle$ *be a stratifiable deductive database,* $\mathtt{ms}(\mathcal{R}^Q)$ *the corresponding set of Magic Sets transformed rules with respect to a given query* Q*, and* λ^s *a soft stratification on* $\mathtt{ms}(\mathcal{R}^Q) \cup \{\mathtt{seed\_rule}(Q)\}$ *inducing the soft partition* $\mathcal{P} = P_1 \uplus \ldots \uplus P_n$*. Evaluating these rules using the soft consequence operator* $T^s_{\mathcal{P}}$ *then always yields the correct well-founded model of* $\mathcal{D}^m = \langle \mathcal{F} \cup \{\mathtt{seed}(Q)\}, \mathtt{ms}(\mathcal{R}^Q) \cup \{\mathtt{seed\_rule}(Q)\} \rangle$*. Thus, the following holds:*

$$\mathtt{lfp}\ (T^s_{\mathcal{P}}, \mathcal{F} \cup \{\mathtt{seed}(Q)\}) = \mathcal{W}_{\mathcal{D}^m}.$$

PROOF: The theorem is shown by induction on the number l of components in the partition $\mathcal{P}$ induced by λ^s on $\mathtt{ms}(\mathcal{R}^Q) \cup \{\mathtt{seed\_rule}(Q)\}$. Without loss of generality we can assume that $\mathtt{seed\_rule}(Q) \in P_1$ because no other rule may derive facts unless this rule has been fired first. The application of $T^s_{\mathcal{P}}$ then starts again with the first component P_1 of partition $\mathcal{P}$.

Suppose that $l = 1$: All negative literals in P_1 have empty sets of required rules as they refer to base relations only. Hence, the well-founded model of the semi-positive rule set P_1 for an arbitrary fact base X is given by

$$\begin{aligned} \mathcal{W}_{\langle P_1, X \rangle} &= \text{lfp}\ (T_{P_1}, X) \\ &=_{def} T_{P_1}(T_{P_1}(\ldots T_{P_1}(X)\ldots)) \\ &=_{def} \text{lfp}\ (T^s_{P_1}, X) \end{aligned}$$

This holds in particular for the fact base $X = \mathcal{F} \cup \{\mathtt{seed}(Q)\}$.

Suppose that $l > 1$: Assuming

$$\mathtt{lfp}\ (T^s_{P_1 \uplus \ldots \uplus P_{l-1}}, X) = \mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_{l-1}, X \rangle}$$

holds for any fact base X, we have to show that

$$\mathtt{lfp}\ (T^s_{P_1 \uplus \ldots \uplus P_l}, X) = \mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_l, X \rangle}.$$

According to definition 13 the least fixpoint computation of $T^s_{P_1 \uplus \ldots \uplus P_l}$ with respect to the fact base $\mathcal{F} \cup \{\mathtt{seed}(Q)\}$ corresponds to the following sequence of separate fixpoint computations

$$\begin{aligned} F_1 &:= T_{P_l}[\text{lfp}\ (T^s_{P_1 \uplus \ldots \uplus P_{l-1}}, \mathcal{F} \cup \{\mathtt{seed}(Q)\})] \\ F_2 &:= T_{P_l}[\text{lfp}\ (T^s_{P_1 \uplus \ldots \uplus P_{l-1}}, F_1)] \\ &\ldots \\ F_m &:= T_{P_l}[\text{lfp}\ (T^s_{P_1 \uplus \ldots \uplus P_{l-1}}, F_{m-1})], \end{aligned}$$

performed until no more new facts can be derived; that is $F_m = F_{m+1}$. Using the induction hypothesis, the fixpoint computations of partition $P_1 \uplus \ldots \uplus P_{l-1}$ with respect to the different base facts F_i $(1 \leq i \leq m)$ are correct and therefore coincide with the corresponding well-founded models. Thus, we have

$$\begin{aligned} F_1 &= T_{P_l}(\mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_{l-1}, \mathcal{F} \cup \{\mathtt{seed}(Q)\} \rangle}) \\ F_2 &= T_{P_l}(\mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_{l-1}, F_1 \rangle}) \\ &\ldots \\ F_m &= T_{P_l}(\mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_{l-1}, F_{m-1} \rangle}). \end{aligned}$$

Let us suppose that $\mathtt{lfp}\ (T^s_{P_1 \uplus \ldots \uplus P_l}, \mathcal{F} \cup \{\mathtt{seed}(Q)\}) \subseteq \mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_l, \mathcal{F} \cup \{\mathtt{seed}(Q)\} \rangle}$ does not hold. Then there must be a fact $f \equiv p(\vec{c})$ and a set of base facts F_j with $j \in \{1, \ldots, m\}$ such that $f \in F_j$ and $f \notin \mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_l, \mathcal{F} \cup \{\mathtt{seed}(Q)\} \rangle}$. As the computations of the well-founded models with respect to partition $P_1 \uplus \ldots \uplus P_{l-1}$ are correct, there must be a rule $R \in P_l$ with $\mathtt{pred}(f) = \mathtt{pred}(R)$ such that the application of R leads to the erroneous derivation of f. On the one hand, any negative literal in the body of R is evaluated correctly because its corresponding req-set is complete (see [10, 15]) and only consists of rules located in components $P_1 \ldots P_{l-1}$ (because of the soft stratification property) whose corresponding well-founded model is determined correctly according to the induction hypothesis. On the other hand, any positive literal is also evaluated correctly, as there is only one application of T_{P_l} and therefore every substitution must have come from the previously determined well-founded model $\mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_{l-1}, F_{j-1}\rangle}$. But then it can be concluded that the erroneous derivation f may only be due to an erroneous fact base F_{j-1}. Analogously, it can be followed that F_1 must have been an erroneous fact base and because of the correct application of T_{P_l} the well-founded model $\mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_{l-1}, \mathcal{F}\cup\{\mathtt{seed}(Q)\}\rangle}$ must have been incorrect which is a contradiction to the induction hypothesis.

Let us suppose that $\mathtt{lfp}\ (T^s_{P_1 \uplus \ldots \uplus P_l}, \mathcal{F} \cup \{\mathtt{seed}(Q)\}) \supseteq \mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_l, \mathcal{F}\cup\{\mathtt{seed}(Q)\}\rangle}$ does not hold. Then there must be a fact $f \equiv p(\vec{c})$ such that $f \in \mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_l, \mathcal{F}\cup\{\mathtt{seed}(Q)\}\rangle}$ and $f \notin F_m$. As the computations of the well-founded models with respect to partition $P_1 \uplus \ldots \uplus P_{l-1}$ are correct, there must be a rule $R \in P_l$ with $\mathtt{pred}(f) = \mathtt{pred}(R)$ such that the final application of T_{P_l} with respect to $\mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_{l-1}, F_{m-1}\rangle}$ does not derive f (which must be erroneous because of $F_m = F_{m+1}$). Analogously to the previous case, we can assume that all positive as well as all negative literals within the body of R are correctly evaluated over $\mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_{l-1}, F_{m-1}\rangle}$. Therefore, the previously determined fact base F_{m-1} cannot be correct. Analogously it can be followed again that the computation of F_1 must have been erroneous and subsequently the well-founded model $\mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_{l-1}, \mathcal{F}\cup\{\mathtt{seed}(Q)\}\rangle}$ must have been incorrect. This again contradicts our induction hypothesis.

Thus, we conclude $\mathtt{lfp}\ (T^s_{P_1 \uplus \ldots \uplus P_l}, \mathcal{F} \cup \{\mathtt{seed}(Q)\}) = \mathcal{W}_{\langle P_1 \uplus \ldots \uplus P_l, \mathcal{F}\cup\{\mathtt{seed}(Q)\}\rangle}$. □

6. DISCUSSION

For illustrating our approach, let us consider the following example of a stratifiable database $\mathcal{D} = \langle \mathcal{F}, \mathcal{R}\rangle$ with

$\underline{\mathcal{R}:}$	$\underline{\mathcal{F}:}$
$i(X) \leftarrow \neg s(X) \wedge j(X,Y) \wedge i(Y)$	$k(8),\ k(9)$
$i(X) \leftarrow k(X)$	$j(6,4),\ j(7,4),\ j(4,8)$
$s(X) \leftarrow b(X,Y) \wedge s(Y)$	$g(3),\ g(5)$
$s(X) \leftarrow g(X)$	$b(1,2),\ b(2,3),\ b(4,5).$

and the query $Q \equiv i(6)$. The transformed rules after applying Magic Sets are

$$\begin{array}{l} i_b(X) \leftarrow m\_i_b(X) \wedge \neg s_b(X) \wedge j(X,Y) \wedge i_b(Y) \\ i_b(X) \leftarrow m\_i_b(X) \wedge k(X) \\ s_b(X) \leftarrow m\_s_b(X) \wedge b(X,Y) \wedge s_b(Y) \\ s_b(X) \leftarrow m\_s_b(X) \wedge g(X) \end{array}$$

$$\begin{array}{l} m\_i_b(X) \leftarrow m\_s\_i_b(X) \\ m\_i_b(Y) \leftarrow m\_i_b(X) \wedge \neg s_b(X) \wedge j(X,Y) \\ m\_s_b(X) \leftarrow m\_i_b(X) \\ m\_s_b(Y) \leftarrow m\_s_b(X) \wedge b(X,Y). \end{array}$$

The following negative cycle then can be found in the corresponding dependency graph:

$$s_b \xrightarrow{neg} m\_i_b \xrightarrow{pos} m\_s_b \xrightarrow{pos} s_b$$

For computing the well-founded model of the Magic Set transformed database $\mathcal{D}^m := \langle \mathcal{F} \cup \{\mathtt{seed}(Q)\}, \mathtt{ms}(\mathcal{R}^Q) \cup \{\mathtt{seed\_rule}(Q)\}\rangle$ we compare the alternating fixpoint computation by Van Gelder and the structured bottom-up method by Balbin et al. with our approach based on soft stratification. First let us trace the alternating fixpoint computation as introduced in [17]. We begin with an empty set of negative conclusions $\tilde{I}_0 = \emptyset$ such that initially only positive rules have a chance to fire. Thus, the application of the proposed stability transformation yields the first set of positive conclusions

$$F_1 := \mathtt{S}_{D^m}(\tilde{I}_0) = \mathcal{F} \cup \{\mathtt{seed}(Q)\} \cup \{m\_i_b(6),\ m\_s_b(6)\}.$$

The sets of negative conclusions $\tilde{I}_i$ are obtained by complementing F_i, e.g.:

$$\tilde{I}_1 := \neg \cdot (\mathcal{H}_{D^m} \setminus F_1).$$

In general the sets F_i and $\tilde{I}_i$ alternately underestimate and overestimate the sets of positive and negative conclusions, respectively. The subsequent applications of $\mathtt{S}_{D^m}$ produce the following sequence of negative and positive conclusions:

$$\begin{array}{l} \tilde{I}_1 = \{\neg s(1), \neg s(2), \neg s(3), \neg s(4), \neg s(5), \neg s(6), \ldots\} \\ F_2 := \mathtt{S}_{D^m}(\tilde{I}_1) = \mathcal{F}_1 \cup \{m\_i_b(4),\ m\_i_b(8)\} \\ \qquad\qquad \cup \{m\_s_b(4), m\_s_b(5), m\_s_b(8)\} \\ \qquad\qquad \cup \{s_b(4), s_b(5), i_b(4), i_b(8)\} \end{array}$$

$$\begin{array}{l} \tilde{I}_2 := \{\neg s(1), \neg s(2), \neg s(3), \neg s(6), \ldots\} \\ F_3 := \mathtt{S}_{D^m}(\tilde{I}_1) = \mathcal{F}_1 \cup \{m\_i_b(4), m\_s_b(4), m\_s_b(5)\} \\ \qquad\qquad \cup \{s_b(4), s_b(5)\} \end{array}$$

$$\begin{array}{l} \tilde{I}_3 = \tilde{I}_2 \\ F_4 = F_3 \end{array}$$

As no more negative as well as positive conclusions can be derived, a fixpoint has been reached. This total alternating fixpoint model $A^+ = \mathcal{F} \cup \{\mathtt{seed}(Q)\} \cup \{m\_i_b(4), m\_i_b(6), m\_s_b(4), m\_s_b(5), m\_s_b(6), s_b(4), s_b(5)\}$ coincides with the corresponding well-founded model.

Let us compare this result with the application of the soft consequence operator. The set of required rules for the body literal $\neg s_b(X)$ within the answer rule for defining i_b is

$$\begin{array}{rl} req(\neg s_b(X)) := \{ & m\_s_b(X) \leftarrow m\_i_b(X), \\ & s_b(X) \leftarrow m\_s_b(X) \wedge b(X,Y) \wedge s_b(Y), \\ & s_b(X) \leftarrow m\_s_b(X) \wedge g(X), \\ & m\_s_b(Y) \leftarrow m\_s_b(X) \wedge b(X,Y)\ \} \end{array}$$

which coincides with the required rule set of the corresponding negative body literal within the sub-query rule for defining m_i_b. The following partition $\mathcal{P} = P_1 \uplus P_2 \uplus P_3$ of the Magic Sets transformed rule set $\mathtt{ms}(\mathcal{R}^Q) \uplus \{\mathtt{rule\_seed}(Q)\}$ satisfies the condition of soft stratification:

$\underline{P_1}$:

$$\begin{array}{l} m\_s_b(X) \leftarrow m\_i_b(X) \\ s_b(X) \leftarrow m\_s_b(X) \wedge b(X,Y) \wedge s_b(Y) \\ s_b(X) \leftarrow m\_s_b(X) \wedge g(X) \\ m\_s_b(Y) \leftarrow m\_s_b(X) \wedge b(X,Y) \end{array}$$

$\underline{P_2}$:

$$i_b(X) \leftarrow m\_i_b(X) \wedge \neg s_b(X) \wedge j(X,Y) \wedge i_b(Y)$$
$$m\_i_b(Y) \leftarrow m\_i_b(X) \wedge \neg s_b(X) \wedge j(X,Y)$$

$\underline{P_3}$:

$$m\_i_b(X) \leftarrow m\_s\_i_b(X)$$
$$i_b(X) \leftarrow m\_i_b(X) \wedge k(X)$$

The computation of lfp $(T^s_{\mathcal{P}}, \mathcal{F})$ induces the following sequence of sets:

$$\begin{aligned}
F_1 &:= \mathcal{F} \cup \{\texttt{seed}(Q)\}\\
F_2 &:= T_{P_3}(F_1) = \{m_i_b(6)\}\\
F_3 &:= T_{P_1}(F_2) = \{m_s_b(6)\}\\
F_4 &:= T_{P_2}(F_3) = \{m_i_b(4)\}\\
F_5 &:= T_{P_1}(F_4) = \{m_s_b(4)\}\\
F_6 &:= T_{P_1}(F_5) = \{m_s_b(5)\}\\
F_7 &:= T_{P_1}(F_6) = \{s_b(5)\}\\
F_8 &:= T_{P_1}(F_7) = \{s_b(4)\}\\
F_9 &= F_8
\end{aligned}$$

This result coincides with the total alternating fixpoint model. As this computation is strictly monotonic, any overestimations are avoided. That is, in contrast to the alternating fixpoint computation the facts $m\_i_b(8), m\_s_b(8), i_b(8), i_b(4)$ are not derived. In addition, it is possible to apply a semi-naive evaluation method in order to avoid the recomputation of certain facts. A possible drawback of our approach could be the expensive search for the next partition set to be applied which might require testing all 'lower' partitions. This can be partly avoided by providing additional information on literal dependencies in order to obviate the consideration of partition sets which cannot be affected by newly derived facts.

The structured bottom-up method by Balbin et al. [2] uses a function $eval/2$ for evaluating the Magic Set transformed rule set. Every time a negative literal is considered, the function $eval/2$ is recursively called for performing a local fixpoint computation over the relevant portion of the Magic Set transformed rules. The nested fixpoint computations terminate as soon as no more facts can be added to the global database state M_e. In our example the evaluation process starts with $M_e := \mathcal{F} \cup \{m\_i_b(6)\}$ and the function call $eval(i_b, M_e)$. The overall evaluation process then looks as follows:

$$\begin{array}{l}
eval(i_b, \mathcal{F} \cup \{m_i_b(6)\})\\
M_e := M_e \cup \mathcal{F} \cup \{m_i_b(6)\}
\end{array}$$

$$\left.\begin{array}{l}
eval(s_b, \{m_s_b(6)\})\\
\quad M_e := M_e \cup \{m_s_b(6)\}\\
\quad \ldots\\
\quad M_e := M_e \cup \emptyset\\
eval(s_b, \{m_s_b(6)\})\\
\quad M_e := M_e \cup \{m_s_b(6)\}\\
\quad \ldots\\
\quad M_e := M_e \cup \emptyset\\
M_e := M_e \cup \{m_i_b(4)\}
\end{array}\right\} \quad \texttt{I}$$

$$\left.\begin{array}{l}
eval(s_b, \{m_s_b(4)\})\\
\quad M_e := M_e \cup \{m_s_b(4)\}\\
\quad \ldots\\
\quad M_e := M_e \cup \{m_s_b(5),\ s_b(4),\ s_b(5)\}\\
eval(s_b, \{m_s_b(4)\})\\
\quad M_e := M_e \cup \{m_s_b(4)\}\\
\quad \ldots\\
\quad M_e := M_e \cup \{m_s_b(5),\ s_b(4),\ s_b(5)\}\\
M_e := M_e \cup \emptyset
\end{array}\right\} \quad \texttt{II}$$

Within the top level function call two iteration rounds `I` and `II` can be identified each performing a separate fixpoint computation for the two negative queries against s_b. As each negative (derived) literal causes a separate function call, many facts are repeatedly computed in the example above. In addition this separation of context makes it difficult or even impossible to apply further rule optimization techniques as proposed, e.g., in [13, 14].

Of course, for finite Herbrand universes and fixed rule sets, any of the above mentioned approaches requires time polynomial in the size of the Herbrand universe. The actual efficiency therefore strongly depends on the chosen implementation and the applied optimization techniques. The discussion above, however, already indicates some principle problems of the alternative approaches in contrast to our proposed method.

7. CONCLUSION

In this paper, we have presented a new evaluation method for computing the well-founded model of Magic Set transformed deductive databases. We solved a stratification problem which may arise when Magic Sets is applied to a stratifiable deductive rule set. Our approach represents a refinement of the weak stratification approach eliminating some of its deficiencies. At the same time, our method can be easily combined with other transformation based approaches in order to increase efficiency.

Acknowledgments

I like to thank Christian Dorau and Rainer Manthey for insightful discussions.

8. REFERENCES

[1] Balbin, I., Meenakshi, K., Ramamohanarao, K.: *A Query Independent Method for Magic Set Computation on Stratified Databases.* Fifth Generation Computer Systems 1988: 711-718.

[2] Balbin, I., Port, G. S., Ramamohanarao, K., Meenakshi, K.: *Efficient Bottom-UP Computation of Queries on Stratified Databases.* JLP 11(3&4): 295-344 (1991).

[3] Bancilhon, F., Maier, D., Sagiv, Y., Ullman,J. D.: *Magic Sets and Other Strange Ways to Implement Logic Programs.* PODS 1986: 1-16.

[4] Bancilhon, F., Ramakrishnan, R.: *An Amateur's Introduction to Recursive Query Processing Strategies.* SIGMOD Conference 1986: 16-52.

[5] Beeri, C., Ramakrishnan, R.: *On the Power of Magic*. JLP 10(1/2/3&4): 255-299 (1991).

[6] Behrend, A.: *Efficient Computation of the Well-Founded Model Using Update Propagation*. LPAR 2001: 422-437.

[7] Bry, F.: *Logic Programming as Constructivism: A Formalization and its Application to Databases*. PODS 1989: 34-50.

[8] Gupta, A., Mumick, I. S.: *Magic-sets Transformation in Nonrecursive Systems*. PODS 1992: 354-367.

[9] Griefahn, U.: *Reactive Model Computation - A Uniform Approach to the Implementation of Deductive Databases*. Dissertation, University of Bonn, 1997.

[10] Kemp, D., Srivastava, D., Stuckey, P.: *Bottom-Up Evaluation and Query Optimization of Well-Founded Models*. TCS 146(1 & 2): 145-184 (1995).

[11] Kerisit, J.-M., Pugin, J.-M.: *Efficient Query Answering on Stratified Databases*. Fifth Generation Computer Systems 1988: 719-726.

[12] Mumick, I. S., Finkelstein, S. J., Pirahesh, H., Ramakrishnan, R.: *Magic is Relevant*. SIGMOD Conference 1990: 247-258.

[13] Naughton, J. F., Ramakrishnan, R., Sagiv, Y., Ullman, J. D.: *Efficient Evaluation of Right-, Left-, and Multi-Linear Rules*. SIGMOD Conf. 1989: 235-242.

[14] Naughton, J. F., Ramakrishnan, R., Sagiv, Y., Ullman, J. D.: *Argument Reduction by Factoring*. TCS 146(1&2): 269-310, 1995.

[15] Ramakrishnan, R.: *Magic Templates: A Spellbinding Approach to Logic Programs*. JLP 11(3&4): 189-216 (1991).

[16] Rohmer, J., Lescoeur, R., Kerisit, J.-M.: *The Alexander Method - A Technique for the Processing of Recursive Axioms in Deductive Databases*. New Generation Computing 4(3): 273-285 (1986).

[17] Van Gelder, A.: *The Alternating Fixpoint of Logic Programs with Negation*. PODS 1989: 1-10.

[18] Van Gelder, A.: *The Alternating Fixpoint of Logic Programs with Negation*. JCSS 47(1): 185-221 (1993).

[19] Van Gelder, A., Ross, K., Schlipf, J.: *The Well-Founded Semantics for General Logic Programs*. JACM 38(3): 620-650 (1991).