Updates in Disjunctive Deductive Databases: A Minimal Model Based Approach

Adnan H. Yahya

Electrical Engineering Department, Birzeit University, Palestine
Email: yahya@ee.birzeit.edu

Extended Abstract

Abstract. The issue of updates in Disjunctive Deductive Databases (DDDBs) under the minimal model semantics is addressed. We consider ground clause addition and deletion in a DDDB. The approach of this paper is based on manipulating the clauses of the theory to produce the required change to the minimal model structure necessary to achieve the clause addition/deletion update. First we deal with ground positive clause updates in ground DDDBs. Later we consider positive, then general, clause addition/deletion in the class of range restricted DDDBs. When we give more than one algorithm for a case we comment on the comparative merits and limitations of each. We use the freedom offered by the multiple possibilities for achieving an update to select the one with the least change to the minimal model structure of the theory. We argue that such minimality is desirable if one interprets the minimal model structure as representing the possible states of the modeled world and therefore an update must affect them minimally.

keywords: Disjunctive Deductive Databases (DDDBs), Database Updates, Minimal Model Semantics, Minimal Model Generation.

1 Introduction

As a routine task in database maintenance, the update problem was extensively covered in the literature [4, 2, 5, 6, 13, 8, 9, 10, 14]. Different semantics were suggested and several methods for accomplishing updates were advanced. Since more than one way may be available to accomplish an update, minimality under certain criteria can be used to decide in favor of a particular choice.

Minimal models play an important role in understanding disjunctive deductive databases (DDDBs). Given that the minimal model structure is used to define the semantics of a DDDB, it is natural to specify updates in terms of changes to the minimal model structure. Update minimality in this case may be interpreted as the least change to the minimal model structure of the theory which need not be minimal in terms of the change to the clausal structure. It is known that minor changes to the clausal structure can reflect drastically on the minimal model structure of a theory and vice versa. Of interest also is the study of the change to the clausal structure so that certain changes to the minimal model structure of the database can be achieved.

In this paper we address the issue of executing clause addition/deletion updates using an approach based on manipulating the minimal model structure of the theory. We present several methods to deal with the update problem and show the merits and limitations of each. We define update minimality in terms of the effect on the minimal model structure of the theory.

The rest of the paper is organized as follows. In the next section we give some definitions and background material. In section 3 we address the issue of updating ground DDDBs through isolating the component of the theory that needs to be modified to achieve the required change to the minimal model structure. In section 4 we treat the class of range restricted DDDBs: a larger and more natural class of DDDBs and offer 2 algorithms to accomplish each type of update. In section 5 we attempt to extend our results to accommodate nonpositive clause updates. We comment on the possible interpretations of such updates and the way these interpretations can influence the update algorithms. In the last section we compare our approach with others reported in the literature, make some comments and point to possible directions for further research.

For space considerations we deleted all the proofs. They will appear in the full version of this paper[1].

2 Background and Notation

We adopt the usual notation relating to DDDBs as, e.g., in [11]. Here we limit ourselves to the basic material needed for presenting the results of this paper.

Definition 1. (DDDB) A *disjunctive deductive database (DDDB)*, DB, is a set of clauses in implication form: $C = A_1 \vee \cdots \vee A_m \leftarrow B_1 \wedge \ldots \wedge B_n$, where $m, n \geq 0$ and the A_i and B_j are atoms in a First Order Language (FOL) $\mathcal{L}$. C is positive if $n = 0$ (that is, the body is $\top$, *true*, empty) and negative or denial if $m = 0$ (that is, the head is $\perp$, *false*, empty). By $Head(C)$ we denote the disjunction of atoms $A_1 \vee \cdots \vee A_m$ and by $Body(C)$ we denote the conjunction of atoms $B_1 \wedge \ldots \wedge B_n$. So $C = Head(C) \leftarrow Body(C)$.

[1] The current draft of the full version of this paper can be found at the web address http://www.planet.edu/˜yahya/update.html.

The Herbrand base of DB, HB_{DB}, is the set of all ground atoms that can be formed using the predicate symbols and constants in $\mathcal{L}$. A *Herbrand interpretation* is any subset of HB_{DB}. A Herbrand model of DB, M, is a Herbrand interpretation such that $M \models DB$ (all clauses of DB are *true* in M). M is *minimal* if no proper subset of M is a model of DB. The set of all minimal models of DB is denoted by $\mathcal{MM}(DB)$.

Definition 2. (range-restriction) A clause C is range-restricted if every variable occurring in the head of C also appears in the body of C. A database is range-restricted if and only if all its clauses are range-restricted.

By $Atoms(C)$ we denote the set of (positive) atoms in a clause C.

By $min(S)$ we denote the set of minimal elements (relative to set inclusion) of the set S.

If A is an atom and DB is a DDDB then $DB \vee A = \{C \vee A | C \in DB\}$.

If C is a clause and DB is a DDDB then by $\mathcal{MM}(DB)_C$ and $\mathcal{MM}(DB)_{\neg C}$ we denote the set of minimal models of DB is which C is *true* and *false*, respectively. Clearly, $\mathcal{MM}(DB) = \mathcal{MM}(DB)_C \cup \mathcal{MM}(DB)_{\neg C}$.

A model is treated as a conjunction of atoms and its negation is the disjunction of all negative literals corresponding to the atoms in that model. If $M = P_1 \wedge ... \wedge P_n$ then $\neg M = M \rightarrow \perp = \neg P_1 \vee ... \vee \neg P_n$

Definition 3. A positive clause, C, is minimally derivable from DB iff it is derivable from DB but no proper subclause of C is derivable from DB. Clearly, a positive clause C is derivable from DB iff a (not necessarily proper) subclause of C is minimally derivable from DB.

Lemma 4. *[17] A clause C is minimally derivable from a DDDB, DB, iff $DB \vdash C$ and $\forall A \in Atoms(C) \exists M \in \mathcal{MM}(DB) | M \cap Atoms(C) = \{A\}$.*

Under the minimal model semantics the natural way to define database completions is in terms of the (Extended) Generalized Closed World Assumption, (E)GCWA [12, 17]. It reconciles the concepts of derivability from the completed database and being *true* in all minimal models.

We consider two classes of updates: adding a clause to a DDDB and deleting a clause from a DDDB. Under the minimal model semantics the addition can be viewed as making the clause *true* in all, and the deletion as making it *false* in at least one, minimal model of the database.

Definition 5. Let DB be a DDDB and C be a clause such that $DB \vdash C$. We say that DB' is the result of deleting C from DB iff $DB' \nvdash C$. In terms of minimal models: $\forall M \in \mathcal{MM}(DB) M \models C$ and $\exists N \in \mathcal{MM}(DB') s.t. N \not\models C$.

Definition 6. Let DB be a DDDB and C be a clause such that $DB \not\vdash C$. We say that DB' is the result of adding C to DB if $DB' \vdash C$. In terms of minimal models: $\forall M \in \mathcal{MM}(DB') M \models C$ and $\exists N \in \mathcal{MM}(DB) s.t. N \not\models C$.

There may exist a number of DB's accomplishing an update. We are interested in minimal changes in some sense, particularly in databases accomplishing the update with the least change to the minimal model structure of DB. Our approach is based on manipulating the minimal model structure of the theory to achieve the required update.

Lemma 7. *Let DB be a DDDB and let $\mathcal{M}$ be a set of models for DB.*

If $DB' = DB \cup \{M \rightarrow \bot | M \in \mathcal{M}\}$ then DB' has the set of minimal models
$\mathcal{MM}(DB') = \mathcal{MM}(DB) \setminus \mathcal{M}$.

If C is a positive clause and $DB'' = DB \cup \{M \rightarrow C | M \in \mathcal{M}\}$ then DB'' has the set of minimal models $\mathcal{MM}(DB'') = min(\mathcal{MM}(DB)_C \cup \{(M \cup \{A\})^* | M \in \mathcal{M}$ and $A \in Atom(C)\})$, where $(M \cup \{A\})^*$ is the the extension of $(M \cup \{A\})$ to a model of DB''.

Lemma 8. *Let M be an interpretation for a DDDB, DB. If $DB' = DB \vee M = \{DB \vee A | A \in M\}$ then DB' has the set of minimal models $\mathcal{MM}(DB') = min(\mathcal{MM}(DB) \cup \{M\})$.*

In the following two sections we address the issue of adding/deleting a ground positive clause C to/from a disjunctive deductive database DB. We consider several possibilities for accomplishing such an update in different classes of theories all of which lead to altering the minimal model structure of the original theory to obtain the updated database. As shown in [8, 9] it is possible to convert several other update types to this form. We will discuss possible extensions of our results to the case of nonpositive clauses in later sections.

3 Updates in Ground DDDBs

If the database is ground then it is possible to use a positive clause C to split DB into components with particular properties of minimal models regarding C.

Definition 9. Let $C = A_1 \vee A_2 \vee ... \vee A_m$. Let $A = A_1$. Rewrite DB as $\{F_i = A \vee F'_i\} \cup \{E_j = \neg A \vee E'_j\} \cup \{C_k\}$, where F_i are the clauses containing A, E_j the clauses containing $\neg A$ and C_k the clauses with no occurrences of A. Let $DB_A = \{A\} \cup \{E'_i\} \cup \{C_k\}$ and $DB_{\neg A} = \{F'_j\} \cup \{C_k\}$ Clearly,

$\mathcal{MM}(DB) = \mathcal{MM}(DB_A \vee DB_{\neg A})$ [16]. Additionally, we let $DB^{-A} = \{F'_i \vee E'_j\} \cup \{C_k\}$. DB^{-A} has no occurrences of A as well.

All the minimal models for DB_A (if any) contain A while none of the minimal models of $DB_{\neg A}$ has A since A doesn't occur in $DB_{\neg A}$.

In the next iteration we let $DB = DB_{\neg A}$, $A = A_2$ and repeat the expansion for the new DB and A and continue in this fashion to produce $DB_1, DB_2, ..., DB_m$ corresponding to the atoms of the clause C.

$DB_{-(i+1)}$ and DB_{i+1} are achieved from expanding DB_{-i}, on A_{i+1}. That is, $DB_{-(i+1)} = (DB_{-i})_{\neg A_{i+1}}$ and $DB_{i+1} = (DB_{-i})_{A_{i+1}}$ for $m-1 \leq i \leq 0$, where $DB_{-0} = DB_0$ is DB itself.

DB_i has A_i as a unit clause and has no occurrences of A_j for $j < i$.

The remaining component of the database, DB_{-m}, has no occurrences of any atom of C. Both DB_i and DB_{-i} are defined for $m \leq i \leq 0$. However, for compactness, by DB_{m+1} we denote the set DB_{-m}.

The following Lemma is an extension of a result proved in [16]:

Lemma 10. *[16] Let DB be a DDDB and let DB_{A_i}, for $m+1 \leq i \leq 0$ be as given in Definition 9 above. Then:*

$$DB = \vee_{i=1}^{i=m+1} DB_i.$$

3.1 Clause Addition

Let DB be a DDDB and let C be a positive clause of length m. Assume that $DB \not\vdash C$ and we want to modify DB to DB' such that $DB' \vdash C$. It is possible to accomplish this task by adding the clause C to DB and have $DB' = DB \cup \{C\}$. However, one may demand that the change to the minimal model structure of DB be minimal. That is we would like DB' to have exactly the same set of minimal models as DB except that an atom (or more) from C is added to those models of DB not having an atom of C. This leaves open the question of which atom(s) of C to add to each model. We can accomplish this task: by adding a single atom, some atoms or all atoms of C to every minimal model of DB not having an atom of C. To do that we need to determine the component of DB that needs to be modified in order for DB' to derive C. That is, we would like to isolate the part of DB that generates minimal models of DB not containing atoms of C. Earlier it was shown that the only component that makes the clause C nonderivable from DB is $DB_{m+1} = DB_{-m}$. We can restrict our modification to this component of DB.

Theorem 11. (Clause Addition) *Let DB be a DDDB and C be a clause such that $DB \not\vdash C$ and let DB_{A_i}, for $m+1 \geq i \geq 1$ be as defined* *above. Then: C derivable from $DB' = \vee_{i=1}^{i=m} DB_i \vee DB'_{m+1}$ ($DB' \vdash C$) if DB_{m+1} is changed to DB'_{m+1} in any of the following ways:*

1. *To add a single atom of C, say A, as a unit clause to DB_{m+1}. This is equivalent to adding A to every minimal model of DB falsifying C.*
2. *To add all the atoms of C, $A_1, A_2, ..., A_m$, as unit clauses to DB_{m+1}. This is equivalent to adding the set $\{A_1, A_2, ..., A_m\}$ to every minimal model of DB not having an atom of C.*
3. *To add C itself as a clause to DB_{m+1}. This is equivalent to generating m new models for each model not having an atom of C by adding one atom of C to every minimal model of DB not having an atom of C.*
4. *An arrangement that is intermediate between the discussed options (adding a subset of the atoms of C to DB_{m+1}).*

Note that in all cases clauses with no occurrences of atoms from the clause under consideration, $\{C_k\}$, will be retained in the updated database. Therefore, we can limit our consideration to clauses with atoms from C. These are the clauses that need to undergo change [14]. This can reduce the average cost of performing updates, though not in the worst case.

To elect among the available choices for adding C one may demand that C be minimally derivable in the sense that no proper subclause of C is derivable from the updated database DB'.

Example 1. $DB = \{P(a) \vee P(b), P(a) \vee P(c) \vee P(e), P(b) \vee P(c) \vee P(d), P(c) \vee P(d) \vee P(e)\}$. $\mathcal{MM}(DB) = \{\{P(a), P(c)\}, \{P(a), P(d)\}, \{P(b), P(c)\}, \{P(b), P(e)\}\}$. Assume we want to add the clause $P(c) \vee P(d)$.

$DB_1 = DB_{P(c)} = \{P(a) \vee P(b), P(c)\}$, $DB_{\neg P(c)} = \{P(a) \vee P(b), P(a) \vee P(e), P(b) \vee P(d), P(d) \vee P(e)\}$, $DB_2 = DB_{\neg P(c) P(d)} = \{P(a) \vee P(b), P(a) \vee P(e), P(d)\}$,

$DB_3 = DB_{\neg P(c) \neg P(d)} = DB_{m+1} = \{P(b), P(e)\}$.

Selecting $P(c)$ for addition we get $DB'_3 = \{P(b), P(e), P(c)\}$ and

$DB' = \vee_{i=1}^{i=2} DB_i \vee DB'_3 = \{P(a) \vee P(b), P(c) \vee P(d), P(a) \vee P(c) \vee P(e)\}$ which has the set of minimal models

$\{\{P(a), P(c)\}, \{P(a), P(d)\}, \{P(b), P(c)\}, \{P(b), P(d), P(e)\}\}$ and in which the clause $P(c) \vee P(d)$ is derivable.

We could have selected $DB'_3 = \{P(b), P(e), P(d)\}$ in which case we get

$DB' = \vee_{i=1}^{i=2} DB_i \vee DB'_3 = \{P(a) \vee P(b), P(c) \vee P(d), P(a) \vee P(c) \vee P(e)\}$ which has the set of minimal models

$\{\{P(a), P(c)\}, \{P(a), P(d)\}, \{P(b), P(c)\}, \{P(b), P(d), P(e)\}\}$ and in which the clause $P(c) \vee P(d)$ is also derivable.

The change to the model structure is minimal in the sense that the clause under consideration is derivable and we retained as many minimal models of DB as possible.

3.2 Clause Deletion

Now given a positive ground clause $C = A_1 \vee A_2 \vee ... \vee A_m$ derivable from DB, $(DB \vdash C)$, we want to modify DB to DB' so that $DB' \not\vdash C$. First we have the following lemma:

Lemma 12. *Let DB be a DDDB, A be an atom of the Herbrand base of DB and $DB^{-A} = \{F'_i \vee E'_j\} \cup \{C_k\}$, where F_i are the clauses containing A, E_j the clauses containing $\neg A$ and C_k the clauses with no occurrences of A and F' and E' are the result of removing the occurrences of A from F and E, respectively (as in Definition 9). Then M is a minimal model of DB iff $M \setminus \{A\}$ is a minimal model of DB^{-A}.*

Theorem 13. (Clause Deletion) *Let DB be a DDDB and $C = A_1 \vee ... \vee A_m$ be a clause such that $DB \vdash C$ (minimally). Let A be an atom occurring in C. Let M be a minimal model of DB containing A and none of the other atoms of C ($M \in \mathcal{MM}(DB)$ and$A \in M$ and $\forall A_i \neq A, A_i \notin M$). Let $M' = M \setminus \{A\}$ and let DB^{-A} be as in Lemma 12. Let $AUG_A^M(DB) = \{E'_j\} \cup \{C_k\} \cup \{A \vee B | B \in M'\}$ and let $DB' = DB_{\neg A} \vee AUG_A^M(DB)$. Then $DB' \not\vdash C$ and DB' has as its set of minimal models the minimal elements of the set $\mathcal{MM}(DB) \cup \{M'\}$. $\mathcal{MM}(DB') = min(\mathcal{MM}(DB) \cup \{M'\})$.*

This approach is biased towards a particular atom A and requires that M contain an occurrence of A and none of the other atoms in the clause to be deleted[2].

Example 2. $DB = \{P(a) \vee P(b), P(a) \rightarrow P(c) \vee P(d), P(b) \rightarrow P(c) \vee P(d)\}$. $\mathcal{MM}(DB) = \{\{P(a), P(c)\}, \{P(a), P(d)\}, \{P(b), P(c)\}, \{P(b), P(d)\}\}$. $C = P(c) \vee P(d)$. Take $P(c)$ as the candidate atom and $\{P(a), P(c)\}$ as the candidate model for removal (update).

$\{F'_i\} = \{P(a) \rightarrow P(d), P(b) \rightarrow P(d)\}, \{E'_j\} = \emptyset$, $\{C_k\} = \{P(a) \vee P(b)\}$.

$DB_{\neg P(c)} = \{P(a) \vee P(b), P(a) \rightarrow P(d), P(b) \rightarrow P(d)\}$.

$AUG_{P(c)}^{\{P(a),P(c)\}}(DB) = \{P(a) \vee P(c)\} \cup \{C_k\}$.

---

[2] One can modify the procedure to make it less sensitive to the atom and the model at hand.

$DB' = \{P(a) \vee P(b), P(b) \rightarrow P(c) \vee P(d) \vee P(a)\}$ with the minimal models $\{\{P(a)\}, \{P(b), P(c)\}, \{P(b), P(d)\}\}$. Or alternatively,

$DB' = \{P(a) \vee P(b), P(a) \rightarrow P(c) \vee P(d) \vee P(b)\}$ with the minimal models $\{\{P(b)\}, \{P(a), P(c)\}, \{P(a), P(d)\}\}$.

Any of these databases can serve as a solution to the problem at hand. The change to the model structure is minimal in the sense that the clause under consideration is not derivable anymore, and we retained as many minimal models as possible from the original minimal model structure. Note that the change in the structure of the candidate for removal model rendered other models nonminimal and forced their removal.

4 Updates in Range Restricted DDDBs

The approach to update discussed so far is applicable to ground disjunctive databases. Groundness was needed to isolate the components that have to be modified to accomplish the update. One may argue that it can be applied to nonground theories if we choose to ground them. The size of the resulting theory may be prohibitive. In this section we present algorithms for update in range restricted theories: a larger class than ground theories but still a natural one that is encountered frequently [3].

4.1 Update Through Model Suppression/Augmentation

Given a positive clause C and a DDDB, DB such that $DB \not\vdash C$ and in view of Lemma 7 it is possible to add negative clauses to DB so as to suppress the minimal models that falsify C. On the other hand, if $DB \vdash C$ and in view of Lemma 8 it is possible expand the model structure of DB so that to add a minimal model that falsifies C. We state the following results the proofs of which are immediate.

Theorem 14. (Clause Addition via Model Suppression) *Let DB be a DDDB and C be a positive ground clause such that $DB \not\vdash C$. Let $DB' = DB \cup \{M \rightarrow \perp | M \in \mathcal{MM}(DB)_{\neg C}\}$. $\mathcal{MM}(DB') = \mathcal{MM}(DB)_C$[3]. $DB' \vdash C$[4].*

As for clause deletion, this approach is not applicable since all minimal models of DB satisfy C and removing them all will result in an

[3] A similar approach can be used make C *false* in every minimal model of DB'', $DB'' = DB \cup \{M \rightarrow \perp | M \in \mathcal{MM}(DB)_C$. $\mathcal{MM}(DB'') = \mathcal{MM}(DB)_{\neg C}\}$. This may be of interest when deleting a clause is interpreted as being *false* in all minimal models of the theory.

[4] Note that if C is in no minimal model of DB then DB' is inconsistent.

inconsistent theory. This asymmetry is the result of having clause addition defined in terms of *all* minimal models and clause deletion in terms of *at least one* minimal model.

Theorem 15. (Clause Deletion via Model Expansion) *Let DB be a DDDB and C be a positive ground clause such that $DB \vdash C$. Let $DB' = DB \vee M'$, where $M' = M \setminus Atoms(C)$ and $M \in \mathcal{MM}(DB)$. $\mathcal{MM}(DB') = min(\mathcal{MM}(DB) \cup \{M'\})$. $DB \not\vdash C$.*

Example 3. Let DB be the set of clauses:

$$
\begin{aligned}
&DB = \{ \\
&\quad C_1 = \top \rightarrow P(a), &&C_2 = \top \rightarrow Q(b) \\
&\quad C_3 = P(x) \rightarrow Q(x) \vee R(x), &&C_4 = P(x) \wedge R(x) \rightarrow S(x), \\
&\quad C_5 = Q(x) \rightarrow P(x) \vee R(x), &&C_6 = S(a) \wedge R(b) \rightarrow \perp\}.
\end{aligned}
$$

$\mathcal{MM}(DB) = \{\{P(a), Q(b), P(b), Q(a)\}, \{P(a), Q(b), R(b), Q(a)\}, \{P(a), Q(b), P(b), R(a), S(a)\}\}$.

Clearly, $DB \vdash C_8$ and $DB \not\vdash C_7$.

- Addition: Add $C_7 = R(a) \vee S(b)$. $DB' = DB \cup \{$
  $P(a) \wedge Q(b) \wedge R(b) \wedge Q(a) \rightarrow \perp,\ P(a) \wedge Q(b) \wedge R(b) \wedge Q(a) \rightarrow \perp\}$
  or equivalently:
  $DB' = DB \cup \{Q(a) \wedge R(b) \rightarrow \perp, Q(a) \wedge P(b) \rightarrow \perp\}$.
- Deletion: Delete $C_8 = R(b) \vee P(b)$.
  $M = \{P(a), Q(b), P(b), Q(a)\}$.
  $M' = \{P(a), Q(b), Q(a)\}$.
  $M' = \{P(a) \wedge Q(b) \wedge Q(a)\}$.

  $$
  \begin{aligned}
  &DB = \{ \\
  &\quad C_1 = \top \rightarrow P(a), &&C_2 = \top \rightarrow Q(b) \\
  &\quad C_3' = P(x) \rightarrow Q(x) \vee R(x) \vee M', &&C_4' = P(x) \wedge R(x) \rightarrow S(x) \vee M', \\
  &\quad C_5' = Q(x) \rightarrow P(x) \vee R(x) \vee M', &&C_6 = S(a) \wedge R(b) \rightarrow \perp\}.
  \end{aligned}
  $$

  Note that rather than augmenting each clause with every atom of M' we augmented clauses by the conjunction of atoms in M' and simplified the formulas with heads intersecting with M'.

4.2 Update through Clause Expansion/Addition

Theorem 16. (Clause Addition) *Let DB be a DDDB, C be a positive clause such that $DB \not\vdash C$. Define $DB' = DB \cup \{M \rightarrow C | M \in \mathcal{MM}(DB)_{\neg C}\}$. DB' accomplishes the addition update of C. Additionally, $\mathcal{MM}(DB)_C \subseteq \mathcal{MM}(DB')$.*

There may exist several ways to add $M \to C$ and the discussion of the point for Theorem 11 is applicable here as well.

Theorem 17. (Clause Deletion) *Let DB be a DDDB, C be a positive clause such that $DB \vdash C$ and let $M \in \mathcal{MM}(DB)$ and $N = M \setminus Atoms(C)$. Let $\mathcal{C} = \{C \in DB$ such that $N \not\models C\}$. Define $DB' = (DB \setminus \mathcal{C}) \cup \{C' | Atoms(C) \cap Atoms(C') = Atoms(C)$ and $Body(C) = Body(C')$ and $C \in \mathcal{C}$ and $C' \cap N \neq \phi\}$. DB' accomplishes the deletion update of C and $\mathcal{MM}(DB') = min(\mathcal{MM}(DB) \cup \{N\})$.*

In view of Lemma 4 we may select the model that contains a single atom, say A, of C. In this case only clauses with A in the head will be falsified by N.

Note that rather than adding N to every clause of DB we restricted our modification to elements of $\mathcal{C}$.

Example 4. Let DB be the set of clauses of Example 3, i.e.:

$$
\begin{array}{ll}
\multicolumn{2}{l}{DB = \{} \\
\quad C_1 = \top \to P(a), & C_2 = \top \to Q(b), \\
\quad C_3 = P(x) \to Q(x) \vee R(x), & C_4 = P(x) \wedge R(x) \to S(x), \\
\quad C_5 = Q(x) \to P(x) \vee R(x), & C_6 = S(a) \wedge R(b) \to \bot\}.
\end{array}
$$

- Addition: Add $C_7 = R(a) \vee S(b)$.
  $DB' = DB \cup \{$
  $P(a) \wedge Q(b) \wedge R(b) \wedge Q(a) \to R(a) \vee S(b)$,
  $P(a) \wedge Q(b) \wedge S(b) \wedge Q(a) \to R(a) \vee S(b)\}$.
- Deletion: Delete $C_8 = R(b) \vee P(b)$.
  $M = \{P(a), Q(b), P(b), Q(a)\}$. $N = \{P(a), Q(b), Q(a)\}$.
  $\mathcal{C} = \{Q(b) \to P(b) \vee R(b)\}$.

$$
\begin{array}{ll}
\multicolumn{2}{l}{DB' = \{} \\
\quad C_1 = \top \to P(a), & C_2 = \top \to Q(b) \\
\quad C_3 = P(x) \to Q(x) \vee R(x) \vee P(a), & C_4 = P(x) \wedge R(x) \to S(x) \\
\quad C_5 = Q(a) \to P(a) \vee R(a), & C_6 = S(a) \wedge R(b) \to \bot, \\
\quad C_9' = Q(b) \to P(b) \vee R(b) \vee N\}. &
\end{array}
$$

Note that when dealing with a minimal model M it is frequently possible to restrict our consideration to those atoms of M that are not common to all other minimal models of the theory and still achieve the required results, possibly with more efficiency.

5 Nonpositive Clause Updates

In this section we address the issue of adding/deleting a nonpositive clause to a DDDB. As is the case for ground clauses, a clause in implication form $C = Head(C) \leftarrow Body(C)$ is in DB if and only if all minimal models of DB satisfy C. That is, C is not in DB if and only if there exists a minimal model of DB falsifying C: $\exists M \in \mathcal{MM}(DB)$ such that $M \models Body(C)$ and $M \not\models Head(C)$. The previous results can be viewed as a special case of the general clause updates.

5.1 Clause Addition Update

Let DB be a DDDB and C be a clause such that $\mathcal{MM}(DB)_{\neg C} = \{M \in \mathcal{MM}(DB)\ M \models Body(C)$ and $M \not\models Head(C)\}$ is not empty. Let $DB' = DB \cup \{M \rightarrow Head(C) | M \in \mathcal{MM}(DB)_{\neg C}\}$. DB' accomplishes the addition of C to DB.

The result can be viewed as an application of Theorem 16. Clearly there may exist many ways to have $M \rightarrow Head(C)$ as was discussed in Theorem 11.

5.2 Clause Deletion Update

To delete C from DB we have to create a minimal model of the updated theory that doesn't satisfy C. To minimize the change we always select a model that satisfies $Body(C)$ and make sure that the modification of that model retains the satisfiability of $Body(C)$ but falsifies $Head(C)$. Clearly, once that model is selected and since $Head(C)$ is a positive clause then we can use the results of earlier sections to accomplish the deletion of $Head(C)$ (Theorem 17).

It may be the case that such a model doesn't exist. That is, all minimal models of DB satisfy C by falsifying its body. Then we have to create it. That is, we need to create a model that satisfies the body of C but not its head.

Example 5. Let DB be the set of clauses of Example 3, i.e.:

$$
\begin{array}{ll}
DB = \{ & \\
\quad C_1 = \top \rightarrow P(a), & C_2 = \top \rightarrow Q(b), \\
\quad C_3 = P(x) \rightarrow Q(x) \vee R(x), & C_4 = P(x) \wedge R(x) \rightarrow S(x), \\
\quad C_5 = Q(x) \rightarrow P(x) \vee R(x), & C_6 = S(a) \wedge R(b) \rightarrow \bot\}.
\end{array}
$$

– Addition:
Add $C_{11} = P(a) \rightarrow Q(a)$.
$DB' = DB \cup \{$
$P(a) \wedge Q(b) \wedge P(b) \wedge R(a) \wedge S(a) \rightarrow Q(a)$,
$P(a) \wedge Q(b) \wedge S(b) \wedge Q(a) \rightarrow R(a) \vee S(b)\}$.
– Deletion:
Delete $C_{12} = Q(b) \rightarrow P(b) \vee R(b)$.
Select $M = \{P(a), Q(b), P(b), Q(a)\}$. $N = \{P(a), Q(b), Q(a)\}$.
and delete $P(b) \vee R(b)$ from M.
$\mathcal{C} = \{Q(b) \rightarrow P(b) \vee R(b)\}$.

$$
\begin{array}{ll}
DB' = \{ & \\
C_1 = \top \rightarrow P(a), & C_2 = \top \rightarrow Q(b) \\
C_3 = P(x) \rightarrow Q(x) \vee R(x) \vee P(a), & C_4 = P(x) \wedge R(x) \rightarrow S(x) \\
C_5 = Q(a) \rightarrow P(a) \vee R(a), & C_6 = S(a) \wedge R(b) \rightarrow \bot, \\
C_9' = Q(b) \rightarrow P(b) \vee R(b) \vee N\}. &
\end{array}
$$

Note that the clause $S(b) \rightarrow R(b)$ is true in all minimal models but no minimal model has $S(b)$.

6 Conclusion and Remarks

We presented an approach to the update problem in DDDBs based on modifying the theory to achieve the required change to the minimal model structure. We considered adding/deleting ground positive clauses to ground DDDBs then to range restricted DDDBs then extended the results to general clause updates.

Update minimality was defined in terms of its effect on the minimal model structure of the theory. This sounds natural if one interprets minimal models as the possible states in which the theory can be (had our information about the world been complete). In such a case it is natural to try to minimize the effect of an update on the number of models as well as the effect on the content of individual models.

The issue of update minimality received much attention in the literature [5, 6, 10, 8]. Examples are to define update minimality in terms of the weakness of the update [10] or in the number of clauses modified to accomplish the update [8]. Clearly, least change to the minimal model structure discussed here need not be minimal by other measures of update minimality.

Our approach depends heavily on computing one or more minimal models of the theory with particular properties. An algorithm to perform this computation was given in [3] and proven to be minimal model sound and complete for the class of range restricted theories: it returns all and only minimal models of its input. The algorithm depends heavily in utilizing denial rules to guide the model search process. The detailed discussion, an implementation and comments on efficiency results can also be found in [3].

Our approach differs from those discussed in [13, 4] in that we consider a more general class of theories and updates. In [7] DDDBs are considered and the approach can be extended to the case of range restricted theories. However, they treat only positive clause updates and our approach tends to expand much less of the model structure through exploiting the ordering on model generation induced by the update clause [16, 3]. We don't assume a particular representation of the model structure nor do we require it all to be present for an update. The models are generated "ad-hoc" and in the order most appropriate for the update. We offered several methods for accomplishing updates as opposed to the one offered in [7] which can be viewed as a special case of our methods.

Combined with the algorithms given in [16, 3] for minimal model generation our approach can result in substantial savings in constructing the minimal model structure as opposed to generating a complete set of models then minimizing among them. Additionally, we can have the update clause guide the system to generate the most relevant models first in the spirit of [15]. In a sense we can integrate the updating process into the model generation process.

The work [10] addressed the view update problem in stratified disjunctive databases. This reduces to positive clause addition/deletion updates when DDDBs are considered. Our results can be employed in that approach as well. The work handles nonpositive additions as well but we offer more choices and we advise employing algorithms that enable directed search for the required models. The approach of [8] deals with normal databases and it is of interest to extend our results to normal disjunctive databases.

We don't restrict ourselves to modifying the extensional part of the theory as is generally the case in [8, 7, 10] to accomplish an update. Rather, we do that by adding arbitrary clauses to the DDDB or modifying its IDB clauses. Such a modification is also employed in [14, 1]. We believe that it is possible to modify our approach so that it keeps track of the clauses added for the sole purposes of accomplishing an update as opposed to the *original* clauses of the theory. This may help in making the updates reversible in the sense that a sequence of adding and deleting the same clause may take us back to the original theory which is not usually the case for our algorithms as presented or for most other algorithms available.

The updates as presented here were interpreted as additions/deletions to the theory (EDB,IDB). One can also talk about adding/deleting integrity constraints. While the interpretation of the integrity constraint addition is obvious: retain all the minimal models that satisfy the constraint and remove the others, the deletion is not. To delete a constraint it may be sufficient to just stop enforcing it and not necessarily making it false in some minimal model. So a deleted IC may be true in all minimal models still.

Our restriction to the case of range-restricted theories is dictated by the class of theories that can be handled by the minimal model generation procedure rather than a limitation of the adopted approach.

Efficiency improvement methods can be employed to improve the performance of the algorithms presented here. Examples are the incremental generation of models and the inclusion of the relevant portions of the constraints added to the theory.

Topics for future work include extending the approach reported here to larger classes of theories such as theories with negation in rule bodies. Another possibility is to investigate possibility of accomplishing minimal updates under our criteria using other (say syntactic) methods and to compare them with the methods discussed here.

Acknowledgements:

I thank Dietmar Seipel for his comments on earlier drafts of this paper and the two anonymous referees for their valuable reports.

References

1. K. Benkerimi and J. C. Shepherdson. Partial deduction of updatable definite logic programs. *Journal of Logic Programming*, 18:1–26, 1994.
2. F. Bry. Intensional Updates: Abduction vis Deduction. In D. Warren and P. Szeredi, editors, *Proceedings of the International Conference on Logic Programming*, pages 561–575, MIT Press, 1990.
3. F. Bry and A. Yahya. Minimal model generation with positive unit hyper-resolution tableaux. In P. Miglioli, U. Moscato, D. Mundici, and M. Ornaghi, editors, *Proceedings of the Fifth Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, pages 143–159, Palermo, Italy, May 1996. Springer-Verlag. Vol. 1071, Full version: http://www.pms.informatik.uni-muenchen.de/publikationen/.
4. R. Fagin, G. Kuper, J. Ullman, and M. Vardi. Updating logical databases. In *Advances in Computing Research*, pages 1–18, 1986.
5. P. Gärdenfors. Belief revision: An introduction. Belief Revision. (P. Gärdenfors Editor), Cambridge University Press, 1992. July 1995. PP. 1-20.

6. P. Gärdenfors and H. Rott Belief Revision, Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 4, D. M. Gabbay, C. J. Hogger, and J. A. Robinson, eds., Oxford University Press, Oxford 1995. PP. 35-132.
7. J. Grant, J. Gryz, and J. Minker. Updating disjunctive databases via model trees. Technical Report CS-TR-3111 UMIACS-TR-93-70, Department of Computer Science and UMIACS, University of Maryland, College Park, July 1995.
8. A. Guessoum and Lloyd J. Updating knowledge bases. *New Generation Computing Journal*, 8:71–89, 1990.
9. A. Guessoum and Lloyd J. Updating knowledge bases ii. *New Generation Computing Journal*, 10:73–100, 1991.
10. J.Grant, J.Horty, J. Lobo, and J. Minker. View updates in disjunctive deductive databases. *Journal of Automated Reasoning*, 11:249–267, 1993.
11. J. Lobo, J. Minker, and A. Rajasekar. *Foundations of Disjunctive Logic Programming*. MIT Press, 1992.
12. J. Minker. On indefinite databases and the closed world assumption. In *Lecture Notes in Computer Science 138*, pages 292–308. Springer-Verlag, 1982.
13. R. Reiter. On specifying database updates. *J. Logic Programming*, 25(6):53–91, June 1995.
14. F. Rossi and S. Naqvi. Contributions to the view update problem. In *Proc. of the International Conference on Logic Programming, Lisbon*, pages 398–415, 1989.
15. A. Yahya. Generalized query answering in disjunctive databases: Procedural and nonmonotonic aspects. In *Proceedings of the Fourth International Conference on Logic Programming and Nonmonotonic Reasoning LPNMR'97*, Dagstul, Germany, July 1997. Springer-Verlag. To Appear.
16. A. Yahya, J.A. Fernandez, and J. Minker. Ordered model trees: A normal form for disjunctive deductive databases. *J. Automated Reasoning*, 13(1):117–144, 1994.
17. A. Yahya and L.J. Henschen. Deduction in Non-Horn Databases. *J. Automated Reasoning*, 1(2):141–160, 1985.